International Journal of Interactive Multimedia and Artificial Intelligence

June 2022, Vol. VII, Number 4
ISSN: 1989-1660

unir LA UNIVERSIDAD EN INTERNET

"Errors using inadequate data are much less than those using no data at all."
Charles Babbage

INTERNATIONAL JOURNAL OF INTERACTIVE MULTIMEDIA AND ARTIFICIAL INTELLIGENCE

ISSN: 1989-1660 –VOL. 7, NUMBER 4

OPEN ACCESS JOURNAL

ISSN: 1989-1660

The International Journal of Interactive Multimedia and Artificial Intelligence is covered in Clarivate Analytics services and products. Specifically, this publication is indexed and abstracted in: *Science Citation Index Expanded, Journal Citation Reports/ Science Edition, Current Contents®/Engineering Computing and Technology.*

COPYRIGHT NOTICE

Editor's Note

The International Journal of Interactive Multimedia and Artificial Intelligence – IJIMAI (ISSN 1989-1660) provides an interdisciplinary forum in which scientists and professionals can share their research results and report new advances in Artificial Intelligence (AI) tools or tools that use AI with interactive multimedia techniques.

The present volume (June 2022), consists of 20 articles of diverse applications of great impact in several fields. The issue consistently showcases the utilization of AI techniques or mathematical models with an artificial intelligence base, as a standard element. Different manuscripts on usability and satisfaction, machine learning models, genetic algorithms, computer entertainment technologies, oral pathologies, optimistic motion planning, data analysis for decision making, etc. can be found in this volume.

The volume begins with a systematic review of the literature on recommender systems that use the information on social relationships between users. As the main findings, D. Medel et al. present a complete review where social relations were classified into three groups: trust, friend activities, and user interactions. Likewise, the collaborative filtering approach was the most used, and with the best results, considering the methods based on memory and model. The most used metrics and the recommendation methods studied in mobile applications are also mentioned in this article.

Neural collaborative filtering is an important field in recommender systems, as this provides some models that obtain accurate predictions and recommendations. The following manuscript by J. Bobadilla et al. provides a proposed neural architecture; and also tests that the quality of its recommendation results is as good as the state of art baselines. Experiments have been performed making use of four popular public datasets, showing generalizable quality results. Overall, the proposed architecture improves individual rating prediction quality, maintains recommendation results, and opens the doors to a set of relevant collaborative filtering fields.

Previous articles have presented a neural architecture, but the following one presented by Z. Anari et al. proposes an appropriate membership function for fuzzy association rule mining. Membership functions have a significant impact on the outcome of the mining association rules. This study, as the first attempt, used a team of continuous action-set learning automata (CALA) to find both the appropriate number and positions of trapezoidal membership functions (TMFs). Additionally, to increase the convergence speed of the proposed approach and remove bad shapes of membership functions, a new heuristic approach has been proposed. Experiments on two real data sets showed that the proposed algorithm improves the efficiency of the extracted rules by finding optimized membership functions.

The next article by Y. Song analyses the creative methods of stimulating divergent thinking, aggregation thinking, and transformation thinking from the innovation principle of TRIZ theory as the origin, and applies them to the creative mechanism and application program of print advertising creativity. The whole process is led by rational principles of perceptual thinking, driven by specific principles of abstract images, to explore the thinking source of the creative design essence of print advertising. The theory and its application mechanism become a new thinking method and application attempt in the creative field of print advertisement.

At present accessibility in interactive applications has been profoundly improved, especially when it comes to dealing with the experiences of blind and visually impaired people while performing everyday tasks. In their work, M. Lopez-Ibanez et al. present a series of articles that explore different trends in the field of accessible video games for the blind or visually impaired. Reviewed articles are distributed in four categories covering the following subjects: (1) video game design and architecture, (2) video game adaptations, (3) accessible games as learning tools or treatments, and (4) navigation and interaction in virtual environments. From this work, a relative stagnation in the field of human-computer interaction for the blind and visually impaired is detected.

If we go into the medical field A. Laishram et al. provide a novel approach to classify different oral pathologies using Orthopantomogram (OPG) images based on Convolutional Neural Network (CNN). Their article provides a novel approach for the classification of types of teeth (viz., incisors, and molar teeth) and also some underlying oral anomalies such as fixed partial denture (cap) and impacted teeth. In their work, an algorithm implementing CNN with Dropout and then the fully connected layer has been trained using hybrid GA-BP learning. Using the dropout regularization technique, overfitting has been avoided and thereby making the network correctly classify the objects. The CNN has been implemented with different convolutional layers and the highest accuracy of 97.92% has been obtained with two convolutional layers.

In the field of information and communication technologies, Internet of Things, Machine learning, and Cloud computing are the emerging domains. These techniques can help save the lives of millions of people in the medical assisted environment and can be utilized in the healthcare system where health expertise is less available. In this regard, J. Ahamed et al. develop an efficient cardiovascular disease prediction model for Jammu and Kashmir (India). Hence, they conclude this work by stating that the combination of IoT, Machine learning, and Cloud computing is shown to be a future reality for the prediction of diseases in general and cardiovascular diseases.

Changing to the area of sampling-based motion planning in the field of robot motion planning, the following article by L. Kenye et al. presents a new sampling-based planning strategy called Optimistic Motion Planning using Recursive Sub-Sampling (OMPRSS), for finding a path from a source to a destination without having to construct a roadmap or a tree.

Switching topics, the volume continues with an article proposed by P.S. Lamba et al. whose focus is based on a novel real-time multimodal eye blink detection method using an amalgam of five unique weighted features namely (Vertical Head Positioning, Orientation Factor, Proportional Ratio, Area of Intersection, and Upper Eyelid Radius), extracted from the circle boundary formed from the eye landmarks. Precision, recall, F1-score, and ROC curve measure the proposed method's performance qualitatively and quantitatively. Increased accuracy (of around 97.2%) and precision (97.4%) are obtained compared to other existing unimodal approaches.

Video surveillance is one of the important state-of-the-art topics to monitor different areas of modern society surveillance like the general public surveillance system, city traffic monitoring system, and forest monitoring system. Hence, surveillance systems have become especially relevant in the digital era. The need for video surveillance systems and their video analytics has become inevitable due to an increase in crimes and unethical behaviour. Thus, enabling the tracking of an individual object in video surveillance is an essential part of modern society. In this regard, in the next article, M. Adimoolam et al. propose a system that has successfully tracked multiple objects from multiple channels and is a combination of dense block, feature selection, background

DOI: 10.9781/ijimai.2022.05.005

subtraction, and Bayesian methods. The results of the experiment conducted demonstrated an accuracy of 98% and 1.11 prediction time and these results have also been compared with existing methods such as Kalman Filtering (KF) and Deep Neural Network (DNN).

In recent years facial verification has experienced a breakthrough, not only due to the improvement in accuracy of the verification systems but also because of their increased use. This use would extend more if the problems of complex calculation of Deep Learning models, that usually need to be executed on machines with specialized hardware, were solved. This would allow to run this software on computers with low computing resources, such as Smartphones or tablets. To solve this problem, the next article presents the proposal of a new neural model, called Light Intrusion-Proving Siamese Neural Network, LIPSNN. This new light model, proposed by A. Alcaide et al., which is based on Siamese Neural Networks, is fully presented from the description of its two-block architecture, going through its development, including its training with the well-known dataset Labelled Faces in the Wild, LFW; to its benchmarking with other traditional and deep learning models for facial verification in order to compare its performance for its use in low computing resources systems for facial recognition. It can be concluded that the LIPSNN can be an alternative to the existing models to solve the facet problem of running facial verification in low computing resource devices.

Within the same subject, the following article proposed by N.K. Benamara et al. presents a new heterogeneous face recognition approach. This approach includes four scientific contributions. To show the efficacy and the robustness of the proposed TV-CycleGAN (Cycle Generative Adversarial Network), experiments have been applied on three challenging benchmark databases, including different real-world scenarios. This approach also outperforms some recent state-of-the-art methods in terms of F1-Score, AUC/EER, and other evaluation metrics.

One of the most popular topics at present is the Internet of Everything (IoE), where all devices are connected to the web. Large-scale networking benefits the community by increasing connectivity and giving control of physical devices. On the other hand, there exists an increased 'Threat' of an 'Attack'. In their article, M. Deore et al. propose the use of a visualization technique where the disassembled malware code is converted into grey images, as well as the use of Image Similarity-based Statistical Parameters (ISSP) such as Normalized Cross-correlation (NCC), Average difference (AD), Maximum difference (MaxD), Singular Structural Similarity Index Module (SSIM), Laplacian Mean Square Error (LMSE), MSE and PSNR. Identification of malware (testing phase) is also performed in less time. The fusion of image and statistical parameters enhances the system performance with greater accuracy.

However, if we are going to discuss the topic of the national security system, one of the most dangerous situations a warship may face is a missile attack launched from other ships, aircraft, submarines, or land. In addition, given the current scenario, it is not ruled out that a terrorist group may acquire missiles and use them against ships operating close to the coast, which increases their vulnerability due to the limited reaction time. One of the means the ship has for its defense is decoys, designed to deceive the enemy missile. However, for their use to be effective it is necessary to obtain, in a quick way, a valid launching solution. In their work, R. Touza et al. design a methodology to solve the problem of decoy launching and to provide the ship immediately with the necessary data to make the firing decision.

The order in which the trajectory is executed is a powerful source of information for recognizers. However, there is still no general approach for recovering the trajectory of complex and long handwriting from static images. In the next article, M. Diaz et al. introduce a new system to estimate the order recovery of thinned static trajectories, which allows to effectively resolve the clusters and select the order of the executed pen-downs. They expect the proposed system, whose code is made publicly available to the research community, to reduce potential confusion when the order of complex trajectories is recovered, and this will in turn make the trajectories recovered to be viable for further applications, such as velocity estimation.

Another of the most popular topics at present is machine learning-based supervised single-channel speech enhancement. It has achieved considerable research interest. M.I. Khattak et al. propose an extended Restricted Boltzmann Machine (RBM) for the spectral masking-based noisy speech enhancement which is described in their article. The results showed that the proposed method successfully attenuated the noise and gained improvements in speech quality and intelligibility over conventional approaches.

Among the various fields, one of the most sought-after fields for Artificial Intelligence models is the field of education. One of the open problems within this field is the prediction of students' grades. This problem aims to predict early school failure and dropout and to determine the well-founded analysis of student performance for the improvement of educational quality. In this regard, the work by H. Alonso-Misol Gerlache et al. provides a model which deals with the problem of predicting grades of UNIR university master's degree students, and also it is able to predict situations with an accuracy above 96%.

Within the same field, the following article proposed by A.B. Urbina Nájera et al. presents an experimental study to obtain a predictive model that allows anticipating a university dropout. The study uses 51,497 instances with 26 attributes obtained from social sciences, administrative sciences, and engineering collected from 2010 to 2019. Artificial neural networks and decision trees were implemented as classification algorithms, and also, algorithms of attribute selection and resampling methods were used to balance the main class. The model has allowed predicting an approximate number of possible dropouts per period, contributing to the involved instances in preventing or reducing dropouts in higher education.

Jumping from the education field to the business and research field, data mining makes it possible to explore and find unseen connections between variables and facts observed in different domains, helping us to better understand reality. The programming methods and frameworks used to analyse data have evolved over time. Currently, the use of pipelining schemes is the most reliable way of analysing data and due to this, several important companies are currently offering this kind of service. M. Novo-Lourés et al. focus specifically on the pipelining schemes. In this context, this study introduces different improvements, such as the design of different types of constraints for the early detection of errors, the creation of functions to facilitate debugging of concrete tasks included in a pipeline, the invalidation of erroneous instances and/or the introduction of the burst-processing scheme. Adding these functionalities, they developed Big Data Pipelining for Java, a fully functional new pipelining framework that shows the potential of these features.

Volume finishes by demonstrating a novel approach by A.J. Fernández-García et al. to support preliminary data analysis in the engineering field that enables engineers and researchers to quickly and easily analyse the potential for inferring knowledge that may lie hidden in their data. Similarly, it assists them in comparing machine learning models using different implementations from different providers of well-known algorithms, without the need of prior knowledge about how to create those models with each provider.

Nilanjan Dey
Associate Editor
JIS University, Kolkata, India

TABLE OF CONTENTS

Social Relations and Methods in Recommender Systems: A Systematic Review

Diego Medel[1], Carina S. González-González[2]*, Silvana V. Aciar[1]

[1] Faculty of Science, National University of San Juan (Argentina)
[2] Department of Computer Engineering and Systems, University of La Laguna (Spain)

Received 30 October 2021 | Accepted 3 November 2021 | Published 20 December 2021

LA UNIVERSIDAD
EN INTERNET

Abstract

With the constant growth of information, data sparsity problems, and cold start have become a complex problem in obtaining accurate recommendations. Currently, authors consider the user's historical behavior and find contextual information about the user, such as social relationships, time information, and location. In this work, a systematic review of the literature on recommender systems that use the information on social relationships between users was carried out. As the main findings, social relations were classified into three groups: trust, friend activities, and user interactions. Likewise, the collaborative filtering approach was the most used, and with the best results, considering the methods based on memory and model. The most used metrics that we found, and the recommendation methods studied in mobile applications are presented. The information provided by this study can be valuable to increase the precision of the recommendations.

Keywords

Collaborative Filtering, Recommender Systems, Social Relationships, Systematic Review, Trust.

DOI: 10.9781/ijimai.2021.12.004

I. Introduction

Due to the large amount of information traffic generated by social networks, researchers are presenting various techniques in the recovery of relevant information. The delivery of personalized and adaptive content is a research problem in information retrieval (IR) and recommender systems (RS). Collaborative filtering (CF) is the most widely used model with the best results in RSs [1]. In general, RSs are classified into collaborative filtering and content-based filtering (CBF) [1] [2]. CF estimates the interest of an item according to the interests of 'similar' users to the target user, with item ratings being the most used information in CF [3]. Likewise, collaborative filtering algorithms are divided into two main categories: memory-based and model-based methods. Memory-based methods use similarity measures that act on the user-item rating matrix.

The similarity metric is used to calculate the distance between a user-item pair. Model-based methods, on the other hand, use machine learning algorithms [4]. The scarcity of data (less information) causes problems such as cold start. This lack of information on the scalability of the system affects the recommendations [1]. Researchers agree that using a user’s contextual information can improve recommender system performance. However, the primary concern in designing a system with context information is finding those factors that are of value for prediction or recommendation. Another essential aspect to consider is privacy and security [5]. The authors acknowledge as a strategy, to use additional information, provided mainly from social networks in the recommender systems to reduce these drawbacks, such as the use of context information (time and location) of the event [6], user-generated tags [3]. The social influence derived from friendship relationships and interactions between users of a social network can be used to improve the accuracy of recommendations [7]. Today, social networks are used by millions of people. Virtual sociability between people can occur through publications, comments, images, likes, etc. Social networks, such as Instagram, allow us to publish photos (and videos) for your followers. In this sense, Twitter will enable us to post text with a character limit, or in Tinder, it is possible to contact people within a certain radius to make appointments, which are some examples. Applications such as Foursquare, Facebook Places, and Yelp are popular social networks among location-based services, allowing users to share their geographic location and location-related content online quickly. Other types of social networks are social event-based, such as Meetup, for group meetings.

The objective of this work is the analysis of the current recommendation methods, the algorithms they use and whether information about social relationships is taken into account. The present article is organized in the following way: Section II defines the methodology used to search for the primary documents in this paper; Section III reports the results of the study and, finally, conclusions are presented in Section IV.

II. Method

This paper uses the systematic literature review (SLR) methodology proposed by Kitchenham and Charters [8]. This method is rigorous and well-defined, allowing us to obtain a structured and well-organized report with a clear definition of how the process is carried out. The objective is to specify research questions and to search for relevant studies considered as primary articles, identifying the state in the research area, as well as evaluating the contributions and gaps to draw partial conclusions for each research question and to build

* Corresponding author.

E-mail address: carina.gonzalez@ull.edu.es

an overall outcome of the report. The search was divided into three phases: planning the review, conducting the review, and reporting the results. Each phase is explained in the following subsections.

A. Planning the Review

1. Identification of the Need for Revision

The introduction mentioned the drawbacks of a recommender system due to the constant growth of information. For this reason, the researchers present various techniques for recovering relevant information. However, with the use of social networks, it has been possible to obtain meaningful data from the user (preferences, activities, location, friendships, etc.), which has improved the RS's accuracy. However, the need arises to analyze what information in the user's social context is useful and would allow the recommendations to be improved. In this paper, we studied which information describing social relationships is used in RS.

2. Specification of Research Questions

The objective of this paper is to provide answers to four research questions (see Table I).

TABLE I. Research Questions

	Description
RQ1	Which social relations information is considered in the construction of recommender systems?
RQ2	What methods or tools are used by RSs that consider social relations information?
RQ3	How are recommender systems that employ social relations evaluated?
RQ4	Which RS applications use social relations information?

The answers to these questions can be easily linked to the objective: to identify social relationships as information (RQ1), to know methods or techniques that implement RSs with this information (RQ2), to know the impact of their use on the accuracy of recommendations predictions (RQ3) and, besides, to identify the applications that use this kind of information (RQ4).

3. Search Method Definition

Two search methods were chosen and applied sequentially: automatic search and snowballing search. In the first instance, a search for primary articles in digital libraries was performed. Three databases were used. After obtaining a series of primary papers, we performed the snowballing technique proposed by Wohlin [9]. In terms of the data sources of the papers to be included in the search process, we selected the Scopus, IEEE, and ACM Digital Library electronic databases, as they fulfill the following requirements [10]:

- The database is available for us through our institution.
- The database can use logical expressions or a similar mechanism.
- The database allows full-length searches or searches only in specific fields of the works.
- The database allows additional filtering options such as publication year or publication language.
- The databases contain the most relevant journals and conference papers in the field of computer science.

The review period was from January 2014 to March 2021 because this area is considered very recent. An advanced search was conducted using these keywords: ("recommender system" OR "recommendation systems") AND ("social relations" OR "social network" OR "social influence") AND ("context" OR "context-aware").

B. Conducting the Review

This section describes the selection criteria for the primary papers, the process of searching and selecting studies, a synthesis of the data extraction, and an explanation of a validity check of the set of documents obtained.

1. Defining Exclusion and Inclusion Criteria

A set of inclusion and exclusion criteria was defined in this systematic review. Specifically, five inclusion criteria (IC) and the corresponding four exclusion criteria (EC) were defined:

- IC1: Type of publication: Empirical research and peer-reviewed articles and systematic reviews. AND
- IC2: Recommender system: inclusion of social relations information. AND
- IC3: Keywords defined in the search: "recommender system", "recommendation systems", "social relations", "social network", "social influence", "context", "context-aware". AND
- IC4: Period: Published from January 1, 2014 to March 31, 2021. AND
- IC5: Publication criterion: Written in English, any country.

According to the exclusion criteria:

- EC1: Type of publication: No original data, such as reports, opinion studies, essays, or comments and no research. OR
- EC2: No abstract available (first screening). OR
- EC3: Study could not be retrieved (second screening). OR
- EC4: The paper is not written in English.

2. Search Process and Study Selection

We used a combination of keywords in the search. It is important to mention that different combinations of search terms were applied to establish a result that maximizes retrieval (trying to track all the literature) and accuracy (that the articles found are relevant). Two search methods were used, the automatic search and the snowball search. These methods were divided into five steps to be applied in the searching of papers and to select a set of primary documents to extract the data for presentation in this review. The first step for study selection is an automatic search. This process involves applying search strings to digital databases to obtain the first set of primary studies. The second step is the analysis of the title of the selected papers: in this step, the inclusion/exclusion criteria, defined above, are applied. The third step is the analysis of metadata. We took the metadata found in the results of the previous phase studies. These metadata refer to the abstract and keywords of each article and determine whether they meet the selection criteria. The fourth step is the analysis of the full text. In this step, the full text of the articles was obtained to carry out a more exhaustive analysis of their compliance with the selection criteria. Those that met the inclusion criteria were chosen. The fifth step is the snowball search technique. The last step consists of applying the selection criteria to the research works found in the second search method: The snowball search technique. The aim is to select the documents that may have escaped the automatic search to find all possible evidence. This method consists of reading the list of references (Backward Snow Balling) for each article in the set of items and analyzing the quotes made on these articles (Forward Snow Balling), to find other sources or primary articles.

3. Data Extraction and Synthesis

As we mentioned above, the implementation of all steps of the research method described in the previous section resulted in 109 articles considered as primary study publications between January 2014 and March 2021. These detail or propose the development of a recommender system using social relations information. The results of each step of this work are described below.

The results obtained after carrying out this process are described through a PRISMA flow [11] (Fig. 1) :

After applying the search strings in each source, 563 papers were collected, of which 266 are from Scopus, 198 from IEEE Xplorer and 99 from ACM Digital Library.

- After removing 140 duplicated papers. Once the criteria were applied to title, abstract and Metadata, there are 165 papers (39 % of the unique papers retrieved).
- 103 full-text papers were then analyzed. (62 articles were excluded). Six papers were added by applying the snowballing search.
- Finally, a total of 109 papers were analyzed (25.76% of the unique papers retrieved).

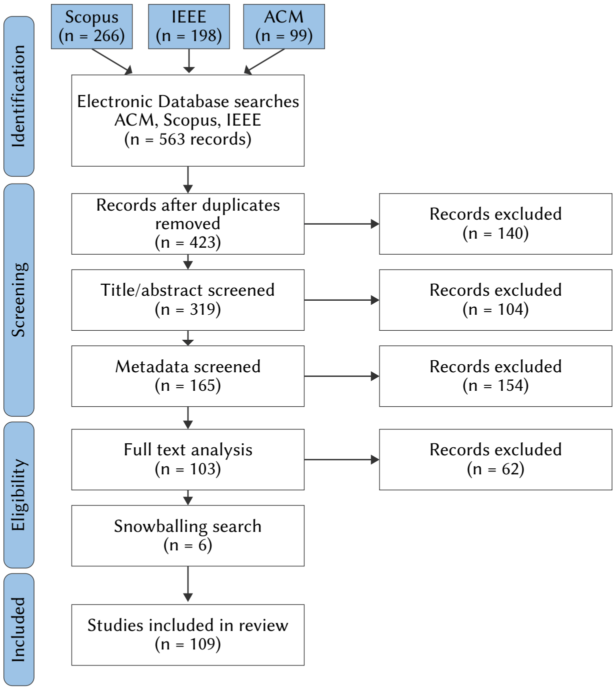

Identification
Screening
Eligibility
Included
Scopus (n = 266)
IEEE (n = 198)
ACM (n = 99)
Electronic Database searches ACM, Scopus, IEEE (n = 563 records)
Records after duplicates removed (n = 423)
Records excluded (n = 140)
Title/abstract screened (n = 319)
Records excluded (n = 104)
Metadata screened (n = 165)
Records excluded (n = 154)
Full text analysis (n = 103)
Records excluded (n = 62)
Snowballing search (n = 6)
Studies included in review (n = 109)

Fig. 1. PRISMA flow. Adapted from [11].

4. Validation Control

To reduce bias and subjectivity in the article selection process, two researchers were randomly selected to analyze a series of articles and, under their criteria, determine which items met and did not meet the selection criteria. The researchers worked from steps two through four of the search and selection process described above. To do this, 150 articles (35% of the total obtained in step 1) were randomly selected for each reviewer. Still, there was an attempt to maintain an overlap of 15 articles and reduce bias among reviewers. The papers accepted by each reviewer were incorporated into the primary set of documents, and no decision problems arose between the repeated documents analyzed by each reviewer.

C. Reporting Results

To develop the analysis of the primary studies, section III of this paper presents a report of the results obtained (last step of the applied article selection methodology).

III. Results

In this section, the results of the systematic review are reported and discussed to answer the four questions mentioned in the previous section. Table II summarize the main results of this review.

Following, we describe the main results organized by different dimensions: a) social relations information considered in recommender systems; b) evaluation of recommender system that implement this type of information; c) methods or tools used by RSS who consider social relations information and d) recommender systems applications that consider social relations information.

A. Social Relations Information Considered in Recommender Systems

As an answer to question RQ1, the works analyzed to show that the information considered as social relations are categorized into three main groups. Being trust, friend activities, and interactions between users. However, these groups are not disjunctive (see Table I).

1. Trust Among Users

Trust is defined as the richness of interactions and commitment among network members that produce positive results. In a general sense, it indicates the existence of affinity between users [35]. Trust is useful in improving the accuracy of recommendations [46]. In turn, [57], [61] states that recommender systems can benefit from incorporating relationships of distrust by considering specific differences between users. The review noted that trust relationships might be explicitly visible or calculated [62]. Both [36]–[39], [46], [61]–[66] assume that the value of trust is explicitly proportionate. That is, in their work, they assume the existence of trusted networks between users. The social network's Epinions and Douban allow the user to explicitly specify other users as trusted (to the trusted list) or untrusted (to the blocked list) if the content is valuable or not useful to the user.

Depending on each approach, the authors choose different ways to calculate the trust value. In [40], they raise the concept of trust transitivity, i.e., if the user trusts user b and user c, then user a trusts user c. In [41], they adopt the confidence calculation from the number of followers. That is, if the user "a" follows user "b" and friends of "b", then user "b" has a high confidence value. Another way is to evaluate the degree of trust between two users based on the preferences of the friends [42], the number of friends in common [43] or through social interactions (comments, publications) [40], as well as to consider the time factor, denoting a higher level of trust in recent interactions [44], [45].

However, some authors choose to classify trust according to different criteria. In [35], they propose a calculation involving two types of confidence. On the one hand, popular trust describes how popular the user is (if people read their posts, many followers, etc.). On the other hand, engagement trust describes users who engage in the social network, such as replying to posts, sending friend requests, etc. Similarly in [47], they consider local trust according to the user profile (qualifications, experience) and global trust according to their public information. Another way to measure trust is according to the reputation of the user, according to the number of users who trust him [48].

2. Friend Activities

According to [12], people are often influenced by their friends and neighbors and tend to visit places that their friends have already visited. Also, people do not usually do activities alone, but do them together [23]. In this study, places visited and/or participation in events held by users was considered social activities. Social networks such as Foursquare or Yelp allow users to rate their experience according to their experience and the places they have visited. Similarly, Meetup enables users to rate events in which they have participated.

Location-Based Social Networks (LBSN) allow users to check-in specific locations (points of interest, POIs) where they are and write a review of the location. Each check-in enables users to obtain

TABLE II. General Summary

Social Relations Information Considered in Recommender Systems		
Friends' activities	[6], [12]–[34]	
Trust among users	[4], [35]–[66]	
Interactions between users	[5], [6], [15], [27], [30]–[32], [34], [44], [67]–[111]	
Evaluation of Recomender System that Implement this Type of Information		
Accuracy	[19], [41], [42], [91], [107]	
Recall	[1]–[3], [15], [20], [22], [23], [28]–[33], [41], [42], [44], [49], [50], [54], [56], [64]–[66], [69], [76], [78], [80], [103], [110], [112], [113]	
Precision	[1]–[3], [14], [15], [19], [20], [22], [23], [28]–[33], [41], [42], [44], [48]–[50], [54], [56], [60], [66], [69], [72], [73], [76], [78], [80], [97], [98], [100], [103], [110], [112], [113]	
F-measures	[3], [16], [20], [22], [29], [41], [42], [44], [49], [56], [69], [110], [113]	
G-measures	[41], [42]	
Hit rate	[72], [74]	
NDCG normalized discounted cumulative gain)	[3], [6], [17], [21], [28], [50], [51], [71], [72], [74]	
MAP (mean average precision)	[28], [50], [80]	
AUC (average area under curve)	[1], [16], [71]	
Coverage	[1], [4], [80]	
Error measures	**Mean absolute error (MAE)**	[1], [4], [27], [43], [51], [53], [58], [59], [63], [68], [76], [77], [90], [102], [106], [112]
	Mean square error (RMSE)	[1], [43], [51], [63], [68], [69], [76], [77], [90], [102], [106]
	Normalized Mean Absolute Error (NMAE)	[58], [59], [82]
Other evaluations	[87][98][73]	
Methods or Tools Used by RSs Who Consider Social Relacions Information		
Memory-based CF	**Similarity among users**	[4], [42], [54], [56], [58]–[60], [72], [76], [80], [82], [86]–[88], [91], [98]–[100], [110]
	Predict rating	[20], [23], [35], [45], [48], [56], [59], [61], [98]
	Sentimental analysis	[97], [105]
Model-based CF	**Factorization matrix**	[2], [3], [6], [17], [19], [21], [27], [31], [36], [43], [49]–[51], [57], [61], [64], [65], [68], [69], [71], [74], [76], [77], [84], [90], [103], [106]
Network-based recommendation models	**Random Walk**	[14], [15], [29], [38], [78], [113]
	PageRank	[62], [64], [111]
	Hyperlink-Induced Topic Search (HITS)	[29], [33], [41]
	Louvain modularity algorithm	[62]
	K-means	[89]
	Latent Bias Model (LBM)	[95], [107], [108]
	Association Rules	[32], [94]
Recommender Systems Applications that Consider Social Relations Information		
Social context	**Brightkite**	[15], [16], [28], [29], [60], [78], [80]
	Book-Crossing	[69]
	Ciao	[65]
	Delicious	[2], [76]
	Douban	[20], [36], [50], [64], [74], [77]
	Epinions	[4], [36], [38], [46], [50], [57], [61], [62], [64]–[66]
	Facebook	[73], [86], [88], [94], [104]
	FilmTrust	[59]
	Flixster	[4], [74]
	Foursquare	[14], [16], [23], [29], [30], [33], [60], [72], [78], [103], [113]
	Gowalla	[15], [16], [23], [28]–[32], [60], [78], [80], [103]
	Renren	[106]
	Meetup	[6], [17], [19]–[22], [104]
	MovieLens	[3], [69], [76], [112]
	Lastfm	[3], [58], [76]
	QWS	[82]
	Yelp	[27], [28], [43], [77]
	Weeplaces	[31]
	Weibo	[68], [71], [90], [106]
Health context	[47], [87]	
Educational context	[42], [84], [100], [111]	
General computing context	[25], [26], [73], [82], [86], [88], [91], [95], [98], [99], [107], [108]	

information such as latitude and longitude of the location, time, and review, among others. The recommender systems used in these social networks recommend points of interest according to the use of information provided by the check-ins; such as considering the check-ins made by the user's friends [12], [23], [28]–[33] or the frequent check-ins [13]–[15], [34]. Another way is according to the popularity of a place, by estimating an average check-in count of friends [31] or by limiting the geographic area concerning the friends' domicile and locations visited [34]. In contrast, in [16], they do not recommend POIs but instead friends, considering information such as frequency and proximity (location) of check-ins.

In turn, in Event-Based Social Network (EBSN), users can create, promote, and share social events. In this sense, referral systems suggest events on different topics to users according to their tastes, proximity, and time [17]. When a user is invited to an event, he or she has a pending request (RSVPed), in [6], [18], [19] the authors use the amount of RSVPed that friends have as information to recommend events. In [20], [21], they assume that the participation of a friend in the event can influence the user, as well as, if a user often participates in events organized by an organizer, the probability that he or she will attend other similar events organized by the organizer will be high. Besides, the average rating and contents of events held are considered [22].

Previously, activities performed by users on social networks were mentioned that the researchers assume "influence" in the recommendations. The following are activities or actions carried out by users, in a general sense, that the authors use as information in their work proposals:

- The use of friends' short- and long-term preferences (or interests), with the short term being the last interest or preference seen in the previous browsing session and long term being an average of the importance [24].
- Evaluating the actions of friends, considering them as actions to copy, create and follow personal processes, is understood as an individual process to a set of tasks with a specific order to achieve an objective [25].
- Evaluating the curiosity among friends, according to the evaluation of the item, for seeing if it has an impact on the recommendation. That is, whether it has a high rating, a film, rated by a friend, can cause curiosity in the target user [26].

3. Interactions Between Users

The concept of social networks is commonly seen as interactions between individuals, and they have become very popular in recent times [111]. In general, social networks allow the modeling of social interactions and relationships. The connecting relationship between users in a social network is often referred to as "virtual friendship" [67]. Each user is a node, and each link is an interaction or relationship [62]. The following is a description of some of the interactions found in this work:

- According the type of social connection, the relationship of friendship or of followers/followers [3], [7], [15], [34], [67]–[83], of students [84], [111] or of employees [85].
- list of friends [86].
- common friends [72], [87]–[91].

According to their actions:

- the number of "likes" recorded by the user an in user publications [88].
- the number of common activities (e.g., liked/followed/tagged the same blog/wiki/community) performed [3], [92].
- the number of reactions or comments to published content among users [93], [94].
- sending and receiving tweets between two users [30] [31].
- the number of tweets sent to friends [95].

According to the analysis of interactions:

- analysis of similarity by means of tags between users [96].
- analysis of feelings of comments between users [32][97].
- analysis of feelings about the content of user interactions, (retweet, mentions, likes, comments, etc.) [97].
- analysis of closeness between two people with information provided by mobile devices, in which they adopt records of calls, messages and, in addition, publications on social networks [98], [99].
- proximity analysis based on the number of interactions (messages) and preferred content between two users with respect to date and time [100].
- the proximity analysis between attendees and presenters with respect to frequency and time of attendance at presentations [101].
- analyze the relationship between the user who wrote a review of an article and the user who comments on it [102].
- calculation of the implicit relationships of two users who registered at the same place within a time interval, using two independent approaches, diversity being the number of times they coincided in different places and frequency being the number of times they coincided in the same place [103].

B. Methods or Tools Used by RSs That Consider Social Relations Information

Concerning models, recommender systems are divided into two main categories: collaborative filtering (CF) and content-based filtering (CBF). The collaborative filtering approach is the most widely used, divided into two methods: memory-based and model-based [1]. Both models reported in the analysis are described below:

1. Memory-based CF

Memory-based algorithms are subdivided in two ways: user-based CF and item-based CF [1]. However, the analysis of this work is focused on the similarity between users. The methods most used are the a) calculate the similarity between users and b) predicting rating.

Regarding the a) calculate the similarity between users, it is obtained by comparing the ratings of two users. The classical measures to calculate the similarity between users are the cosine similarity [86], [87], [91], the correlation coefficient of Pearson [4], [76], [82], [98] and Jaccard [42]. In turn, the authors incorporate additional information such as the friendship relationship between users [80], the degree of connection between users according to common friends [87] and the amount of "likes" between them [88], the number of interactions [99], common interests [100] or check-ins between friends [72], [99]. Another implementation is to generate the similarity function using a genetic algorithm model, in this case, using social variables (age, gender, educational background, and relationship status) of the user and his friends [104]. About the b) predict rating; likewise, social influence among users is calculated by the closeness between two users a and b; and user activity a [20], [23], [98] or by the user's reputation (number of users with confidence in user a) [48]. In this way, the final value of influence is used as a parameter in the rating prediction function. Accordingly, the confidence value is calculated according to the level of trust or distrust [61], by the popularity and commitment of a user concerning the rest [35] or according to the level of trust between user interactions for the time [45]. Also, using sentimental analysis of user tweets, assuming that the emoticons describe negative and positive facial expressions; sentimental similarities are computed by Karhunen-Loeve (KL) transform [97], or the implementation of standard algorithms such as K-nearest neighbor algorithm ID3 to classify feelings into positive or negative [105].

2. Model-based CF

Memory-based recommender systems are easy to implement and understand their algorithms. However, these systems are not practical when dealing with large numbers of users and items. Model-based recommender systems arise later to avoid this drawback [112]. These RSs require a prior learning phase to determine the model's optimal parameters before making a recommendation. Once this phase is complete, RSs can quickly predict user ratings [1]. Within this model-based CF, there are grouping models, network-based models, Bayesian models, and linear factor models, such as the factorization matrix [51]. The MF model allows for the modeling of the intrinsic characteristics of each user and each item. The latent factor model can be divided into two types: basic factorization matrix (MF) that only uses rating matrix to make predictions, and those that add information to MF that are content information, social information, and context information [76]. The following is a list of additional information that researchers incorporate into their proposals: user attributes [43], user and event characteristics [6], [17], [19], [21], check-ins between users [31], social connections (friend, follower/follower) between users [2], [3], [68], [69], [71], [74], [77], [84], [106], trust [36], [49]–[51], [61], [64], [65], [103], implicit trust (according to places visited in common) [103] , affinity [76], distrust [51], [57], [61], item categories and keywords [90], tags between users and user interactions [3].

3. Network-based Recommendation Models

Social networks are modeled as a network with user nodes and items connected by edges that describe the relationships. In this sense, they apply models such as random walk for a recommendation of friends [29], points of interest (POI) [14], [15], [78], [113], products and services in electronic commerce [38]. In addition, recommender systems use ranking algorithms to evaluate items such as PageRank [62], [64], [111], Hyperlink-Induced Topic Search (HITS) [29], [33], [41]. Also, community detection algorithms, like the Louvain modularity algorithm [62], K-means [89]. On the other hand, in [95], [107], [108], they propose a Latent Bias Model (LBM). LBM allows the use of appropriate terms to capture the importance of different characteristics for prediction, such as the friendship relationship. The use of deep learning has been successfully applied in the field of RH, achieving a significant improvement over traditional models [12]. Also, in [32], [94], they implement data mining to generate association rules for obtaining suggestions.

C. Evaluation of Recommender Systems That Implement This Type of Information.

The authors use several metrics to validate their research. Generally, they use datasets to evaluate the performance of the recommender system. However, Table II summarizes the metrics used. The most used metrics are described below.

1. Accuracy

Accuracy is a well-known and used metric in the field of Artificial Intelligence; it measures the closeness of a measurement to the real value given by a system [41], [42]. The ACC is the fraction of all correctly classified instances, and it is a metric used to measure the classification quality of a classifier [19]. The accuracy in a recommender system is determined by the number of satisfactory recommendations for the number of possible recommendations [41] (see Eq. 1). For example, in [91], they measure accuracy by first asking users to report the areas they expected to visit in the context of a shopping trip. They recommend the route and record the areas they attended.

$$accuracy = \frac{number\ of\ successful\ recommendations}{number\ of\ possible\ recommendations} \tag{1}$$

2. Recall

The Recall metric is a metric used in the Recovery of Information Retrieval (IR), in the field of user referral systems it is essential to receive recommendations in an orderly manner, from best to worst. The metric recall is the ability to obtain all satisfactory recommendations present in the pool [41]. For example, in [44], to evaluate the buddy recommender system, the recall metric corresponds to the number of trusted friends returned by the method compared to the total number of actual trusted friends for each user.

3. Precision

Like recall, accuracy is a metric used in IR, by which, it describes the fraction or proportion of instances received that are relevant [42]. Of the articles analyzed, the precision metric is the most used. For example, in [97], the accuracy in the Top-k recommendation is defined as the ratio of the number of relevant users to the number of recommended users for a given k (see Eq. 2). In [44], the accuracy is the number of actual trusted friends that were returned by the system compared to the total number of returned friends. However, the precision value depends on the size of the list, k value [72].

$$P@k = \frac{number\ of\ relevant\ users\ of\ top\ k\ users}{k} \tag{2}$$

4. F-measures and G-measures

F-measures defines the harmonic mean of the precision and recall metric [42]. That is, they try to join the precision and recall values in a single value. It is essential to evaluate the conjunction of these two metrics since optimizing one metric by decreasing the other is unaffected. The parameter β allows weighting the precision and recall in different ways. By varying β, it is possible to obtain different values of these metrics [41] (see Eq. 3). F-measures refers to true positives to the arithmetic mean of expected and actual positives. On the other hand, the geometric mean (G-measures) of recall and precision effectively normalizes the true positives to the geometric mean of the predicted and actual positives [3], [41].

$$F_{\beta} = \frac{precision\ .\ recall}{(1-\ \beta).precision +\ \beta\ .recall} \tag{3}$$

5. Hit Rate

Hit rate shows the proportion of users who are given at least one true recommendation. It is a metric that is independent of the size of the output list. It is 1.0 if the output list contains at least one true recommendation, otherwise 0.0 [72].

6. Error Measures

The error rate can be reduced to a single evaluation metric by taking the mean absolute error (MAE) or the root mean squared error (RMSE). Whatever the error metric is used, the main objective is to reduce this error and try to generate customized lists that the user will consume and give an excellent rating of the recommended items [63]. Therefore, a small value of MAE (see Eq. 4) or RMSE (see Eq. 5) means high accuracy in the recommendation [90], [102], [106].

$$MAE = \frac{\sum_{(i,j)\in}|R_{ij} - r_{ij}|}{|T|} \tag{4}$$

$$RMSE = \sqrt{\frac{1}{|T|}\sum\nolimits_{(i,j)\in T}(R_{ij} - r_{ij})^2} \tag{5}$$

Other metrics used in top-n recommendations are nDCG (normalized discounted cumulative gain) [6], [17], [21], [28], [71], [72], MAP (mean average precision) [28] both metrics consider the ranking (or range) of the recommendations. Another metric used is

AUC (average area under the curve) [16], [71]. The higher the NDCG value, the better the ranking list [71].

7. Other Evaluations

To evaluate the recommendation of friends in [87], the authors studied the system over a period. A total of 1787 recommendations were generated, of which 258 (14.44%) were friend requests, and 63% of 258 were accepted. Another way is to analyze user feedback to evaluate user satisfaction with the recommended ads [98]. In [73], they evaluated the recommender system using a group of users who interacted with a television program recommender system to analyze its performance. They assume that the use of social relationship information provided by Facebook showed that the initial results increased accuracy to 10%.

D. Recommender Systems Applications That Consider Social Relations Information

1. Social Context

Location-based social networks (LBNS) allow users to check-in, rate, and comment on their experiences in a place. In analyzing the articles, the authors evaluate their proposals on datasets such as Gowalla [28], [30]–[32], Brightkite [15], [16], [28], [29], [78], [80], Foursquare [14], [28], [30], [72], Yelp [28], [43], [77], Weeplaces [31]. The recommender systems used by these online networks allow for the recommendation of points of interest (POIs) in a general sense (hotels, restaurants, activities, among others). However, in [32], they recommend places to shop (shops in shopping) or popular activities as a point of interest [29]. Like the LBNS, the authors use online opinion networks to evaluate their systems. Social networks such as Epinions or Douban allow the user to comment on articles (such as films, books, cars, software, etc.). Of the works analyzed, they use Epinions [36], [38], [46], [57], [61], [62], [65] or Doubans [20], [36], [64], [74], [77], these social networks allow users to make ratings, and besides, allows the user to generate their network of trust. In [57], they state that because of the explicit relationships of trust and mistrust, Epinions is appropriate to study trust-based recommender systems.

Event-based social networks (EBNS) are online social networks where users can create, promote, and share social events with other users [21]. Meetup allows us to join a group on a topic and organize events. From the analysis made, the works [6], [17], [19]–[22], [104] Meetup to test their proposals.

In addition to the above, the systems suggest friends to the user according to the following considerations: sentimental analysis of comments [105], tweets between users [97], confidence analysis[41], [44], [52], [66], number of common friends [89], friendly relations [71], the current location of the user [16], [29] or suggestion of accompaniment for activities [23].

On the other hand, the recommendation of contexts according to the degree of trust between users. That is, if the user "a" has confidence with user "b," and the user likes movies, the system recommends movies to user "a" [64]. Another implementation is to propose activities and places according to social groups, previous choices, and location of the user [45].

2. Health Context

The analysis carried out shows a recommender system, in which friends with similar characteristics of cardiovascular problems are analyzed to provide a set of recommendations or suggestions regarding health information [87]. Another application is the recommendation of dental professionals according to a confidence analysis they provide [47].

3. Educational Context

Studies have described a system for recommending teamwork according to student interactions [111], a method for choosing a professional career [42], and another for recommending courses, tasks or exercises to improve student performance [84], or educational resources (e.g., YouTube videos) according to the student's profile [100].

4. General Computing Context

A mashup recommender system is used (integrator and reuse of web services), based on social information that allows web services to be searched [82], [88], arguing that social relations and similar interests among users allow services to be easily recommended. Another application is the recommendation of privacy options in social networks, considering the user's list of friends and friends of friends [86].

Another work focuses on recommendation of mobile applications based on user interactions, considering expert users as contacts in the social network of the user with the highest number of installed applications [99]. Other works recommend TV programs (Electronic Program Guide) [73], movies [26], shopping routes [91], advertising [98], personal processes such as tasks to achieve a goal [25] or multimedia content [95], [107], [108].

In the following section, we present the main conclusions and contributions of this systematic literature review.

IV. Conclusions

The objective of this work was to investigate and analyze what type of information on the existing relationships between users is considered in recommender systems. For this reason, the systematic review is oriented more specifically to this analysis, and a more general systematic review of recommender systems was not carried out. The analysis concluded that the existing methods in recommender systems do not consider the relationships between groups of people if a product/service is recommended. For example, a movie is recommended to you because you will watch it with your father. Also, we consider that the information provided by this study can be valuable to increase the precision in the recommendations, and this can be the principal value of this work for the collaborative filtering area.

- *RQ1: Which social relations information is considered in the construction of recommender systems?* The authors argue that the social relationships between users used as information help to improve recommendations. To obtain a better analysis, it was classified into three main groups. However, these groups are related to each other.
    - Trust: The concept of trust is widely used to describe the degree of a relationship between two people. Trust relationships between users allow for improved efficiency in RSs; however, the trust used as information varies according to the author. The value of trust can be explicit or implicit. It is understood as an exact value when the user chooses his trust (or mistrust) persons. The Epinions social network allows us to generate a list of trusted and untrusted users. Another way to calculate trust is implicit, the degree of trust is calculated by the behavior (check-ins, activities, interactions between users) of the user concerning others in a social network.
    - Friend activities: People are often influenced by their friends and tend to do similar activities such as visiting places their friends have already visited or attended similar events. Besides, people often do not do activities alone, but do them in the company. Thus, the authors use these friend activities as influencing factors in the referral process. This information is usually provided by location and event-based social networks, such as Foursquare, Gowalla, Yelp, and Meetup.

  - User interactions: are widely used by authors as additional information for the recommendation process, or to obtain trust value. Interaction between users is understood as the action or behavior between users in a social network. Being the number of likes, followers/followers, comments, publications, some examples found. However, the relationships of friendship or common friends are the most used. However, the works that consider friendship relationships do not specify the type of relationship or bond (married, engaged, divorced, or other describing a family relationship) due to security issues or lack of information in the datasets used. In [84], [85], [111], they specify a type of relationship between users, such as employee-employer or student-teacher. Therefore, considering the type of relationship between users could further improve the accuracy of recommendation systems.

- *RQ2: What methods or tools are used by RSs that consider social relations information?* RSs based on collaborative filtering use social relationships as additional information. Within CF, we find two models, memory-based and model-based, widely accepted with their advantages and disadvantages. The Nearest Neighbors technique is mainly used to obtain the most similar K users, considering the information detailed in RQ1. This technique employs metrics such as cosine similarity, Pearson correlation, and Jaccard index. Model-based HR techniques start from a previous learning phase to determine the optimal parameters of the model before making a recommendation. Once this phase is completed, RSs can quickly predict user ratings. The most widely used technique is the factorization matrix (FM) because of its high recommendation accuracy and efficiency. FM allows the modeling of the latent characteristics of each user and each item. However, with the constant growth of information, this model presents difficulties in terms of accuracy when suggesting content. Nevertheless, the authors incorporate into the model additional information detailed in RQ1 such as trust (or mistrust) relationships, social influence, user's own characteristics, among others.
- *RQ3: How are recommender systems that employ social relations evaluated?* Various metrics are used to evaluate the efficiency of RSs, such as accuracy, recall, and precision. Accuracy and recall metrics are used to measure the efficiency of recommendations. Accuracy measures the closeness of a measurement to the real value given by a system. Recall takes care of checking what percentage of the items significant to a user were recommended to him. Error measures such as Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) are the most commonly used. It calculates the absolute distance between the suggestions made, and the real rating of the user, considering high performance and a low error value. Other metrics, such as Area Under Curve (AUC), evaluate the quality of the recommendations.
- *RQ4: Which RS applications use social relations information?* The authors use social network datasets to experiment with their proposals. According to the analysis made, datasets such as Epinions and Douban allow the user to generate their own list of trusted and/or distrusted users. Also, location and event-based networks are widely used due to the influence that friends of the user can have in visiting places or attending social events—taking into account information such as current location (latitude, longitude), check-in, or event, as well as the time factor. However, this review determined the clever use of recommendation systems that employ social relationships as additional information.

However, the problems of data sparsity, and the cold start was driven by the constant growth of information has become a complex problem in obtaining accurate recommendations. Today, authors not only consider the historical behavior of the user but also find contextual information of the user, such as trust relationships, friendship relationships, activities of friends, time information, location. Consequently, many studies implement recommendation models that integrate various social factors to improve their efficiency and alleviate problems of data sparsity and cold start. In contrast, the computational cost is increasing and complex.

We found studies that show that their social circle influences people in decision making. Each person influences to a greater or lesser extent according to the underlying context. In this sense, it is interesting to know in what context this happens; that is, a parent's opinion has a more significant (or lesser) influence than that of a friend in a family decision.

This review has shown how papers employ this strategy in the field of recommender systems. This is due to the constant growth of information generated by social applications. For example, if a person has a friendship relationship with another person, they may share the same tastes; in this sense, we obtain relevant, valuable information in recommender systems to improve their suggestions by applying this hypothesis.

This work provides a general classification of the types of social relationship information. Each information was identified, analyzed, and classified to obtain a better interpretation. In turn, in each group, there is a subclassification to improve their interpretation. Similarly, the methods implemented by the recommender systems for this type of information were identified and analyzed. Likewise, this work describes the different metrics used by researchers to evaluate their recommender system proposals. Finally, a classification of the applications that consider social relationship information according to the contexts (social, health, education, general) is presented.

In this sense, this work provides comprehensive information on the use of social relation information in the field of recommender systems. An exhaustive work has been carried out, following a consolidated methodology. So, this work covers fundamental pillars for future work, such as identifying the information, the methods that use it, its evaluation, and the field in which it is applied.

As future works, we will carry out a study applied to specific contexts such as academic, tourism, social. The aim is to identify which information is relevant according to the applied field and to propose new proposals for precision improvements based on the results obtained. Besides, the results of this systematic review have allowed generating suggestions about some of the recommendation methods studied in a mobile application. That is, using information about a specific user (user profile) and, mainly, adding information about their social link (friend, brother, married, brother-in-law, boyfriend, etc.) with other people. It is also intended to extend this prototype to different contexts, bearing in mind that the user's social links are closely linked to the scope of application. Also, we will analyze how our proposals behave using the evaluation metrics identified in this work. The aim is to obtain results that minimize problems such as cold start or lack of data.

Acknowledgment

The authors thank CONICET (Consejo Nacional de Investigaciones Científicas y Técnicas de Argentina) and AUIP (Asociación Universitaria Iberoamericana de Postgrado) for their support to the doctoral research.

References

[1] R. Chen, Q. Hua, Y. S. Chang, B. Wang, L. Zhang, and X. Kong, "A survey

of collaborative filtering-based recommender systems: from traditional methods to hybrid methods based on social networks," *IEEE Access*, vol. 6, no. c, pp. 64301–64320, 2018, doi: 10.1109/ACCESS.2018.2877208.

[2] Z. Sun *et al.*, "Recommender systems based on social networks," Journal of Systems and Software., vol. 99, pp. 109–119, 2015, doi: 10.1016/j.jss.2014.09.019.

[3] J. Zhou *et al.*, "Social network and tag sources based augmenting collaborative recommender system," *IEICE Transactions on Information and Systems*, vol. 98, no. 4, pp. 902–910, 2015.

[4] P. Moradi and S. Ahmadian, "A reliability-based recommendation method to improve trust-aware recommender systems," *Expert Systems with Applications*, vol. 42, no. 21, pp. 7386–7398, 2015, doi: 10.1016/j.eswa.2015.05.027.

[5] E. Ashley-Dejo, S. Ngwira, and T. Zuva, "A survey of Context-Aware Recommender System and services," *2015 Int. Conf. Comput. Commun. Secur. ICCCS 2015*, 2016, doi: 10.1109/CCCS.2015.7374144.

[6] Y. Liao, X. Lin, and W. Lam, "Exploring influence among participants for event recommendation," *Proc. 2016 IEEE/ACM Int. Conf. Adv. Soc. Networks Anal. Mining, ASONAM 2016*, pp. 417–420, 2016, doi: 10.1109/ASONAM.2016.7752268.

[7] S. Diego, P. Javier, A. B. Gil, L. Vivian, and N. Moreno-garc, "Distributed Computing and Artificial Intelligence, Special Sessions, 15th International Conference," vol. 801, pp. 267–274, 2019, doi: 10.1007/978-3-319-99608-0.

[8] B. Kitchenham and S. Charters, "Guidelines for performing Systematic Literature reviews in Software Engineering Version 2.3," *Engineering*, vol. 45, no. 4ve, p. 1051, 2007, doi: 10.1145/1134285.1134500.

[9] C. Wohlin, "Guidelines for Snowballing in Systematic Literature Studies and a Replicationin Software Engineering," 2014, doi: 10.1145/2601248.2601268.

[10] A. García-Holgado, S. Marcos-Pablos, and F. García-Peñalvo, "Guidelines for performing Systematic Research Projects Reviews," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 6, no. 2, p. 9, 2020, doi: 10.9781/ijimai.2020.05.005.

[11] A. García-Holgado and F. J. García-Peñalvo, "Mapping the systematic literature studies about software ecosystems," *ACM Int. Conf. Proceeding Ser.*, pp. 910–918, 2018, doi: 10.1145/3284179.3284330.

[12] Y. Tan, Q. T. Eds, and D. Hutchison, *Data Mining and Big Data*, vol. 10387. 2017.

[13] R. Baral, S. S. Iyengar, and T. Li, "CLoSe: Contextualized Location Sequence Recommender," *[RecSys2018]Proceedings 12th ACM Conf. Recomm. Syst.*, pp. 470–474, 2018, doi: 10.1016/j.ecoenv.2014.11.008.

[14] F. Mourchid, J. Ben Othman, A. Kobbane, E. Sabir, and M. El Koutbi, "A Markov chain model for integrating context in recommender systems," *2016 IEEE Glob. Commun. Conf. GLOBECOM 2016 - Proc.*, pp. 0–5, 2016, doi: 10.1109/GLOCOM.2016.7841514.

[15] M. Ye, P. Yin, and W.-C. Lee, "Location {R}ecommendation for {L}ocation-based {S}ocial {N}etworks," *Proc. SIGSPATIAL GIS*, no. Iccc, 2010.

[16] J. C. Valverde-Rebaza, M. Roche, P. Poncelet, and A. de A. Lopes, "The role of location and social strength for friendship prediction in location-based social networks," *Information Processing & Management.*, vol. 54, no. 4, pp. 475–489, 2018, doi: 10.1016/j.ipm.2018.02.004.

[17] Y. Liao, X. Lin, and W. Lam, "Whether this participant will attract you to this event? Exploiting Participant Influence for Event Recommendation," *Proc. - IEEE Int. Conf. Data Mining, ICDM*, pp. 1035–1040, 2017, doi: 10.1109/ICDM.2016.149.

[18] Y. Gu, J. Song, W. Liu, L. Zou, and Y. Yao, "Context Aware Matrix Factorization for Event Recommendation in Event-Based Social Networks," *Proc. - 2016 IEEE/WIC/ACM Int. Conf. Web Intell. WI 2016*, pp. 248–255, 2017, doi: 10.1109/WI.2016.0043.

[19] Y. Gu, J. Song, W. Liu, L. Zou, and Y. Yao, "CAMF: Context Aware Matrix Factorization for Social Recommendation," *Web Intelligence*, vol. 16, no. 1, pp. 53–71, 2018, doi: 10.3233/WEB-180373.

[20] M. Wei and D. Wang, "CPERS: Contextual and personalized event recommender system," *Proc. - 2016 Int. Conf. Comput. Sci. Comput. Intell. CSCI 2016*, pp. 421–426, 2017, doi: 10.1109/CSCI.2016.0086.

[21] A. Q. Macedo, C. Grande, and C. Grande, "Context-Aware Event Recommendation in Event-based Social Networks Categories and Subject Descriptors," *ACM Conf. Recomm. Syst. RecSys 2015*, pp. 123–130, 2015, doi: 10.1145/2792838.2800187.

[22] Z. Wang, Y. Zhang, Y. Li, Q. Wang, and F. Xia, "Exploiting social influence for context-aware event recommendation in event-based social networks," *Proc. - IEEE INFOCOM*, 2017, doi: 10.1109/INFOCOM.2017.8057167.

[23] W. Tu, D. W. Cheung, N. Mamoulis, M. Yang, and Z. Lu, "Activity Recommendation with Partners," *ACM Transactions on the Web*, vol. 12, no. 1, pp. 1–29, 2017, doi: 10.1145/3121407.

[24] Y. Wang and L. Charlin, "Session- Based Social Recommendation via Dynamic Graph Attention Networks," pp. 555–563, 2019.

[25] A. H. H. Ngu, S. Fang, and H. Paik, "Social-PPM: Personal experience sharing and recommendation," *Proc. - 2016 IEEE 2nd Int. Conf. Collab. Internet Comput. IEEE CIC 2016*, pp. 29–36, 2017, doi: 10.1109/CIC.2016.16.

[26] Q. Wu, S. Liu, C. Miao, Y. Liu, and C. Leung, "A Social Curiosity Inspired Recommendation Model to Improve Precision, Coverage and Diversity," *Proc. - 2016 IEEE/WIC/ACM Int. Conf. Web Intell. WI 2016*, pp. 240–247, 2017, doi: 10.1109/WI.2016.0042.

[27] K. Gao *et al.*, "Exploiting Location-Based Context for POI Recommendation When Traveling to a New Region," *IEEE Access*, vol. 8, pp. 52404–52412, 2020, doi: 10.1109/ACCESS.2020.2980982.

[28] C. Yang, L. Bai, C. Zhang, Q. Yuan, and J. Han, "Bridging Collaborative Filtering and Semi-Supervised Learning," *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. - KDD '17*, pp. 1245–1254, 2017, doi: 10.1145/3097983.3098094.

[29] H. Bagci and P. Karagoz, "Random walk based context-aware activity recommendation for location based social networks," *Proc. 2015 IEEE Int. Conf. Data Sci. Adv. Anal. DSAA 2015*, pp. 531–536, 2015, doi: 10.1109/DSAA.2015.7344852.

[30] S. Zhang and H. Cheng, "Exploiting context graph attention for POI recommendation in location-based social networks," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 10827 LNCS, pp. 83–99, 2018, doi: 10.1007/978-3-319-91452-7_6.

[31] R. Baral and T. Li, "Exploiting the roles of aspects in personalized POI recommender systems," *Data Mining and Knowledge Discovery*, vol. 32, no. 2, pp. 320–343, 2018, doi: 10.1007/s10618-017-0537-7.

[32] D. Pierzchała, *Application of ontology and rough set theory to information sharing in multi-resolution combat m&s*, vol. 551. 2014.

[33] Y. Ying, L. Chen, and G. Chen, "A temporal-aware POI recommendation system using context-aware tensor decomposition and weighted HITS," *Neurocomputing*, vol. 242, pp. 195–205, 2017, doi: 10.1016/j.neucom.2017.02.067.

[34] C. Rios, S. Schiaffino, and D. Godoy, "A study of neighbour selection strategies for POI recommendation in LBSNs," *International Journal of Molecular Sciences*, vol. 44, no. 6, pp. 802–817, 2018, doi: 10.1177/0165551518761000.

[35] S. Nepal, S. K. Bista, and C. Paris, "Behavior-Based Propagation of Trust in Social Networks with Restricted and Anonymous Participation," *Computational Intelligence*, vol. 31, no. 4, pp. 642–668, 2015, doi: 10.1111/coin.12041.

[36] T. Wang, X. Jin, X. Ding, and X. Ye, "User Interests Imbalance Exploration in Social Recommendation," *Proc. 23rd ACM Int. Conf. Conf. Inf. Knowl. Manag. - CIKM '14*, pp. 281–290, 2014, doi: 10.1145/2661829.2662043.

[37] G. Xu, L. He, and M. Hu, "Document Context-Aware Social Recommendation Method," *2019 Int. Conf. Comput. Netw. Commun. ICNC 2019*, pp. 787–791, 2019, doi: 10.1109/ICCNC.2019.8685666.

[38] X. Li, X. Yan, L. Xie, and C. Men, "Exploiting user's social network: A novel method to recommend most attractive and targeted service," *Proc. 2015 4th Int. Conf. Comput. Sci. Netw. Technol. ICCSNT 2015*, no. Iccsnt, pp. 536–542, 2016, doi: 10.1109/ICCSNT.2015.7490805.

[39] B. Pal, S. Banerjee, and M. Jenamani, "Threshold-Based Heuristics for Trust Inference in a Social Network," *2018 11th Int. Conf. Contemp. Comput. IC3 2018*, no. 1, pp. 1–7, 2018, doi: 10.1109/IC3.2018.8530496.

[40] P. Yu, "Combine Trust and Interest Similarity for Enhanced-Quality Recommendations," *Proc. - 9th Int. Conf. Inf. Technol. Med. Educ. ITME 2018*, pp. 740–744, 2018, doi: 10.1109/ITME.2018.00168.

[41] G. Carullo, A. Castiglione, A. De Santis, and F. Palmieri, "A triadic closure and homophily-based recommendation system for online social networks," *World Wide Web*, vol. 18, no. 6, pp. 1579–1601, 2015, doi: 10.1007/s11280-015-0333-5.

[42] P. Verma, S. K. Sood, and S. Kalra, "Student career path recommendation in engineering stream based on three-dimensional model," *Computer Applications in Engineering Education*, vol. 25, no. 4, pp. 578–593, 2017, doi: 10.1002/cae.21822.

[43] J. D. Zhang, C. Y. Chow, and J. Xu, “Enabling kernel-based attribute-aware matrix factorization for rating prediction,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 4, pp. 798–812, 2017, doi: 10.1109/TKDE.2016.2641439.

[44] A. Kalai, A. Wafa, C. A. Zayani, and I. Amous, “LoTrust: A social Trust Level model based on time-aware social interactions and interests similarity,” *2016 14th Annu. Conf. Privacy, Secur. Trust. PST 2016*, pp. 428–436, 2016, doi: 10.1109/PST.2016.7906967.

[45] M. Frikha, M. Mhiri, M. Zarai, and F. Gargouri, “Using TMT ontology in trust based medical tourism recommender system,” *Proc. IEEE/ACS Int. Conf. Comput. Syst. Appl. AICCSA*, vol. 0, 2016, doi: 10.1109/AICCSA.2016.7945768.

[46] J. P. Mei, H. Yu, Z. Shen, and C. Miao, “A social influence based trust model for recommender systems,” *Intelligent Data Analisys*, vol. 21, no. 2, pp. 263–277, 2017, doi: 10.3233/IDA-150479.

[47] S. Pradhan, V. Gay, and S. Nepal, “Improving dental care recommendation systems using trust and social networks,” *2014 IEEE Int. Conf. Commun. ICC 2014*, pp. 4264–4269, 2014, doi: 10.1109/ICC.2014.6883990.

[48] H. Zhou, Q. Li, and F. Zhou, “Trust-aware collaborative filtering recommendation in reputation level,” *Proc. 2017 IEEE 2nd Adv. Inf. Technol. Electron. Autom. Control Conf. IAEAC 2017*, pp. 2452–2457, 2017, doi: 10.1109/IAEAC.2017.8054464.

[49] G. Guo, J. Zhang, F. Zhu, and X. Wang, “Factored similarity models with social trust for top-N item recommendation,” *Knowledge-Based Systems*, vol. 122, pp. 17–25, 2017, doi: 10.1016/j.knosys.2017.01.027.

[50] B. Yang, Y. Lei, D. Liu, and J. Liu, “Social collaborative filtering by trust,” *IJCAI Int. Jt. Conf. Artif. Intell.*, vol. 39, no. 8, pp. 2747–2753, 2013, doi: 10.1109/TPAMI.2016.2605085.

[51] R. Forsati, M. Mahdavi, M. Shamsfard, and M. Sarwat, “a17-Forsati.Pdf,” vol. 32, no. 4, 2014.

[52] R. Cheng, B. Cui, A. W. Conference, and D. Hutchison, *LNCS 9313 - Web Technologies and Applications*. 2015.

[53] J. Li and R. Yang, “DTCMF: Dynamic Trust-based Context-aware Matrix Factorization for Collaborative Filtering,” pp. 0–5.

[54] H. Liu and N. Jiao, “A hybrid book recommendation algorithm based on context awareness and social network,” *Proc. - 2020 3rd Int. Conf. Adv. Electron. Mater. Comput. Softw. Eng. AEMCSE 2020*, pp. 554–561, 2020, doi: 10.1109/AEMCSE50948.2020.00122.

[55] Y. Yujie, “A Survey on Information Diffusion in Online Social Networks,” *ACM Int. Conf. Proceeding Ser.*, vol. 8, pp. 181–186, 2020, doi: 10.1145/3393822.3432322.

[56] L. Esmaeili, S. Mardani, S. A. H. Golpayegani, and Z. Z. Madar, “A novel tourism recommender system in the context of social commerce,” *Expert Systems with Applications*, vol. 149, p. 113301, 2020, doi: 10.1016/j.eswa.2020.113301.

[57] Y. Zheng, Y. Ouyang, W. Rong, and Z. Xiong, “Multi-faceted Distrust Aware Recommendation,” in *Knowledge Science, Engineering and Management*, 2015, pp. 435–446.

[58] D. Sánchez-Moreno, V. L. Batista, M. D. M. Vicente, Á. L. S. Lázaro, and M. N. Moreno-García, “Exploiting the user social context to address neighborhood bias in collaborative filtering music recommender systems,” *Information*, vol. 11, no. 9, 2020, doi: 10.3390/INFO11090439.

[59] J. Son, W. Choi, and S. M. Choi, “Trust information network in social Internet of things using trust-aware recommender systems,” *International Journal of Distributed Sensor Networks*, vol. 16, no. 4, 2020, doi: 10.1177/1550147720908773.

[60] H. Pan and Z. Zhang, “Research on Context-Awareness Mobile Tourism E-Commerce Personalized Recommendation Model,” *Journal of Signal Processing Systems*, pp. 1–8, 2019, doi: 10.1007/s11265-019-01504-2.

[61] R. Forsati, I. Barjasteh, F. Masrour, A.-H. Esfahanian, and H. Radha, “PushTrust,” *Proc. 9th ACM Conf. Recomm. Syst. - RecSys ’15*, pp. 51–58, 2015, doi: 10.1145/2792838.2800198.

[62] M. Razghandi and S. A. H. Golpaygani, “A Context-Aware and User Behavior-Based Recommender System with Regarding Social Network Analysis,” *Proc. - 14th IEEE Int. Conf. E-bus. Eng. ICEBE 2017 - Incl. 13th Work. Serv. Appl. Integr. Collab. SOAIC 207*, pp. 208–213, 2017, doi: 10.1109/ICEBE.2017.40.

[63] E. Hattab, S. Tedmori, and A. Tahhan, “Boosting Collaborative Filtering by Trust,” *Proc. - 2015 Int. Conf. Dev. eSystems Eng. DeSE 2015*, pp. 120–126, 2016, doi: 10.1109/DeSE.2015.21.

[64] G. Ma, Y. Wang, X. Zheng, and M. Wang, “Leveraging Transitive Trust Relations to Improve Cross-Domain Recommendation,” *IEEE Access*, vol. 6, pp. 38012–38025, 2018, doi: 10.1109/ACCESS.2018.2850706.

[65] H. Liu, Z. Yang, J. Zhang, X. Bai, W. Wang, and F. Xia, “Mining implicit correlations between users with the same role for trust-aware recommendation,” *KSII Transactions on Internet and Information Systems*, vol. 9, no. 12, pp. 4892–4911, 2015, doi: 10.3837/tiis.2015.12.009.

[66] V. Kulkarni and A. S. Vaidya, “Proceedings of the International Conference on Data Engineering and Communication Technology,” vol. 468, 2017, doi: 10.1007/978-981-10-1675-2.

[67] V. Podobnik and I. Lovrek, “Implicit social networking: Discovery of hidden relationships, roles and communities among consumers,” *Procedia Computer Science*, vol. 60, no. 1, pp. 583–592, 2015, doi: 10.1016/j.procs.2015.08.185.

[68] K. Ji and H. Shen, “Using category and keyword for personalized recommendation: A scalable collaborative filtering algorithm,” *Proc. - Int. Symp. Parallel Archit. Algorithms Program. PAAP*, pp. 197–202, 2014, doi: 10.1109/PAAP.2014.40.

[69] C. Xu, “A novel recommendation method based on social network using matrix factorization technique,” *Information Processing & Management*, vol. 54, no. 3, pp. 463–474, 2018, doi: 10.1016/j.ipm.2018.02.005.

[70] M. H. Hussein, “Improve the Quality of Social Community Detection,” no. March, pp. 7–9, 2017.

[71] L. Guo, J. Ma, Z. Chen, and H. Zhong, “Learning to recommend with social contextual information from implicit feedback,” *Soft Computing*, vol. 19, no. 5, pp. 1351–1362, 2015, doi: 10.1007/s00500-014-1347-0.

[72] M. G. Ozsoy, F. Polat, and R. Alhajj, “Time preference aware dynamic recommendation enhanced with location, social network and temporal information,” *Proc. 2016 IEEE/ACM Int. Conf. Adv. Soc. Networks Anal. Mining, ASONAM 2016*, pp. 909–916, 2016, doi: 10.1109/ASONAM.2016.7752347.

[73] E. Al-Mohammed and N. Linge, “A generic, personalized electronic program guide system for accessing multiple online TV providers,” *2016 11th Int. Conf. Internet Technol. Secur. Trans. ICITST 2016*, pp. 291–296, 2017, doi: 10.1109/ICITST.2016.7856715.

[74] H. Bao, L. W. B, and P. Sun, “Advances in Multimedia Information Processing – PCM 2012,” vol. 7674, pp. 630–641, 2012, doi: 10.1007/978-3-642-34778-8.

[75] L. Wu, L. Chen, R. Hong, Y. Fu, X. Xie, and M. Wang, “A Hierarchical Attention Model for Social Contextual Image Recommendation,” *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1, 2019, doi: 10.1109/tkde.2019.2913394.

[76] J. Li, C. Chen, H. Chen, and C. Tong, “Towards Context-aware Social Recommendation via Individual Trust,” *Knowledge-Based Systems*, vol. 127, pp. 58–66, 2017, doi: 10.1016/j.knosys.2017.02.032.

[77] X. Qian, H. Feng, G. Zhao, and T. Mei, “Personalized recommendation combining user interest and social circle,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 7, pp. 1763–1777, 2014, doi: 10.1109/TKDE.2013.168.

[78] D. Dwdu, “Location based recommender system using enhanced random walk model.”

[79] A. Gorrab, F. Kboubi, H. Ben Ghezala, and B. Le Grand, “Towards a dynamic and polarity-aware social user profile modeling,” *Proc. IEEE/ACS Int. Conf. Comput. Syst. Appl. AICCSA*, 2017, doi: 10.1109/AICCSA.2016.7945626.

[80] T. Stepan, J. M. Morawski, S. Dick, and J. Miller, “Incorporating Spatial, Temporal, and Social Context in Recommendations for Location-Based Social Networks,” *IEEE Transactions on Computational Social Systems*, vol. 3, no. 4, pp. 164–175, 2016, doi: 10.1109/TCSS.2016.2631473.

[81] A. Hannech, M. Adda, and H. McHeick, “Cold-start recommendation strategy based on social graphs,” *7th IEEE Annu. Inf. Technol. Electron. Mob. Commun. Conf. IEEE IEMCON 2016*, 2016, doi: 10.1109/IEMCON.2016.7746324.

[82] K. K. Fletcher, “A Method for Dealing with Data Sparsity and Cold-Start Limitations in Service Recommendation Using Personalized Preferences,” *Proc. - 2017 IEEE 1st Int. Conf. Cogn. Comput. ICCC 2017*, pp. 72–79, 2017, doi: 10.1109/IEEE.ICCC.2017.17.

[83] M. Eirinaki, N. Moniz, and K. Potika, “Threshold-bounded influence dominating sets for recommendations in social networks,” *Proc. - 2016 IEEE Int. Conf. Big Data Cloud Comput. BDCloud 2016, Soc. Comput.*

Networking, Soc. 2016 Sustain. Comput. Commun. Sustain. 2016, pp. 408–415, 2016, doi: 10.1109/BDCloud-SocialCom-SustainCom.2016.67.

[84] H. L. Thanh-Nhan, L. Huy-Thap, and N. Thai-Nghe, "Toward integrating social networks into intelligent tutoring systems," *Proc. - 2017 9th Int. Conf. Knowl. Syst. Eng. KSE 2017*, vol. 2017-Janua, pp. 112–117, 2017, doi: 10.1109/KSE.2017.8119444.

[85] S. Kremer-Davidson, I. Ronen, L. Leiba, A. Kaplan, and M. Barnea, "Personal Recommendations for Raising Social Eminence in an Enterprise," *Proc. 2018 Conf. Hum. Inf. Interact. - IUI '18*, pp. 629–639, 2018, doi: 10.1145/3172944.3172956.

[86] A. Srivastava and G. Geethakumari, "A privacy settings recommender system for Online Social Networks," *Int. Conf. Recent Adv. Innov. Eng. ICRAIE 2014*, 2014, doi: 10.1109/ICRAIE.2014.6909142.

[87] E. Lima-Medina, O. Loques and C. Mesquita, ""Minha Saude" a healthcare social network for patients with cardiovascular diseases," 2014 IEEE 3nd International Conference on Serious Games and Applications for Health (SeGAH), 2014, pp. 1-7, doi: 10.1109/SeGAH.2014.7067070.

[88] P. Suppa and E. Zimeo, "A Context-Aware Mashup Recommender Based on Social Networks Data Mining and User Activities," *2016 IEEE Int. Conf. Smart Comput. SMARTCOMP 2016*, 2016, doi: 10.1109/SMARTCOMP.2016.7501672.

[89] P. Tasgave and A. Dani, "Friend-space: Cluster-based users similar post friend recommendation technique in social networks," *Proc. - IEEE Int. Conf. Inf. Process. ICIP 2015*, pp. 658–663, 2016, doi: 10.1109/INFOP.2015.7489465.

[90] J. Pei, V. S. Tseng, P. Conference, and R. Goebel, *Advances in Knowledge Discovery Part I*, no. April. 2013.

[91] Y. M. Li, L. F. Lin, and C. C. Ho, "A social route recommender mechanism for store shopping support," *Decision Support Systems*, vol. 94, pp. 97–108, 2017, doi: 10.1016/j.dss.2016.11.004.

[92] F. Ghaffar, T. S. Buda, H. Assem, A. Afsharinejad, and N. Hurley, "A framework for enterprise social network assessment and weak ties recommendation," *Proc. 2018 IEEE/ACM Int. Conf. Adv. Soc. Networks Anal. Mining, ASONAM 2018*, pp. 678–685, 2018, doi: 10.1109/ASONAM.2018.8508292.

[93] F. Amato, V. Moscato, A. Picariello, and G. Sperli, "Recommendation in Social Media Networks," *Proc. - 2017 IEEE 3rd Int. Conf. Multimed. Big Data, BigMM 2017*, pp. 213–216, 2017, doi: 10.1109/BigMM.2017.55.

[94] S. Utku and C. Eren Atay, "A Mobile Location-Aware Recommendation System," *Proc. Int. Conf. Knowl. Discov. Inf. Retr.*, pp. 176–183, 2014, doi: 10.5220/0005053001760183.

[95] C. Wu, J. Jia, W. Zhu, X. Chen, B. Yang, and Y. Zhang, "Affective Contextual Mobile Recommender System," *Proc. 2016 ACM Multimed. Conf. - MM '16*, pp. 1375–1384, 2016, doi: 10.1145/2964284.2964327.

[96] D. Yang, L. Chen, J. Liang, Y. Xiao, and W. Wang, "Social tag embedding for the recommendation with sparse user-item interactions," *Proc. 2018 IEEE/ACM Int. Conf. Adv. Soc. Networks Anal. Mining, ASONAM 2018*, pp. 127–134, 2018, doi: 10.1109/ASONAM.2018.8508802.

[97] D. Yang, C. Huang, and M. Wang, "A social recommender system by combining social network and sentiment similarity: A case study of healthcare," *Journal Information Science*, vol. 43, no. 5, pp. 635–648, 2017, doi: 10.1177/0165551516657712.

[98] L. F. Lin and Y. M. Li, "A Social Recommendation Mechanism for Enhancing O2O E-Commerce," *Proc. - 2017 6th IIAI Int. Congr. Adv. Appl. Informatics, IIAI-AAI 2017*, pp. 401–406, 2017, doi: 10.1109/IIAI-AAI.2017.99.

[99] D. F. Chamorro-Vela *et al.*, "Recommendation of Mobile Applications based on social and contextual user information," *Procedia Computer Science*, vol. 110, no. 2016, pp. 236–241, 2017, doi: 10.1016/j.procs.2017.06.090.

[100] C. K. Pereira, F. Campos, V. Ströele, J. M. N. David, and R. Braga, "BROAD-RSI – educational recommender system using social networks interactions and linked data," *J. Internet Serv. Appl.*, vol. 9, no. 1, 2018, doi: 10.1186/s13174-018-0076-5.

[101] N. Y. Asabere, B. Xu, A. Acakpovi, and N. Deonauth, "SARVE-2: Exploiting Social Venue Recommendation in the Context of Smart Conferences," *IEEE Transactions on Emerging Topics in Computing*, vol. 6750, no. c, pp. 1–12, 2018, doi: 10.1109/TETC.2018.2854718.

[102] E. Choo, T. Yu, M. Chi, and Y. Sun, "Revealing and incorporating implicit communities to improve recommender systems," *Proc. fifteenth ACM Conf. Econ. Comput. - EC '14*, pp. 489–506, 2014, doi: 10.1145/2600057.2602906.

[103] H. Zhu, P. Z. B, Z. Li, J. Xu, and L. Zhao, *Web and Big Data*, vol. 10367. Springer International Publishing, 2017.

[104] L. M. L. Pascoal, C. G. Camilo, E. Q. Da Silva, and T. C. Rosa, "A social-evolutionary approach to compose a similarity function used on event recommendation," *Proc. 2014 IEEE Congr. Evol. Comput. CEC 2014*, vol. i, pp. 1512–1519, 2014, doi: 10.1109/CEC.2014.6900495.

[105] A. D'Cunha and V. Patil, "Friend recommendation techniques in social network," *Proc. - 2015 Int. Conf. Commun. Inf. Comput. Technol. ICCICT 2015*, pp. 15–18, 2015, doi: 10.1109/ICCICT.2015.7045669.

[106] M. Jiang, P. Cui, F. Wang, W. Zhu, and S. Yang, "Scalable recommendation with social contextual information," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 11, pp. 2789–2802, 2014, doi: 10.1109/TKDE.2014.2300487.

[107] B. Yang, C. Wu, S. Sigg, and Y. Zhang, "CoCo (Context vs. content): Behavior-inspired social media recommendation for mobile apps," *2016 IEEE Glob. Commun. Conf. GLOBECOM 2016 - Proc.*, 2016, doi: 10.1109/GLOCOM.2016.7841666.

[108] C. Wu, Y. Zhang, J. Jia, and W. Zhu, "Mobile Contextual Recommender System for Online Social Media," *IEEE Transactions on Mobile Computing*, vol. 16, no. 12, pp. 3403–3416, 2017, doi: 10.1109/TMC.2017.2694830.

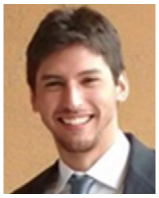
[109] F. Qiu, W. Ge, and X. Dai, "Code Recommendation with Natural Language Tags and Other Heterogeneous Data," *Proc. 2017 Int. Conf. Comput. Sci. Artif. Intell. - CSAI 2017*, pp. 137–142, 2017, doi: 10.1145/3168390.3168407.

[110] A. Khelloufi *et al.*, "A Social-Relationships-Based Service Recommendation System for SIoT Devices," *IEEE Internet Things Journal*, vol. 8, no. 3, pp. 1859–1870, 2021, doi: 10.1109/JIOT.2020.3016659.

[111] P. T. Crespo and C. Antunes, "Predicting teamwork results from social network analysis," *Expert Systems with Applications*, vol. 32, no. 2, pp. 312–325, 2015, doi: 10.1111/exsy.12038.

[112] A. Hernando, J. Bobadilla, and F. Ortega, "A non negative matrix factorization for collaborative filtering recommender systems based on a Bayesian probabilistic model," *Knowledge-Based Systems*, vol. 97, pp. 188–202, 2016, doi: 10.1016/j.knosys.2015.12.018.

[113] H. Bagci, "Appendix A8 – Unit conversion," *Handb. Gener. IV Nucl. React.*, pp. 871–878, 2016, doi: 10.1109/DSAA.2015.7344852.



Diego Alejandro Medel Canales

Diego received the B.Sc. degree from National University of San Juan (UNSJ), Argentina in 2014. Current is a student of a Ph.D. degree in Computer Science at UNSJ and a scholarship at the National Commission of Scientific and Technical Research (CONICET) in Argentina, for research of recommender systems. He also is a professor of Networks and Distributed and Parallel Systems at UNSJ.

Carina Soledad González González

Carina is a Full Professor of Computer Architecture Technology at the Department of Computer Engineering and Systems of the University of La Laguna (Spain). she is a Ph.D. in Computer Science (2001) by the University of La Laguna and a Ph.D. in Social Science and Education (2020) by the University of Huelva (Spain). Her research has focused on the field of Informatics applied to Education and Human-Computer Interaction, participating in different research projects and publishing widely on these topics.

Silvana V. Aciar

Silvana is a PhD in Information Technology by the University of Girona. Actually she works as a researcher at CONICET in the National University of San Juan (UNSJ), Argentina. Her research interests include artificial intelligence, machine learning, recommender systems, opinion mining, accessibility and ontologies. She has published articles in national and international magazines and congresses on these topics.

Neural Collaborative Filtering Classification Model to Obtain Prediction Reliabilities

Jesús Bobadilla*, Abraham Gutiérrez, Santiago Alonso, Ángel González-Prieto

ETSI Sistemas Informáticos, Universidad Politécnica de Madrid, Madrid (Spain)

Received 28 February 2021 | Accepted 3 June 2021 | Published 9 August 2021

unir
LA UNIVERSIDAD
EN INTERNET

Abstract

Neural collaborative filtering is the state of art field in the recommender systems area; it provides some models that obtain accurate predictions and recommendations. These models are regression-based, and they just return rating predictions. This paper proposes the use of a classification-based approach, returning both rating predictions and their reliabilities. The extra information (prediction reliabilities) can be used in a variety of relevant collaborative filtering areas such as detection of shilling attacks, recommendations explanation or navigational tools to show users and items dependences. Additionally, recommendation reliabilities can be gracefully provided to users: "*probably* you will like this film", "*almost certainly* you will like this song", etc. This paper provides the proposed neural architecture; it also tests that the quality of its recommendation results is as good as the state of art baselines. Remarkably, individual rating predictions are improved by using the proposed architecture compared to baselines. Experiments have been performed making use of four popular public datasets, showing generalizable quality results. Overall, the proposed architecture improves individual rating predictions quality, maintains recommendation results and opens the doors to a set of relevant collaborative filtering fields.

Keywords

Artificial Intelligence Systems, Neural Classification, Neural Collaborative Filtering, Recommender Systems.

DOI: 10.9781/ijimai.2021.08.010

I. Introduction

RECOMMENDER Systems (RS) [1]-[2] are Artificial Intelligence systems designed to reduce the Internet information overload problem. RS can recommend items to users, avoiding large manual search to select appropriated products or services. Amazon, TripAdvisor, Netflix and Spotify are remarkable commercial firms that use RS. The selected filtering approaches are the RS core. The most relevant filtering methods are content-based [3] (books abstracts, products descriptions), demographic-based [4] (gender, age, zip), context-aware [5] (gps location), social [6] (followers, followed, tags), Collaborative Filtering (CF) [7]-[8] and hybrid [9] ensembles that joins two or more different filters. From the existing filtering approaches, the CF is the more relevant because it provides improved accuracy. From a machine learning point of view, historically CF has been addressed by using k-nearest neighbours (KNN), then Matrix Factorization (MF) [10] and currently Neural Collaborative filtering (NCF) [11]-[12]. Both MF and NCF models create an internal dense representation of each sparse user and item vector. In the first case we call to the representations: hidden factors, whereas in NCF they are embedding values. MF makes use of the same vector space to code users and items factors; NCF can be designed to make use of the same or different vector spaces. Finally, MF combines factors in a linear mode, whereas NCF combines embedding values in a non-linear mode, making it possible to catch the existing complex non-linear relations between users and items.

* Corresponding author.

E-mail address: jesus.bobadilla@upm.es

NCF has emerged in the RS area providing even better accuracy [13] than the traditional MF approaches [12]. DeepMF [14] is a general framework that implements MF by means of a neural model; it is a regression-based model implemented making use of two Multilayer Perceptrons (MLP) and a 'Dot' output layer. On several benchmark datasets, this model outperforms state of art machine learning models. The NCF [11] model extends the DeepMF [14] approach, replacing the output 'Dot' layer with an MLP and catching the complex non-linear relations between items and users. The NCF model proposed in [11] also introduces input embedding layers to make the model more scalable than the DeepMF [14] approach. Both the DeepMF and the NCF are regression-based architectures. The NCF area has been expanded to several fields beyond the RS domain; as an example, in [15] authors propose a new computational method NCFM (Neural network-based Collaborative Filtering Method) to predict miRNA-disease associations based on deep neural network. An automated and unsupervised method for the mitral valve segmentation using neural network collaborative filtering is explained in [16]. A framework to attack the QoS prediction in the IoT environment [17] combines NCF and fuzzy clustering; the NCF model is designed to leverage local and global features. A two sequential stages model (MF and neural) to improve fairness in RS [18] obtains fair recommendations without losing a significant proportion of accuracy. Additionally, a current RS for researchers and students: Deep Edu [19] makes use of NCF and it outperforms existing Educational services recommendation methods. The NCF input embedding layers have been refined in [20] by means of a user-item interaction graph. Among the remarkable current NCF approaches, we have selected a Joint Neural Collaborative Filtering (J_NCF) [21] that couples deep feature learning and deep interaction modelling with a rating matrix; the Contextual-boosted Deep Neural Collaborative filtering (CDNC) model [22] which simultaneously exploits both item introductions (textual features) and user ratings (collaborative features), making and ensembling of collaborative and content-based filtering; Knowledge graphs have been used to enhance NCF to alleviate the sparsity problem [23]; in [24] authors effectively combines user–item interaction information and auxiliary knowledge information for recommendation task; a NCF is proposed for user generated list recommendation, combining both item-level information and list-level information to improve performance. Finally, a neural embedding collaborative filtering (NECF) [25] is designed by using unsupervised auto-encoders, generating the embedding vectors from the user-item data, followed by a regression stage; as it can be seen, this is a DeepMF version where embeddings are replaced by auto-encoders.

An emerging beyond accuracy RS area is focused on the obtention of reliability values associated to the prediction ones; in this way, each prediction and recommendation will be represented by the pair <prediction, reliability>. The extra information (reliability) can be used to modulate recommendations: "you *probably* will like Avengers: Infinity War", "We *really* recommend you Avatar". Despite the reliabilities usefulness it has not been a main goal in the RS area: traditionally, CF has been excessively focused on accuracy. As a result, we use the number of votes as reliability measure: usually users take note of the number of people that has voted an item, and we prefer some four stars gadget voted by 1500 clients than other similar item voted five stars by 12 clients. Accurate CF methods to obtain prediction reliabilities can lead to nicer recommendation indications, such as the above examples. Additionally, the reliability information can be used for remarkable emerging areas such as detecting shilling attacks [26]: an unsupervised approach for detecting shilling attacks based on user rating behaviours [27] uses Dirichlet allocation model to extract latent topics of user preferences from user rating item sequences, whereas an MF approach to detect shilling attacks [28] tests the unusual reliability variations in the item predictions. Reliability values have also been used to make dynamic browsing of related users or items [29], to explain recommendations [30] and to filter to the most reliable recommendations [31]. A variety of methods and models have been proposed to get prediction and recommendation reliabilities; in a first stage trust-based [32] and similarity measures-based [33] methods were developed. Two remarkable machine learning approaches make use of MF ensembles, the first one [31] designs two sequential MF where the first one obtains prediction errors from known ratings, whereas the second MF make predictions from the previous errors, just getting the expected reliability values. The second MF approach [34] is the Bernoulli Matrix Factorization (BeMF), which is a matrix factorization model based on the Bernoulli distribution to exploit the binary nature of the designed classification model. Basically, BeMF runs a MF for each possible vote in the RS (e.g.: 1 to 5 stars), returning the probability (reliability) of each rating. The RS reliability field is growing fast due to the emerging reliability quality measures: a reliability quality prediction measure (RPI) and a reliability quality recommendation measure (RRI) are proposed in [35]. Both quality measures are based on the hypothesis that the more suitable a reliability measure is, the better accuracy results it will provide when applied. Current NCF is based on regression models [11, 12, 14, 19, 20 , 21, 22], whereas our proposed model is a NCF architecture based on classification. This is an innovative approach whose RS accuracy must be compared to the conventional regression models. We have chosen the classification model because it presents a potential advantage: machine learning classification results provide probability distributions that could be used as reliability values in the CF context. Classification approaches naturally provide reliability values. A classification neural model for RS is proposed in [36], where the item relations patterns are learned in the network. It is designed to include an output layer containing as many neurons as items in the dataset, what usually is an affordable approach, although it could be considered as not scalable in specific scenarios; nevertheless, it provides prediction reliabilities. In [37] two sequential models have been designed to exploit the potentiality of the reliability information: the first model uses MF to obtain reliabilities, whereas the second one is a neural model that improves recommendation accuracy by incorporating in its input layer the previous MF reliability values.

The proposed classification architecture in this paper borrows the NCF classification concept from [36] and the NCF design from [11], trying to catch the individual strengths of both approaches. It also incorporates two innovative contributions to improve scalability through the use of input embeddings and to improve accuracy by replacing the usual regression 'Dot' layer with a 'Concatenate' layer that preserves the abstract information from hidden layers. By running the proposed model, we obtain pairs <prediction value, prediction reliability> and we make use of the reliability 'extra' information to get the most promising recommendations. Since the proposed classification architecture returns prediction reliabilities, it opens the door to remarkable state of art research fields such as mentioned above. The key question now is: will the classification approach return less accurate recommendations than the state of art regression approaches? If the answer is affirmative our classification architecture loses its purpose, whereas if it returns similar or higher accuracy than regression NCF it is valuable to use it and to take advantage of the reliability additional information. Thus, our hypothesis is that the proposed classification architecture provides similar or higher accuracy than the current NCF regression models, making it possible to take advantage of the extra reliability information it returns. The rest of the paper has been structured as follows: in Section II the proposed model is explained, and the experiments design is defined. Section III shows the experiments' results and their discussions. Finally, Section IV contains the main conclusions of the paper and the future works.

II. Materials and Methods

This section contains two different subsections: the first one defines and explains the design of the proposed architecture and its relationship with the state of art models. The second subsection focuses on the experiments design and their implementation, defining the selected datasets, the chosen baselines, the stablished parameter values and the tested quality measures.

A. The Proposed Classification Based Neuronal Architecture

The proposed classification architecture has been designed following the DeepMF and the NCF state of art evolution, taking those elements that are more advantageous to the classification-based approach and removing the ones that are not appropriate. From the DeepMF [14] we have borrowed the two designed MLP branches to separately process item and user's information. Fig. 1 shows the DeepMF architecture with its two separate *N* layers MLP neural networks. However, this DeepMF design has a remarkable drawback: it has not an appropriate scalability.

As it can be seen in Fig. 1, both MLP branches are fed by using very large vectors as input: each user vector of ratings and each item vector of ratings. Please note that each item vector can contain millions of ratings, since some RSs contain explicit or implicit ratings coming from millions of users. Our classification-based solution makes use of an optimized approach to feed the neural network: the use of embedding layers. This architectural solution has been used in the NCF [11] design as shown in Fig. 2: now the input layer is coded by using vectors of bits that are processed in the embedding layers, providing both the user and the item latent vectors. Additionally, and MLP neural network is used to process the latent vectors instead of the "Dot" layer used in the DeepMF approach; this MLP makes possible to find the complex non-linear dependences existing among the embedding values.

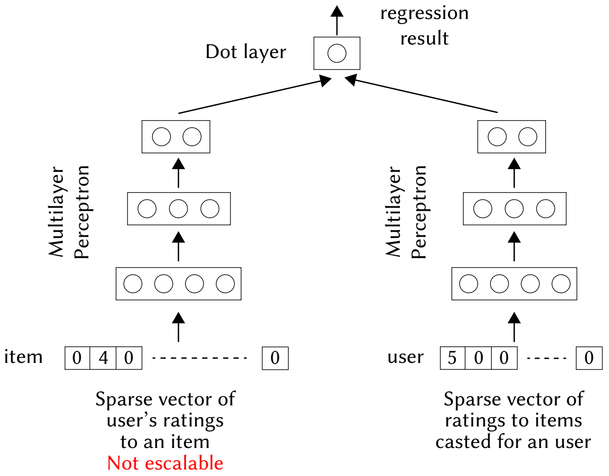


Fig. 1. DeepMF architecture [14].

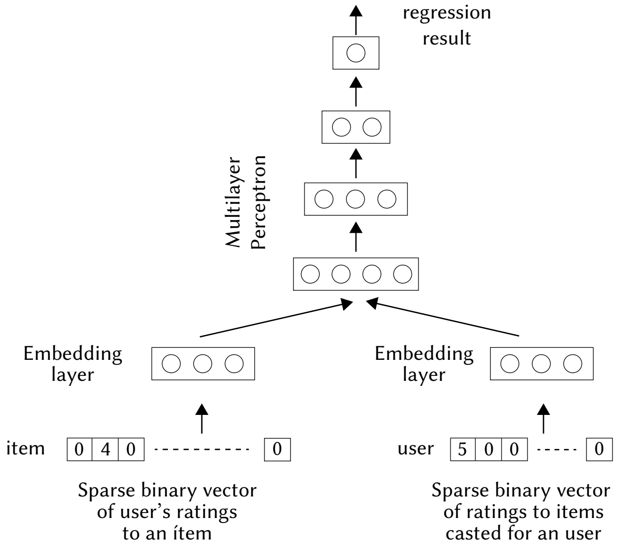


Fig. 2. NCF architecture [11].

Our proposed architecture is shown in Fig. 3; its inputs are just two numbers (for each existing rating): the item code and the user code. These two numbers feed two separate embedding layers that code each number as a vector of *F* values. The embedding process assigns similar embedding layer vector values to similar users (or similar items), making much easier the subsequent MLP tasks. The embedding layer operations make use of a lookup table where the keys are the users (or items) numbers, and the values are the dense vectors of size *F*. In this way we do not longer need to maintain explicit one-hot encoder sparse vectors as the ones shown in the NCF architecture (Fig. 2). Embedding layers are mainly used in natural language processing scenarios due to the huge sparsity of the words in sentence representations. RS datasets also present high sparsity levels, since users only cast ratings from a very reduced proportion of the available items. The embedding layer can be instanced by providing the maximum number of existing elements (users or items in the CF context) and the size of the compressed information (usually from 5 to 20 values in the CF context). The proposed architecture embedding layer for items receives as input each item number from each existing rating. Likewise, the user's embedding layer receives as input each user number from each existing rating. Each input number of the sequence is used as index to access a lookup table (embedding weight matrix) containing vectors for each user (or item). The lookup tables efficiently implement the embedding layers. Both the user and the item embedding layers compress and code each item and user ID (number). Once the embedding training process is finished, in the same way in which NLP related words have close embedding representations, CF related users and items have close embedding representations. Since our proposed architecture is classification based, the output layer has *V* neurons (Fig. 3), in contrast to the regression NCF version (Fig. 2) where one single neuron makes the regression. The *V* value depends on the number of different available implicit ratings or explicit votes in the RS (e.g.: 1, 2, 3, 4 or 5 stars). The output of the multilayer perceptron is positional (one neuron for each rating value, in this case 5-stars). Each output neuron provides a reliability measure. Whereas the NCF classification loss function is binary, in our classification architecture we use a categorical cross entropy loss; thus, our classification output layer returns *V* probabilities: $v_i \in \mathcal{R}$, $i \in \{1, 2, ..., V\}$, $\sum_i v_i = 1$. Please note that the v_i value's *argmax* function provides us with the discrete prediction, whereas the corresponding v_i value can be seen as the prediction reliability. This is a very different scenario to the regression based NCF, where prediction results are continuous prediction values. The classification NCF results are richer than the regression ones and it opens the door to use this additional information to deal with diverse RS goals such as providing prediction reliabilities, improving recommendation quality, making items or users relations graphs, or explaining recommendations.

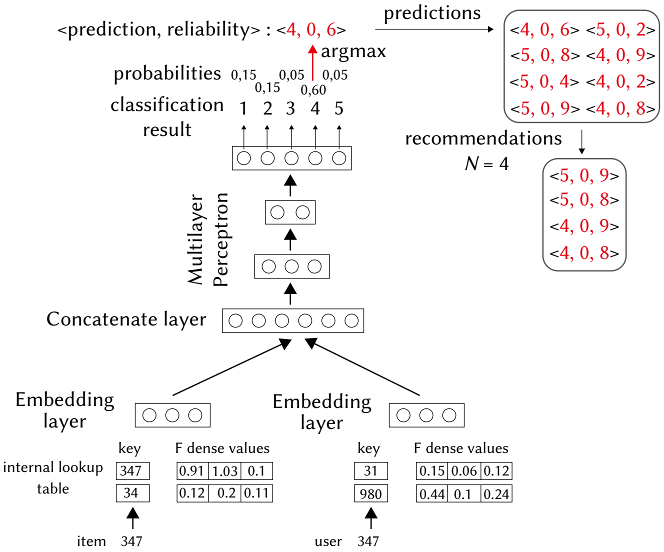


Fig. 3. Proposed architecture and recommendation method.

In short, we encourage the use of an NCF classification-based architecture whose samples are simple item and user numbers, and whose output labels are ratings. This design avoids large input vectors, and it returns <*rating*, *reliability*> prediction pairs. Hereafter we provide an example to show the flexibility of the classification based NCF: we will choose the most promising recommendations from their rich results, compared to the restricted operation evolved in the regression approach. From the following set of pairs <*prediction*, *reliability*>: <5, 0.3>, <5, 0.2>, <5, 1>, <5, 0.9>, <4, 0.8>, <4, 0.4>, <4, 0.7>, <3, 0.7>, the following ordered list of recommendations could be obtained: <5, 1>, <5, 0.9>, <4, 0.8>, <4, 0.7>. We have just filtered to the highest predictions (4 & 5) and the highest reliabilities (reliability >= 0.5), and then we have ordered the resulting pairs attending to their reliability. As it can be seen, risky recommendations have been avoided (<5, 0.2>, <5, 0.3>, <4, 0.4>) in a process that is not available using the NCF regression results.

The Algorithm 1 Keras code section shows a specific implementation of the proposed NCF classification architecture. In this case, the dataset contains 5 categories (1 to 5 stars) and the chosen embedding size is 10 (line 1). Lines 2 to 4 input each movie number code, embedding them and prepare their flatten representation; lines 5 to 7 make the same process for each user number code. Both the movie and user flatten representations are concatenated in the 'Concatenate' layer in line 8. The existing complex non-linear relations between users and items are learnt in the hidden layers coded from line 9 to line 12. Line 13 creates the five categories classification output layer, and it assigns the softmax activation function to the neurons. The whole model is created in line 14, defining the users/items input tensor and the output layer. Finally, the compile and fit methods are set in lines 15 and 16.

Algorithm 1. Classification NCF design

```
1. embed_size = 10; num_categ = 5

2. movie_input = Input(shape=[1])
3. movie_embedding =
    Embedding(num_movies + 1, embed_size)(movie_input)
4. movie_flatten = Flatten()(movie_embedding)

5. user_input = Input(shape=[1])
6. user_embedding =
    Embedding(num_users + 1, embed_size)(user_input)
7. user_flatten = Flatten()(user_embedding)

8. concat =
    Concatenate(axis=1)([movie_flatten, user_flatten])
9. mlp_1 = Dense(80, activation=’relu’)(concat)
10. mlp_2 = Dropout(0.4)(mlp_1)
11. mlp_3 = Dense(25, activation=’relu’)(mlp_2)
12. mlp_4 = Dropout(0.4)(mlp_3)
13. output =
    Dense(num_categ, activation=’softmax’)(mlp_4)

14. model_classification =
     Model([user_input, movie_input], output)
15. model_classification.compile(
       optimizer=’adam’, metrics=[‘mae’],
       loss=’categorical_crossentropy’)
16. history = model_classification.fit(
    [train[:,USER], train[:,ITEM]],
    to_categorical(train[:,RATING]),
    validation_data=([test[:,USER],test[:,ITEM]],
    to_categorical(test[:,RATING])), epochs=EPOCHS,
    verbose=1)
```

B. Experiments Design

This paper's hypothesis claims that classification based NCF provides similar or better accuracy than the regression based NCF model. Additionally, classification based NCF allows to tackle a variety of RS goals in a simpler way than the regression approach. To test the comparative accuracy of both NCF approaches we have designed a set of experiments involving prediction and recommendation results, processing a variety of baselines and using several public CF datasets. The selected RS datasets are: Movielens 100K [38], Movielens 1M [38], MyAnimeList* (a subset of the original dataset) [39] and Netflix* [40] (a subset of the original dataset). Table I shows these datasets main parameter values.

TABLE I. Main Parameters of the Datasets Used in the Experiments

Dataset	#users	#items	#ratings	scores	sparsity
Movielens 100K	943	1682	99,831	1 to 5	93,71
MovieLens 1M	6,040	3,706	911,031	1 to 5	95,94
MyAnimeList*	19,179	2,692	548.967	1 to 10	98,94
Netflix*	23,012	1,750	535,421	1 to 5	98,68

The paper's baselines are: DeepMF [14], NCF regression [11], NCF classification, and binary NCF classification. NCF classification corresponds to the architecture in Fig. 3, whereas binary NCF classification is the NCF classification version where ratings have been converted to "relevant" or "not relevant" (e.g.: relevant ⟺ rating >= 4, not relevant ó rating < 4), and labels have also been converted to the exposed 'relevant' and 'not relevant' discrete classification. The difference between NCF classification and the proposed method is that reliability results are not used in the baseline. Table II abstracts both the proposed and the baselines architectures. Both the DeepMF and the regression architectures make use of a 'Dot' layer to join their embeddings, whereas a 'Concatenate' layer has been used in both the proposed and the baseline classification architectures.

TABLE II. Proposed and Baseline Architectures

Architecture	type	merge layer	Reliability inf.
Proposed	classification	concatenate	yes
DeepMF	regression	dot	no
Regression	regression	dot	no
Classification	classification	concatenate	no
Binary classification	classification	concatenate	no

TABLE III. Experiments and Their Main Parameter Values

Experiment	# recomm. (N)	Relevancy threshold	Prediction value threshold
Precision / recall	{2,4,6, ..., 10}	{3, 4, 5} Movielens and Netflix {7, 8, 9} MyAnimeList	{0}
Quality predicting each rating {1,2,...,5}	{2, 6, 10} Movielens and Netflix {7, 8, 9} MyAnimeList	{4}	{0}
Precision vs. coverage	{10}	{3, 4, 5}	{4, 4.2, 4.4, 4.6, 4.8}

The main experiments make use of the precision and recall recommendation measures, tested by using a range from 2 to 10 number of recommendations (*N*). A secondary set of experiments tests the prediction quality on each of the dataset ratings (1 to 5 stars), using 2, 6 or 10 number of recommendations (7, 8 and 9 for the MyAnimeList dataset). Finally, an experiment tests the precision versus coverage obtained by filtering to predictions higher than a *beta* threshold. In the CF context, the relevancy threshold parameter is used to classify a rating or a prediction in the categorical set {relevant, not_relevant}. Experiments on datasets where votes range from 1 to 5 (Movielens, Netflix, etc.) usually set the relevancy threshold in the value 4; that is: ratings 4 and 5 are considered relevant, and predictions greater than or equal to 4 are considered as candidates to be recommended. The relevancy threshold is also important in the cross-validation testing process, since recommendations made to items voted under 4 are considered as errors, whereas recommendations made to items

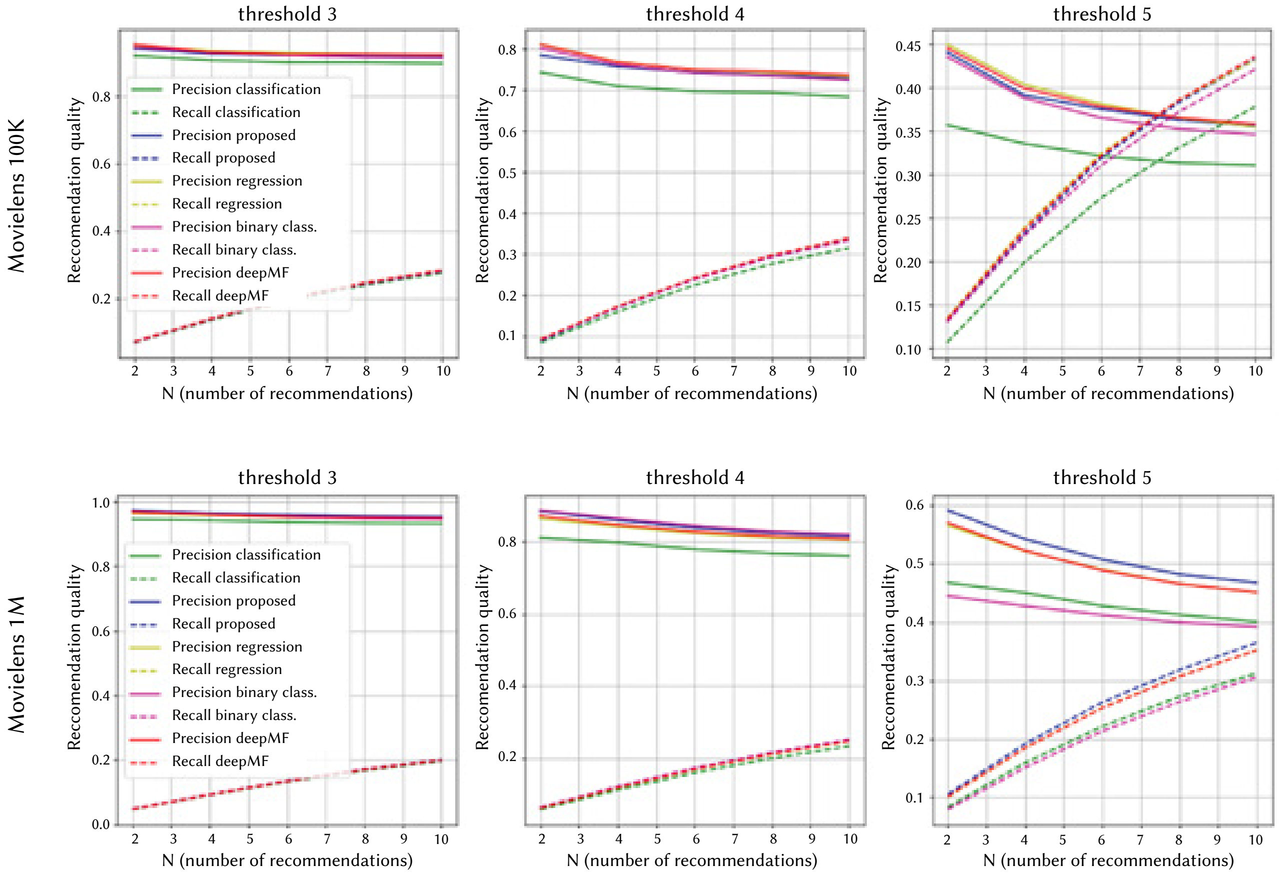


Fig. 4. Precision and recall quality results obtained by using the Movielens 100K (top graphs) and the Movielens 1M (bottom graphs). Three relevancy thresholds have been tested: 3 (left graphs), 4 (center graphs), 5 (right graphs). The proposed architecture has been compared to the baselines: deepMF [14], NCF regression [11], regular NCF classification and binary NCF classification. Precision and recall "proposed" values are the results of the proposed method in the paper (Table II).

voted 4 or above are considered as successes. In this example, setting the relevancy threshold to 5 makes it more difficult to get high success (precision, recall, etc.) levels. Table III shows the experiments and their main parameter values.

III. Results

Our first experiment compares the quality of recommendations obtained by using the proposed method and the baselines. The chosen CF quality measures have been the precision and recall ones. The main parameters of this experiment are abstracted in the Table III first row. Fig. 4 shows the results obtained on both the Movielens 100K (top graphs) and Movielens 1M (bottom graphs). The main conclusion is that, as expected, the proposed classification based NCF method reaches adequate recommendation quality results, compared to state of art NCF baselines. Additionally, we can see that both the recall and the precision evolutions are the expected ones: precision decreases as the number of recommendations (*N*) increases, whereas recall increases as the *N* value increases. The y axis scale shows better results in the graphs corresponding to the 1M Movielens version compared to the 100K one. Fig. 4 results are particularly interesting when the recommendations difficulty is increased: when the relevancy threshold is set to its limit (value 5); in this scenario we find a Movielens 100K optimum number of recommendations: 7 (graph on the top-right of Fig. 4), an also a remarkable behaviour of the proposed method when applied to Movielens 1M (graph on the bottom-right of Fig. 4).

The same kind of experiments have been performed by using the MyAnimeList* and Netflix* datasets (Table I). Results confirm the conclusions obtained in both Movielens datasets, particularly it is confirmed that the proposed NCF classification approach reaches adequate recommendation results. The top graphs of Fig. 5 show the MyAnimeList* results; this dataset has a range of votes from 1 to 10, instead the usual 1 to 5. This circumstance reduces the quality of the binary classification baseline, since the binary differentiation between relevant and not relevant ratings becomes less precise.

Fig. 5 top-center and top-right graphs shows the mentioned quality drop. It is also remarkable that in both Fig. 4 and Fig. 5 we can observe a better behaviour of the NCF classification proposed approach than the NCF classification regular method; this means that an adequate use of the reliability information leads to the expected quality improvements.

The explained experiments are based on the usual approach to test recommendation quality: proportion of the relevant recommended items versus the total number of recommendations (precision) or versus the total number of relevant items (recall). In both cases, we are considering a hit if the recommended item exceeds a relevancy threshold. In some RS scenarios we need a more fine-grained predictions and recommendations. This is the case when we want to be sure we will not like an item, or we will only like it if it has been recommended with a particular value; e.g.: I am interested in a set of songs, but I want to discard everyone who has obtained 1 or 2 stars. I want to watch some films unknown to me, then I will surf by those the RS recommendation is three stars. To test the proposed and the baselines NCF approaches in this scenario, the experiments shown in the second row of Table III have been run. Basically, we obtain the quality of each architecture to appropriately recommend each existing

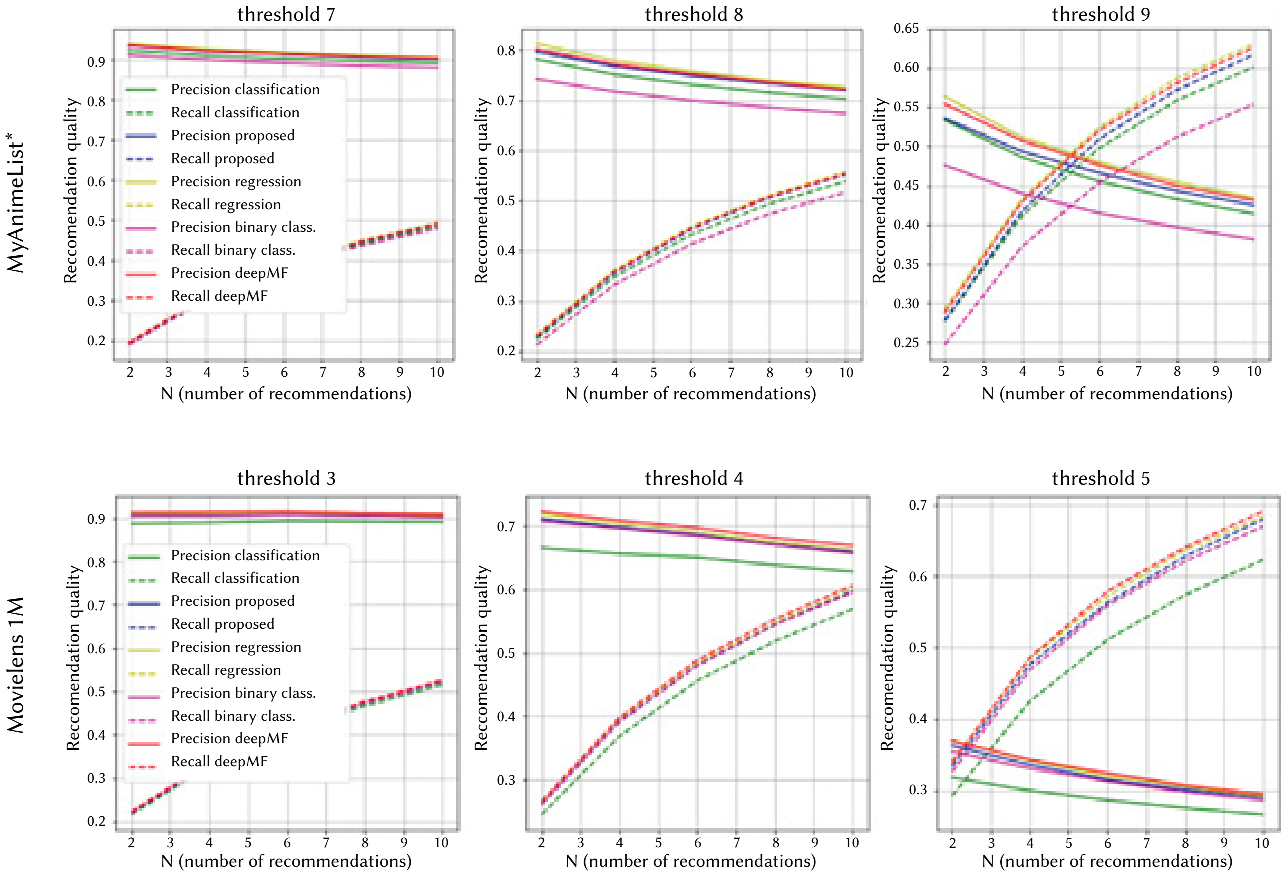


Fig. 5. Precision and recall quality results obtained by using the MyAnimeList* (top graphs) and the Netflix* (bottom graphs). Three relevancy thresholds have been tested: 7 or 3 (left graphs), 8 or 4 (center graphs), 9 or 5 (right graphs). The proposed architecture has been compared to the baselines: DeepMF [14], NCF regression [11], regular NCF classification and binary NCF classification. Precision and recall "proposed" values are the results of the proposed method in the paper (Table II).

rating in the CF dataset. The binary NCF classification has not been included since it is not designed to predict individual ratings.

Fig. 6 shows the obtained results on the covered datasets. Beyond details, it can be seen the superiority of the proposed NCF method in all the contemplated scenarios. This is the expected result, since DeepMF and NCF regression just provide prediction values, whereas the proposed NCF classification architecture returns specific and complete classification information shaped like <*rating*, *reliability*> pairs. The remarkable differences in the quality results of the different ratings (*x*-axis) when predicted by the NCF classification or regression methods is due to the CF datasets are usually biased to high rating values. As an example: both Movielens datasets show a much bigger difference between NCF classification and regression methods for the rating 1 and the rating 2 predictions; users tend to cast high votes (4 and 5), and then the number of low votes (1 and 2) are usually scarce in CF datasets. In this scenario, the NCF classification method particularly shows its superiority over the NCF regression one.

Finally, a set of experiments have been conducted (third row in Table III) to compare precision and coverage in the NCF scenario. It is important to realize that several parameters determine the recommendation coverage in a cross-validation scenario: a) the dataset distribution of ratings, particularly the rating matrix sparsity, b) the requested number of recommendations (*N*), c) the required threshold (4 stars, 5 stars); e.g.: it can be difficult to find users to recommend *N*=10 items with a 5 threshold in a cross-validation testing scenario. Usually, recommendation quality and coverage are inversely related. To conduct the experiments, we introduce a *beta* threshold used to select predictions that are greater than *beta*; e.g.: centered graph in Fig. 7 shows recommendation results where the relevancy threshold is 4 stars (top label); looking at the *x*-axis in its 4.4 *beta* threshold we can know the obtained coverage and recommendation precision selecting predictions equal to or higher than 4.4. As it can be seen, by increasing the *beta* value we also increase precision, but at the cost of a sharp decrease of the coverage. In the current experiment, the key question is to compare the precision versus coverage equilibrium in both the proposed method and the baselines. We can observe that the proposed method gets and intermediate position between DeepMF and NCF regression, and we can conclude that the baselines and the proposed method similarly performs in this particular issue: thus, the proposed classification architecture use does not worsen the RS coverage.

IV. Conclusion

Current collaborative filtering deep learning architectures are focused on the regression approach; they provide accurate predictions and recommendations compared to the state of art, but they do not return any reliability value of such predictions and recommendations. Conversely, classification based deep learning architectures are able to provide both the value and the reliability of each prediction or recommendation. By combining the prediction value and the reliability information it is possible to afford several remarkable tasks, such as obtaining more reliable recommendations, making some reliability-based explanations of the recommendations to users, showing navigable trees that relate users or items, implementing methods to reduce the shilling attacks consequences, etc.

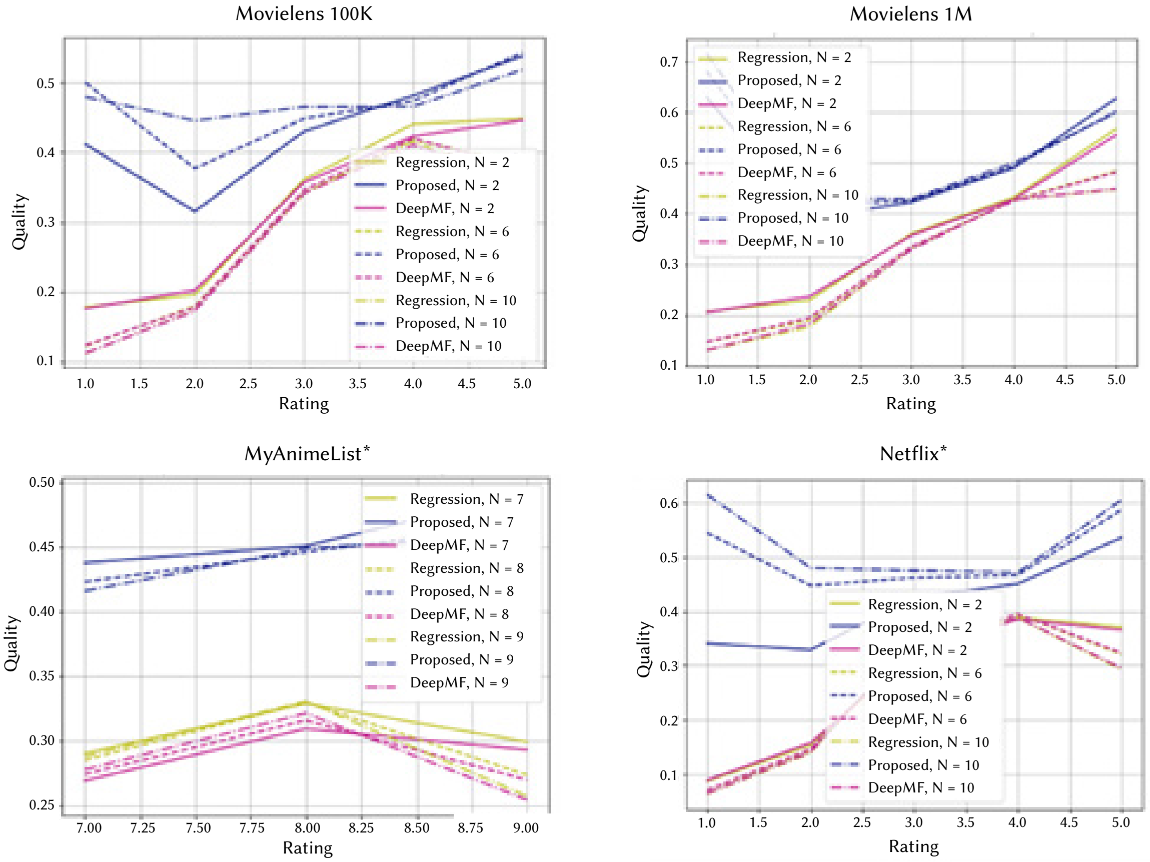


Fig. 6. Obtained precision quality recommending items for each considered rating; e.g.: proportion of hits recommending items that the user has voted with 2 stars. Top-left graph: Movielens 100K, top-right graph: Movielens 1M, bottom-left graph: MyAnimeList*, bottom-right graph: Netflix*. x-axis: ratings (7, 8 and 9 for MyAnimeList*; 1, 2, 3, 4 and 5 in Netflix* and Movielens). The proposed architecture has been compared to the baselines: DeepMF [14], NCF regression [11]. "Proposed" values are the results of the proposed method in the paper (Table II).

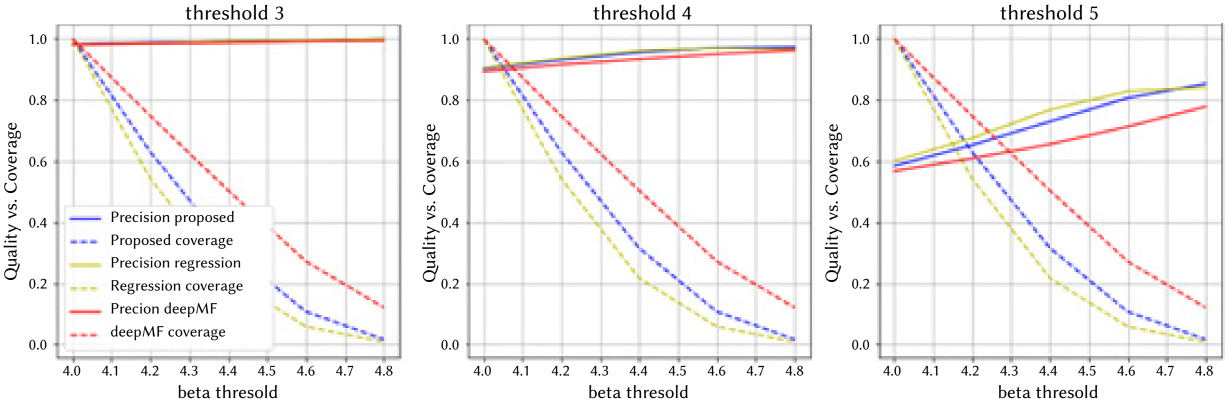


Fig. 7. Precision versus coverage results. Dataset: Movielens 1M; number of recommendations (*N*): 10; relevancy threshold: 3 stars (left graph), 4 stars (center), 5 stars (right graph); *x*-axis: *beta* threshold (selects predictions equal or higher than *beta*). Precision and coverage "proposed" values are the results of the proposed method in the paper (Table II).

Despite the mentioned advantages of the classification-based neural collaborative filtering approach, the proposed approach cannot be embraced without testing its quality performance: similar or better results must be obtained by applying the proposed architecture compared to the state of art regression baselines. Experiments in this paper show that the proposed classification architecture obtains similar recommendation accuracy results than the regression architectures do; precision and recall measures provide comparable quality results in a complete set of experiments where different thresholds and diverse number of recommendations are chosen. Results show a consistent pattern when experiments have been run on four public datasets. On the other hand, the proposed architecture shows improved prediction results: it is able to accurately predict individual ratings, outperforming prediction quality compared with the state of art regression approaches. Finally, the precision versus coverage balance stays similar in both the proposed and the baselines neural architectures.

In short, the proposed classification-based architecture can replace the state of art neural collaborative filtering approaches: its use does not worsen the recommendation quality, it improves the prediction of individual ratings, and it opens the door to a set of relevant collaborative filtering areas. Remarkable future works from this paper are: to make use of reliabilities to detect shilling attacks, to provide reliability values in the users' recommendations, and to filter non reliable recommendations.

Acknowledgment

This research was supported by the *Agencia Estatal de Investigación of Spain* (PID2019-106493RB-I00/AEI/10.13039/ 501100011033) and by the Comunidad de Madrid under Convenio Plurianual with the Universidad Politécnica de Madrid in the actuation line of *Programa de Excelencia para el Profesorado Universitario*.

References

[1] K. Madadipouya, S. Chelliah, "A Literature Review on Recommender Systems Algorithms, Techniques and Evaluations", *Brain: Broad Research in Artificial Intelligence and Neuroscience*, vol. 8, no. 2, 2017, pp. 109-124.

[2] S.S. Sohail, J. Siddiqui, R. Ali, "Classifications of Recommender Systems: A review", *Journal of Engineering Science and Technology Review*, vol. 10, no. 4, 2017, pp. 132-153.

[3] H. Zamani, A. Shakery, "A language model-based framework for multi-publisher content-based recommender systems", *Information Retrieval Journal*, vol. 21, no. 5, 2018, pp. 369-409.

[4] M.Y.H. Al-Shamri, "User profiling approaches for demographic recommender systems", *Knowledge-Based Systems*, vol. 100, 2016, pp. 175-187.

[5] N.M. Villegas, C. Sánchez, J. Díaz-Cely, G. Tamura, "Characterizing context-aware recommender systems: A systematic literature review", *Knowledge-Based Systems*, vol. 140, 2018, pp. 173-200.

[6] A. Rezvanian, B. Moradabadi, M. Ghavipour, M.M. Daliri Khomami, M.R. Meybodi, "Social recommender systems", *Studies in Computational Intelligence*, vol. 820, 2019, pp. 281-313.

[7] A. Hernando, J. Bobadilla, F. Ortega, A. Gutiérrez, "A probabilistic model for recommending to new cold-start non-registered users", *Information Sciences*, vol. 376, 2017, pp. 216-232.

[8] J. Bobadilla, A. Gutiérrez, S. Alonso, R. Hurtado, "A Collaborative Filtering Probabilistic Approach for Recommendation to Large Homogeneous and Automatically Detected Groups", *International Journal of Interactive Multimedia and Artificial Intelligence*, 2020, doi: 10.9781/ijimai.2020.03.002.

[9] V. Yu. Ignat'ev, D. V. Lemtyuzhnikova, D. I. Rul', I. L. Ryabov, "Constructing a Hybrid Recommender System", *Journal of Computer and Systems Sciences International*, vol. 57, no. 6, 2018, pp. 921-926.

[10] H. Li, Y. Liu, Y. Qian, N. Mamoulis, W. Tu, Wenting ; D. Cheung, "HHMF: hidden hierarchical matrix factorization for recommender systems", *Data Mining and Knowledge Discovery*, vol. 33, no. 6, 2019, pp. 1548-1582.

[11] H. Xiangnan, L. Lizi, Z. Hanwang, "Neural Collaborative Filtering", in *International World Wide Web Conference Committee (IW3C2)*, Perth, Australia, 2017, doi: 10.1145/3038912.3052569

[12] D. Bokde, S. Girase, D. Mukhopadhyay, "Matrix Factorization Model in Collaborative Filtering Algorithms: A Survey", *Procedia Computer Science*, vol. 49, 2015, pp. 136-146, doi: 10.1016/j.procs.2015.04.237.

[13] S. Rendle, W. Krichene, L. Zhang, J.R. Anderson, "Neural Collaborative Filtering vs. Matrix Factorization", in *RecSys '20: Fourteenth ACM Conference on Recommender Systems*, Brasil, 2020, pp. 240–248, doi: 10.1145/3383313.3412488.

[14] H.J. Xue, Xi. Dai, J. Zhang, S. Huang, J. Chen, "Deep Matrix Factorization Models for Recommender Systems", in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, Melbourne, Australia, 2017, pp. 3203-3209, doi: 10.24963/ijcai.2017/447

[15] Y. Liu, S.L. Wang, J.F. Zhang, W. Zhang, W. Li, "A neural collaborative filtering method for identifying miRNA-disease associations", *Neurocomputing*, vol. 422, 2021, pp. 176-185.

[16] L. Corinzia, F. Laumer, A. Candreva, M. Taramasso, F. Maisano, J.M. Buhmann, "Neural collaborative filtering for unsupervised mitral valve segmentation in echocardiography", *Artificial intelligence in medicine*, vol. 110, 2020, pp. 101975-101975.

[17] H. Gao, Y. Xu, Y. Yin, W. Zhang, R. Li, X. Wang, "Context-Aware QoS Prediction with Neural Collaborative Filtering for Internet-of-Things Services", *IEEE internet of things journal*, vol. 7, no. 5, 2020, pp. 4532-4542, doi: 10.110'9/JIOT.2019.2956827.

[18] J. Bobadilla, R. Lara-Cabrera, A. González-Prieto, F. Ortega, "DeepFair: Deep Learning for Improving Fairness in Recommender Systems", *International Journal Of Interactive Multimedia And Artificial Intelligence*, 2020, doi: 10.9781/ijimai.2020.11.001.

[19] F. Ullah, B. Zhang, R.U. Khan, T.S. Chung, M. Attique, K. Khan, S. Khediri, S. Jan, "Deep Edu: A Deep Neural Collaborative Filtering for Educational Services Recommendation", *IEEE access*, vol. 8, 2020, pp. 110915-110928.

[20] Y. Guo, Z. Yan, "Recommended System: Attentive Neural Collaborative Filtering", *IEEE access*, vol. 8, 2020, pp. 125953-125960.

[21] W. Chen, F. Cai, H. Chen, M. Rijke, "Joint Neural Collaborative Filtering for Recommender Systems", *ACM transactions on information systems*, vol. 37, no. 4, 2019, pp. 1-30.

[22] S. Yu, M. Yang, Min, Q. Qu, Y. Shen, "Contextual-boosted deep neural collaborative filtering model for interpretable recommendation", *Expert systems with applications*, vol. 136, 2019, pp. 365-375.

[23] L. Sang, M. Xu, S. Qian, X. Wu, "Knowledge graph enhanced neural collaborative recommendation", *Expert systems with applications*, vol. 164, 2021, pp. 113992, doi: 10.1016/j.eswa.2020.113992.

[24] C. Yang, L. Miao, B. Jiang, D. Li, D. Cao, "Gated and attentive neural collaborative filtering for user generated list recommendation", *Knowledge-based systems*, vol. 187, 2020, pp. 104839.

[25] T. Huang, D. Zhang, L. Bi, "Neural embedding collaborative filtering for recommender systems", *Neural computing & applications*, vol. 32, no. 22, 2020, pp. 17043-17057.

[26] M. Si, Q. Li, "Shilling attacks against collaborative recommender systems: a review", *The Artificial intelligence review*, vol. 53, no. 1, 2018, pp. 291-319.

[27] F. Zhang, Z. Ling, S. Wang, "Unsupervised approach for detecting shilling attacks in collaborative recommender systems based on user rating behaviours", *IET information security*, vol. 13, no. 3, 2019, pp. 174-187.

[28] S. Alonso, J. Bobadilla, F. Ortega, R. Moya, "Robust Model-Based Reliability Approach to Tackle Shilling Attacks in Collaborative Filtering Recommender Systems", *IEEE access*, vol. 7, 2019, pp. 41782-41798.

[29] A. Hernando, J. Bobadilla, F. Ortega, A. Gutiérrez, "Method to interactively visualize and navigate related information", *Expert Systems with Applications*, vol. 111, 2018, pp. 61-75.

[30] A. Hernando, R. Moya, F. Ortega, J. Bobadilla, "Hierarchical graph maps for visualization of collaborative recommender systems", *Journal of Information Science*, vol. 40, no. 1, 2014, pp. 97-106.

[31] B. Zhu, F. Ortega, J. Bobadilla, A. Gutiérrez, "Assigning reliability values to recommendations using matrix factorization", *Journal of computational science*, vol. 26, 2018, pp. 165-177.

[32] S. Ahmadian, P. Moradi, F. Akhlaghian, Fardin, "An improved model of trust-aware recommender systems using reliability measurements", in *6th Conference on Information and Knowledge Technology (IKT)*, Shahrud, Iran, 2014, pp. 98-103.

[33] A. Hernando, J. Bobadilla, F. Ortega, J. Tejedor, "Incorporating reliability measurements into the predictions of a recommender system", *Information Sciences*, vol. 218, 2013, pp. 1-16.

[34] F. Ortega, R. Lara-Cabrera, A. González-Prieto, J. Bobadilla, "Providing reliability in recommender systems through Bernoulli Matrix Factorization", *Information sciences*, vol. 553, 2021, pp. 110-128.

[35] J. Bobadilla, A. Gutiérrez, F. Ortega, B. Zhu, "Reliability quality measures for recommender systems", *Information Sciences*, Vol. 442-443, 2018, pp. 145-157.

[36] J. Bobadilla, F. Ortega, A. Gutierrez, S. Alonso, "Classification-based Deep Neural Network Architecture for Collaborative Filtering Recommender Systems", *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 6, no. 1, 2020, pp. 68-77.

[37] J. Bobadilla, S. Alonso, A. Hernando, "Deep learning architecture for collaborative filtering recommender systems", *Applied Sciences*, vol. 10, no. 7, 2020, pp. 2441.

[38] F.M. Harper, J.A. Konstan, "The movielens datasets: History and context",

ACM Transactions on Interactive Intelligent Systems, vol. 5, no. 4, 2015, pp. 1–19.

[39] https://www.kaggle.com/azathoth42/myanimelist

[40] F. Ortega, B. Zhu, J. Bobadilla, A. Hernando, "CF4J: Collaborative filtering for Java", *Knowledge-Based Systems*, vol. 152, 2018, pp. 94–99.

Jesús Bobadilla

Jesús Bobadilla received the B.S. and the Ph.D. degrees in computer science from the Universidad Politécnica de Madrid and the Universidad Carlos III. Currently, he is a full professor with the Department of Information Systems, Universidad Politécnica de Madrid. He is a habitual author of programming languages books working with McGraw-Hill, Ra-Ma and Alfa Omega publishers. His research interests include information retrieval, recommender systems and speech processing. He oversees the FilmAffinity.com research team working on the collaborative filtering kernel of the web site. He has been a researcher into the International Computer Science Institute at Berkeley University and into the Sheffield University. Head of the research group.

Abraham Gutiérrez

Abraham Gutiérrez received the B.S. and the Ph.D. degrees in computer science from the Universidad Politécnica de Madrid. Currently, he is currently an associate professor with the Department of Information Systems, Universidad Politécnica de Madrid. He is the author of search papers in most prestigious international journals. He is a habitual author of programming languages books working with McGraw-Hill, Ra-Ma and Alfa Omega publishers. His research interests include P-Systems, machine learning, data analysis and artificial intelligence. He is in charge of this group innovation issues, including the commercial projects.

Santiago Alonso

Santiago Alonso received his B.S. degree in software engineering from Universidad Autónoma de Madrid and his Ph.D. degree in computer science and artificial intelligence from Universidad Politécnica de Madrid, in 2015, where he is currently an associate professor, participating in master and degree subjects and doing work related with advanced databases. His main research interests include natural computing (P-Systems), and did some work on genetic algorithms. His current interests include machine learning, data analysis and artificial intelligence.

Ángel González-Prieto

Ángel González-Prieto received his Double B.S. in Computer Sciences and Mathematics from Universidad Autónoma de Madrid in 2014, his M.Sc. in Mathematics from the same university in 2015 and his Ph.D. in Mathematics from Universidad Complutense de Madrid in 2018. He has been postdoc at Instituto de Ciencias Matemáticas and, currently, he is teaching assistant at Universidad Politécnica de Madrid. His research interests include machine learning, deep learning, and algebraic geometry.

Automatic Finding Trapezoidal Membership Functions in Mining Fuzzy Association Rules Based on Learning Automata

Z. Anari[1], A. Hatamlou[2], B. Anari[3] *

[1] Department of Computer Engineering and Information Technology, Payam Noor University (PNU), Tehran (Iran)
[2] Department of Computer Engineering, Khoy Branch, Islamic Azad University, Khoy (Iran)
[3] Department of Computer Engineering, Shabestar Branch, Islamic Azad University, Shabestar (Iran)

Received 4 January 2020 | Accepted 9 June 2021 | Published 17 January 2022

unir
LA UNIVERSIDAD
EN INTERNET

Abstract

Association rule mining is an important data mining technique used for discovering relationships among all data items. Membership functions have a significant impact on the outcome of the mining association rules. An important challenge in fuzzy association rule mining is finding an appropriate membership functions, which is an optimization issue. In the most relevant studies of fuzzy association rule mining, only triangle membership functions are considered. This study, as the first attempt, used a team of continuous action-set learning automata (CALA) to find both the appropriate number and positions of trapezoidal membership functions (TMFs). The spreads and centers of the TMFs were taken into account as parameters for the research space and a new approach for the establishment of a CALA team to optimize these parameters was introduced. Additionally, to increase the convergence speed of the proposed approach and remove bad shapes of membership functions, a new heuristic approach has been proposed. Experiments on two real data sets showed that the proposed algorithm improves the efficiency of the extracted rules by finding optimized membership functions.

Keywords

Continuous Action-set Learning Automata (CALA), Data Mining, Fuzzy Association Rules, Learning Automata, Trapezoidal Membership Function.

DOI: 10.9781/ijimai.2022.01.001

I. Introduction

By increasing the volume of data in the databases, effective techniques are required to manage the data in these databases. Data mining is the process of exploring great amounts of data from transactional databases to obtain interesting information [1]-[2]. Some important data mining techniques include clustering [3], classification [4], prediction [5], text mining [6], and association rules [7-10]. Association rule mining is used to produce meaningful relationships among data elements within the transaction databases [11, 12]. One association rule is described as $X \rightarrow Y$, where X and Y belong to itemsets and $X \cap Y = \emptyset$. This means that, if the set of items belongs to X, it most likely also belongs to Y [12]. The Apriori algorithm is an efficient algorithm in data mining [13], and uses statistical techniques to find association rules.

Fuzzy theory used in many fields, such as engineering applications, optimization algorithms, data mining, and intelligent systems [14]-[26]. Some research papers have proposed algorithms for extracting fuzzy association rules [7], [27]-[35]. However, these methods only consider triangular membership functions that are not suitable for some applications. The methods using trapezoidal membership functions (TMFs) supposed that the shape of the trapezoidal membership functions for each linguistic term are well known. If so, some mined results may be inappropriate.

In this study, we addressed the above issue and proposed a novel algorithm that uses continuous action-set learning automata (CALA), named CALA-AFTM, to find position and the number of TMFs in fuzzy association rule mining at the same time. CALA is a mathematical approach that interacts with an environment and by using the environment response, generates a common reinforcement signal [36]. CALA has several advantages. Unlike other metaheuristic methods, to optimize a function, CALA only needs to build one team of LA which is equivalent to one chromosome in other evolutionary algorithms. Therefore, it requires fewer evaluation functions. It also needs fewer parameters to perform the optimization process, and can find the optimal local value. CALA applied in many applications and algorithms [37]-[42].

In this paper, finding positions and number of TMFs for each membership function has been regarded as parameters of the search space. To find these optimal parameters, a novel representation was suggested to build a CALA team. This team is divided into two parts. The learning automaton (LA) of the first part responsible for specifying the optimal number of membership functions, and the LA of the second part is used to optimize the positions of membership functions. Briefly, the main contributions of this research are as follows:

* Corresponding author.

E-mail addresses: zanari323@yahoo.com (Z.Anari), rezahatamloo@gmail.com (A.Hatamlou), anari@iaushab.ac.ir (B.Anari).

- The proposed approach dynamically determines the position and number of TMFs in mining fuzzy association rule.
- A new representation is developed to construct a CALA team.
- To reduce the domain of the search space and increase the speed of convergence, two constraints were proposed.
- A series of experiments on two real datasets conducted to represent the great effectiveness of the proposed approach.

To assess the results, the proposed CALA-AFTM algorithm compared with fuzzy web mining algorithm (FWMA) [31] and VSLA-AFTM algorithm. Two real datasets, CTI and NASA, were employed for the experiments. They were selected because they used the time that users spend on web pages, and since this parameter is a fuzzy variable, it can be considered as TMFs [43]-[47].

The rest of this document is organized as follows: In Section II, a review of related work is presented. Section III provides background information. Proposed CALA-AFTM algorithm is shown in Section IV. Section V provides the data set and experiment results. Finally, conclusion and future research is provided in Section VI.

II. Related Work

Most studies on data mining, used the predefined membership functions to derive fuzzy association rules. [22] described the definition and confidence factor of fuzzy association rule and proposed a fuzzy mining method to derive fuzzy association rules in databases. [20] introduced an algorithm named F-APACS for finding association rules in fuzzy data mining. They used fuzzy linguistic terms to manage quantitative values. This had two advantages: First, the user-supplied threshold did not have to be determined; second, it could find both positive and negative association rules. [48] presented a fuzzy multiple-level approach to search meaningful fuzzy association rules from quantitative values in transaction datasets. In their approach, first each quantitative value was converted to a linguistic variable. Then, for each fuzzy variable the scalar cardinality was computed. Finally, the mining process was conducted to obtain fuzzy association rules.

Several fuzzy data mining algorithms were also suggested to extract both the proper set of membership functions. These algorithms make it possible to automatically adjust membership functions using meta-heuristic approaches to extract the best fuzzy rules. [49] utilized the genetic algorithm and presented a mining approach to find the proper set of membership functions and fuzzy rules. In their approach, each membership function was converted into a fixed-length string. Then, appropriate strings were selected to build an appropriate membership function set.

Similarly, [50] enhanced their earlier method [49] and proposed a fuzzy mining approach using genetic algorithms and k-means clustering algorithm to find the best membership functions and fuzzy rules. In their method, one chromosome was assigned to each membership function. At first, each membership functions set was converted into a fixed-length string. Then, using the k-means clustering algorithm, each chromosome was assigned in one of the clusters. Finally, by assessing the fitness value, the appropriate membership functions were determined.

Using the genetic algorithm another mining approach proposed in [51] to obtain type-2 membership functions. There was a 2-tuple linguistic layout schema that encoded chromosome membership functions. In their algorithm, they used a parameter called the diversity factor to find suitable rules. In addition, [52] utilized the genetic algorithm to obtain the best membership functions and fuzzy rules. In their algorithm, a 2-tuple linguistic approach was proposed to reduce domain search space and remove improper membership functions.

Furthermore, [53] used the genetic algorithm and proposed a fuzzy mining approach to solve the problem of intrusion detection. [54] proposed GA-based approach to find concept-drift patterns and optimal membership functions. They employed a 2-tuple demonstration method to code the membership functions in chromosomes. [55] proposed a mining method and shown a new chromosome representation to identify suitable membership functions. In their approach, each chromosome represented membership functions sets and contained two parts. The first part was shown with binary strings and specified the activation of each membership function. The second part specified the parameters corresponding to the active membership functions. They obtained the optimal membership functions by checking whether the membership functions were active or not. [56] suggested a two-step method to dynamically identify fuzzy rules and optimal membership functions. In step one, they used a GA-based approach and 3-tuple scheme to obtain the optimal membership functions. In step two, they used pattern-growth method to derive fuzzy rules.

Also, [57] suggested a clustering method to automatically tune membership functions and extract weighted fuzzy rules. In their approach, each chromosome encoded by real number values and each population using the fitness function value was evaluated to find the suitable membership functions. using ant colony approach [58] proposed an approach to derive the optimal membership functions. In their method, each membership function encoded by binary strings. First an initial graph was created, then the ants moved in the graph and formed the final membership functions. [59] suggested a mining approach based on improved ant colony method to find appropriate membership functions. They developed a code representation and found the real global optimum solution in a continuous space by introducing certain operators.

[34] proposed a genetic-based approach using master-slave parallel processing technology to find the optimal membership functions and fuzzy rules. The main processor distributes fitness tasks among the slave processors. The fitness function was evaluated by each slave processor, and then the result of each slave processor sends to the main processor. Subsequently, the main processor used all the values of the fitness function to determine the best membership function. [60] presented a memetic based algorithm to extract optimal membership functions. Their algorithm for representing chromosomes the structure type of the membership function is considered. They proposed an approach to eliminate inappropriate membership functions using the nature of the structure in their method, thereby reducing the search space. In addition, their approach used structure types and extend the local search approach to reduce domain search space and find optimal membership functions.

[33] presented a GA-based approach to find appropriate membership functions. In their method, each membership function was demonstrated by three parameters. Each parameter corresponded to one of the vertices of the triangle, and these parameters encoded as chromosomes. Then, an appropriate membership functions sets was derived. Also, two heuristics proposed to remove improper membership functions and reduce domain search space. Using the Levenberg-Marquardt method and bacterial memetic algorithm [61] proposed a memetic based approach to find best fuzzy rules. In their approach, each bacterium represented the parameters of the fuzzy rule. [62] presented a mining approach using bat algorithm to find optimal membership functions dynamically. Their algorithm considered more factors in fitness function and by improving the local and global search extracted more precise rules.

[63] proposed a temporal mining approach, which combines the bee method and the fuzzy temporal mining method. Their algorithm obtained suitable membership functions and extracted fuzzy rules. [64] used a particle swarm optimization approach and proposed a

framework to extract optimal membership functions. In addition to single-objective optimization, some algorithms have also utilized multi-objective optimization to optimize the membership functions and extract association rules. [65] proposed a multi-objective optimization method for a classifier system using genetic algorithm. They used two criteria to assess precision and interpretability. Their method determined both the structure and membership functions on the basis of fuzzy rules. Using a multi-objective genetic algorithm [66] proposed a fuzzy mining approach to find fuzzy rules. They used three criteria as the goals of multi-objective optimization methods, namely, confidence, comprehensibility, and interestingness. [67] proposed a multi-objective-based approach using genetic algorithm for mining fuzzy association rules. In their approach, minimum support and minimum confidence parameters were determined automatically. Also, the fitness function specified the position of chromosomes and did not affect the genetic operator. Therefore, their method converged with the value of any fitness function.

To derive fuzzy web browsing patterns or association rules, many mining approaches have been reported. In [43] a fuzzy mining approach proposed to obtain sequential web patterns from the log records. In their algorithm, the importance of each web page was considered as a fuzzy variable and the importance of each web page transformed into fuzzy values. Using object-oriented concepts and fuzzy sets [29] proposed a web mining approach. In their method, each web page and the visited that web page were represented as a class and instance of that class, respectively. In their method, in the first step, for each web page association rules were determined and in the second step the relationships among all instances were determined. [27] proposed a GA-based approach to mine temporal association rules in web datasets. They showed that converting the data set to a graph extracted more reliable rules.

[68]-[69] introduced a generalized fuzzy web mining algorithm for extracting interesting association rules. Their approach used a fixed number of membership functions to convert quantitative web data to fuzzy values. [28] employed fuzzy concepts and proposed an algorithm to extract fuzzy rules from web usage data. In their method, the time duration of each web page visited by users were considered as a linguistic variable. [69] used a CALA algorithm and developed an algorithm to identify optimal membership functions which used in trust and distrust fuzzy recommender systems. Their method, only adjusted the center parameter of membership functions.

III. Preliminaries

This section describes the concepts associated with fuzzy association rules and CALA.

A. Fuzzy Association Rule

A fuzzy association rule is described as follows: Let $I = \{i_1, i_2, \ldots, i_m\}$ represent a set of items, and let $T = \{t_1, t_2, \ldots, t_n\}$ represent a set of transactions, and each transaction contains items from a set of items I. Thus, each transaction ti includes a subset of the items in I where $T \subseteq I$. An association rule is represented as $x \rightarrow y$, where $x \subseteq y$, $y \subseteq I$ and $x \cap y = \emptyset$, x and y are a set of items known as itemsets. Intuitively, the $x \rightarrow y$ rule means that transactions that contain x tend to contain y. For example, in market basket analysis, the association rule {bread, milk} → {butter} tells those customers have purchased bread and milk and are then likely to purchase butter. The goal association rule mining is to obtain rules in the transaction data set such that the support and confidence of each rule are greater than or equal to the minimum support and minimum confidence. The support value for the rule $x \rightarrow y$ is described as follows:

$$Support(x \rightarrow y) = \frac{|x \cup y|}{|T|} \quad (1)$$

The confidence value for the rule $x \rightarrow y$ is described as follows:

$$Confidence(x \rightarrow y) = \frac{|x \cup y|}{|x|} \quad (2)$$

The fuzzy support value for region $R_{j,k}$ is described as follows:

$$Fuzzy\ Support(R_{j,k}) = \frac{\sum_{i=1}^{n} f_{j,k}^{(i)}}{|T|} \quad (3)$$

where $R_{j,k}$ represents the fuzzy region for the k-th membership function of an item I_j, $f_{j,k}^{(i)}$ demonstrates the fuzzy membership function value of region $R_{j,k}$ in the i-th transaction.

B. Continuous Action-Set Learning-Automata (CALA)

LA is a mathematical tool for solving optimization problems. Each automaton selects an action from its set of actions and applies it to the environment. Then, LA uses the response received from the environment and decides whether the selected action is rewarded or penalized [38],[70]. LA has been used in many fields such as computer networks [71], image processing [42], speech analysis [72], signal processing [73], and clustering [74]. The interaction between the LA and the environment is shown in Fig. 1.

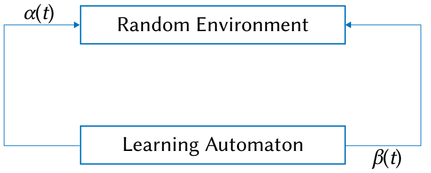


Fig. 1. The interaction between the LA and the environment.

The environment is determined by a triple <α, β, c>, where $\alpha = \{\alpha_1, \alpha_2, \ldots, \alpha_r\}$ shows the input action-set, $\beta = \{\beta_1, \beta_2, \ldots, \beta_m\}$ denotes the environment's response, and $c = \{c_1, c_2, \ldots, c_r\}$ shows the penalty probabilities.

CALA [36] is an optimization tool which is used to minimize a multivariate function. To minimize the multivariate function $f: R^m \rightarrow R$, a team with m LA is required. Each LA LA^i,$(1 \leq i \leq m)$ in this team at instant n uses two internal parameters, mean $\mu^i(n)$ and variance $\sigma^i(n)$. Each LA^i chooses an action $\alpha^i(n) \in N(\mu^i(n), \varphi[\delta^i(n)])$ where $\alpha^i(n)$ is a normal distribution with the mean $\mu^i(n)$ and standard deviation $\sigma^i(n)$. Using the generated actions and the mean value of each LA, two inputs $\alpha(n) = (\alpha^1(n), \alpha^2(n), \ldots, \alpha^m(n) \in R^m)$ and $\mu(n) = (\mu^1(n), \mu^2(n), \ldots, \mu^m(n) \in R^m)$ for computing noisy functions $\beta_\alpha(n)$ and $\beta_\mu(n)$ are described as $\beta_\alpha(n) = f(\alpha(n))$ and $\beta_\mu(n) = f(\mu(n))$. Then, each LA_i updates $\mu^i(n)$ and $\sigma^i(n)$ according (4) to (8), respectively. By repeating the learning process, each leaning automaton finds its optimal action and the final value for $\beta_\mu(n)$ is considered as the minimum value of function f or cost function [36]. The CALA algorithm is represented in Algorithm 1, where $C > 0$ shows the penalty parameter; and $0 < \lambda < 1$ shows the learning parameter and $\delta_l \in R$ represents the lower bound for the variance parameter.

$$\mu^i(n+1) = \mu^i(n) - \lambda F_1^i(\mu(n), \sigma(n), \alpha(n), \beta_\alpha(n), \beta_\mu(n)) \quad (4)$$

$$\sigma^i(n+1) - \lambda F_2^i\big(\mu(n), \sigma(n), \alpha(n), \beta_\alpha(n), \beta_\mu(n)\big) + C[\delta_l - \sigma^i(n)] \quad (5)$$

$$F_1^i\left(\mu(n), \sigma(n), \alpha(n), \beta_\alpha(n), \beta_\mu(n)\right) = \left(\frac{\beta_\alpha(n) - \beta_\mu(n)}{\emptyset[\sigma(n)]}\right)\left(\frac{\alpha^i(n) - \mu^i(n)}{\emptyset[\sigma^i(n)]}\right) \quad (6)$$

$$F_2^i\left(\mu(n), \sigma(n), \alpha(n), \beta_\alpha(n), \beta_\mu(n)\right) = \left(\frac{\beta_\alpha(n) - \beta_\mu(n)}{\emptyset[\sigma(n)]}\right)\left[\left(\frac{\alpha^i(n) - \mu_n^i}{\emptyset[\sigma^i(n)]}\right)^2 - 1\right] \quad (7)$$

$$\emptyset[\delta] = \begin{cases} \delta_l & \text{for } \delta \leq \delta_l \\ \delta & \text{for } \delta > \delta_l \end{cases}, \text{For } \delta, \delta_l \in R \quad (8)$$

The pseudo-code of the CALA algorithm is shown below.

```
Algorithm 1: CALA
  Input: $m$ // $m$ is the number of learning automata in CALA.
  Output: minimum value of $f$ // $f$: $R^m \to R$ is cost function.
// Initialize
Initialize the value of $C, \lambda$ and $\sigma_L$ /* $C$ is a penalizing constant,
    $\lambda$ is a learning rate and CTL is a lower bound.        */
$n \leftarrow 0$  // $n$ is time step.
foreach $LA^i, 1 \le i \le m$ do
    Initialize the mean and variance of the Gaussian distribution
    as $\mu^i(n)$ and $\sigma^i(n)$
end foreach
repeat
    foreach $LA^i, 1 \le i \le m$ do
        Choose an action $\alpha^i(n) \in R$ using Gaussian distribution
            $N(\mu^i(n), \emptyset[\sigma^i(n)])$. /*$\mu^i(n) \in R$ is mean and $\emptyset[\sigma^i(n)] \in R$
            is the standard deviation.
    end foreach
    // using the generating actions build $\alpha(n)$
    $\alpha^i(n) = (\alpha^1(n), \alpha^2(n), ..., \alpha^m(n)) \in R^m$
    // using the the mean value of each LA build $\mu(n)$
    $\mu(n) = (\mu^1(n), \mu^2(n), ..., \mu^m(n)) \in R^m$
    // compute two noisy functions $\beta_\alpha(n)$ and $\beta_\mu(n)$
    $\beta_\alpha(n) \leftarrow f(\alpha(n))$
    $\beta_\mu(n) \leftarrow f(\mu(n))$
    // update parameters of each LA
    foreach $LA^i, 1 \le i \le m$ do
        update $\mu^i(n)$ and $\sigma^i(n)$ according to Eqs. (4) to (8).
    end foreach
    // increase time step.
    $n \leftarrow n + 1$
until for all LA, $\mu^i(n)$ does not change noticeably and $\sigma^i(n)$
    converges to $\sigma_L$;
mínimum value of $f \leftarrow \beta_\mu(n)$
```

IV. The Proposed CALA-AFTM Algorithm

In this section, the cost function and the problem formulation of the proposed algorithm (CALA-AFTM) are defined.

A. Cost Function

To assess the quality of trapezoidal membership functions (TMFs), we modified the cost function proposed in [7] as follows:

$$Cost\ function = \frac{1}{Objective\ function} \tag{9}$$

$$Objective\ function = \frac{\sum_{X \epsilon L_1} fuzzy\ support(R)}{Suitablity} \tag{10}$$

where fuzzy support(R) shows the large 1-itemset R in L_1. Moreover, L_1 shows the large 1-itemset. The fuzzy support(R) is computed according to (3). The suitability parameter is defined as [7]:

$$Suitability = Overlap\ factor + Covrage\ factor \tag{11}$$

The overlap factor for two regions R_i and R_j ($i<j$) is defined as the region covered by two regions R_i and R_j to the minimum of ($wr_{i,4}$-c_i) and (c_j-$wl_{j,1}$). The coverage factor is defined as [7]:

$$Coverage\ factor = \frac{max\ (I_i)}{range(R_1, R_2, \ldots, R_m)} \tag{12}$$

range(R_1, R_2, ..., R_m) denotes the length of the horizontal axis, max (Ii) represents the maximum quantity of item I_i, and m represents the number of membership functions of item I_i. We changed the overlap factor shown in [7] to TMFs as follows:

$$Overlap\_factor = \sum_{i<j}(max\left(\frac{Overlap(R_i, R_j)}{min\ ((wr_{i,4}-c_i),(c_j-wl_{j,1}))}, 1\right) - 1) \tag{13}$$

Let MF_i, ($1 \le i \le m$) denotes fuzzy regions, where $MF_i = \{R_1, ..., R_i\}$. Each R_i shows a trapezoidal membership function. Each R_i is defined as ($wl_{i,1}$, $cl_{i,2}$, $cr_{i,3}$, $wr_{i,4}$), where $wl_{i,1}$ is the left spread, $cl_{i,2}$ is the left center, $cr_{i,3}$ is the right center, and $wr_{i,4}$ represents the right spread of the fuzzy region R_i. c_i shows the middle center and described in (14). In Fig. 2 the TMFs parameters are shown. An Example: Suppose that membership functions for item I_j are presented in Fig. 3.

$$c_i = \frac{cl_{i,2} + cr_{i,3}}{2} \tag{14}$$

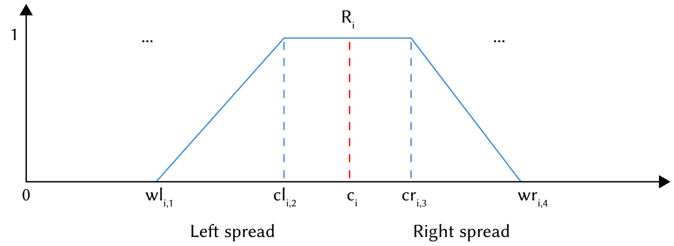


Fig. 2. Parameters of trapezoidal membership functions.

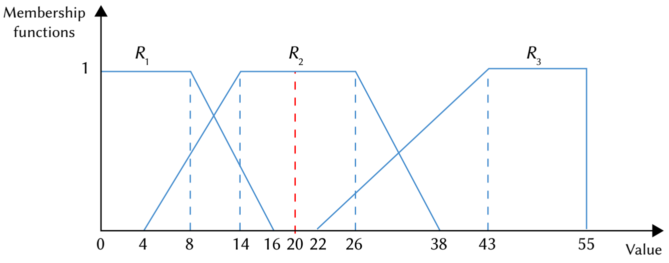


Fig. 3. Membership functions for item I_j.

In Fig. 3, item I_j has three membership functions: R_1 , R_2 and R_3. It can be observed from Fig. 3 that the overlap (R_1, R_2) = 12, the overlap (R_2, R_3) = 16, the overlap (R_1, R_3) = 0.

the minimum spread (R_1, R_2) = min ((16-0), (20-4)) = 16, the minimum spread (R_2, R_3) = min ((38-20), (55-22)) = 18, and the minimum spread (R_1, R_3) = min ((16-0), (55-22)) = 16. So, the overlap factor of item I_j is computed as follows: overlap factor = (max($\frac{12}{16}$,1)-1]+max($\frac{16}{18}$,1)-1+ max($\frac{0}{16}$,1)-1) =0+0+0=0. In this example, the range (R_1, R_2, R_3) = 55. Assume that the maximum number of items I_j in a transaction is 55, max (I_j) = 55; then, the coverage factor of item $I_j$is computed as: coverage factor = $\frac{55}{55}$ = 1. Therefore, the suitability value for the item I_j is determined as follows: suitability = 0 +1 = 1. Suppose that the number of transactions in the data set is six, and item I_j exists in five transaction data set with quantity 5, 12, 24, 30, and 46. Consequently, the fuzzy support (R_1, R_2, R_3) is computed as follows:

$$\text{Fuzzy Support } (\alpha_1) = \frac{1}{6} \times \left(1 + \frac{16\text{-}5}{16\text{-}8} + 0 + 0 + 0\right) = 0.39$$

$$\text{Fuzzy Support } (\alpha_2) = \frac{1}{6} \times \left(\frac{5\text{-}4}{14\text{-}4} + \frac{12\text{-}4}{14\text{-}4} + 1 + \frac{38\text{-}30}{38\text{-}26} + 0\right) = 0.42$$

$$\text{Fuzzy Support } (\alpha_3) = \frac{1}{6} \times \left(0 + 0 + \frac{24\text{-}22}{43\text{-}22} + \frac{30\text{-}22}{43\text{-}22} + 1\right) = 0.24$$

By considering the value 0.31 for the minimum support, the large 1-itemsets is specified as $L_1 = \{\alpha_1, \alpha_2\}$. So, the cost function computed as $\frac{1}{0.39+0.42}$=1.23.

B. Appropriate Trapezoidal Membership Functions

Each trapezoidal membership function must have two conditions, which are shown in (15) and (16), respectively.

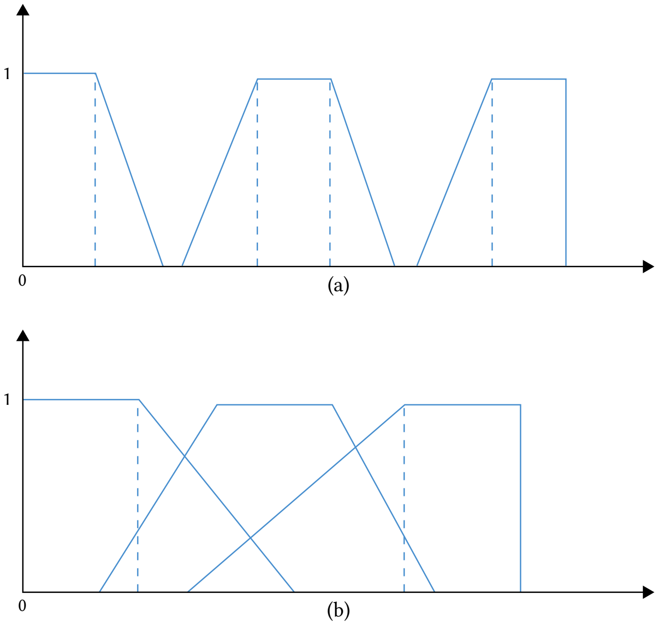
$$cr_{i,3} \le cl_{i+1,2} \le cr_{i+1,3} \le \cdots \le cl_{m-1,2} \le cr_{m-1,3} \le cl_{m,2} \tag{15}$$

$$wl_{i,1} \le cl_{i,2} \le cr_{i,3} \le wr_{i,4} \tag{16}$$

TABLE I. The Number of Suitable Membership Functions for the Three Trapezoidal Membership Functions

$V_1 = (CR_{1,3,}\ WL_{2,1}, WR_{1,4}, CL_{2,2}, CR_{2,3}, WL_{3,1}, WR_{2,4}, CL_{3,3})$	$V_{13} = (WL_{2,1}\ ,CR_{1,3}, WR_{1,4}\ , CL_{2,2}\ ,CR_{2,3}\ ,WL_{3,1}\ ,WR_{2,4}\ , CL_{3,3})$
$V_2 = (CR_{1,3}\ WL_{2,1}\ WR_{1,4}\ CL_{2,2}\ CR_{2,3}\ WL_{3,1}\ CL_{3,3}\ WR_{2,4})$	$V_{14} = (WL_{2,1}, CR_{1,3}\ ,WR_{1,4}\ ,CL_{2,2}\ ,CR_{2,3}\ ,WL_{3,1}\ ,CL_{3,3}\ ,WR_{2,4})$
$V_3 = (CR_{1,3,}\ WL_{2,1},WR_{1,4}, CL_{2,2}, WL_{3,1}, CR_{2,3}, WR_{1,4}, CL_{3,3})$	$V_{15} = (WL_{2,1}\ ,CR_{1,3}, WR_{1,4}\ , CL_{2,2}, WL_{3,1,}\ CR_{2,3,}\ WR_{2,4}\ , CL_{3,3})$
$V_4 = (CR_{1,3}\ , WL_{2,1}\ ,WR_{1,4}\ ,CL_{2,2}\ ,WL_{3,1}\ ,CR_{2,3}, CL_{3,3}\ , WR_{2,4})$	$V_{16} = (WL_{2,1}\ ,CR_{1,3}, WR_{1,4}\ ,CL_{2,2}\ ,WL_{3,1}\ ,CR_{2,3}\ ,CL_{3,3}\ ,WR_{2,4})$
$V_5 = (CR_{1,3}, WL_{2,1}\ ,WR_{1,4}\ , WL_{3,1}\ ,CL_{2,2}, CR_{2,3}\ ,WR_{2,4}\ ,CL_{3,3})$	$V_{17} = (WL_{2,1}\ , CR_{1,3}, WR_{1,4}\ ,WL_{3,1}, CL_{2,2}, CR_{2,3}, WR_{2,4}, CL_{3,3})$
$V_6 = (CR_{1,3}\ ,WL_{2,1}, WR_{1,4}\ ,WL_{3,1}, CL_{2,2}\ ,CR_{2,3}, CL_{3,3}, WR_{2,4})$	$V_{18} = (WL_{2,1}\ , CR_{1,3}, WR_{1,4}, WL_{3,1}, CL_{2,2}, CR_{2,3}, CL_{3,3}\ , WR_{2,4})$
$V_7 = (CR_{1,3}\ , WL_{2,1}, CL_{2,2}\ ,WR_{1,4}, CR_{2,3}\ ,WL_{3,1}\ ,WR_{2,4}\ , CL_{3,3})$	$V_{19} = (WL_{2,1}, CR_{1,3}, CL_{2,2}\ , WR_{1,4}, CR_{2,3,}\ WL_{3,1,}\ WR_{2,4}\ , CL_{3,3})$
$V_8 = (CR_{1,3}\ , WL_{2,1}, CL_{2,2}\ ,WR_{1,4}, CR_{2,3}, WL_{3,1}, CL_{3,3}, WR_{2,4})$	$V_{20} = (WL_{2,1}\ , CR_{1,3}, CL_{2,2}\ , WR_{1,4}, CR_{2,3}\ ,WL_{3,1}, CL_{3,3}, WR_{2,4})$
$V_9 = (CR_{1,3}, WL_{2,1}\ ,CL_{2,2}\ , WR_{1,4}\ ,WL_{3,1}\ ,CR_{2,3}\ ,WR_{2,4}\ , CL_{3,3})$	$V_{21} = (WL_{2,1}, CR_{1,3}, CL_{2,2}\ ,WR_{1,4}, WL_{3,1,}\ CR_{2,3}\ ,WR_{2,4}\ , CL_{3,3})$
$V_{10} = (CR_{1,3}\ , WL_{2,1,}\ CL_{2,2}\ ,WR_{1,4}\ ,WL_{3,1}\ ,CR_{2,3}, CL_{3,3}\ , WR_{2,4})$	$V_{22} = (WL_{2,1}\ ,CR_{1,3}, CL_{2,2}, WR_{1,4}\ ,WL_{3,1}, CR_{2,3}, CL_{3,3}\ , WR_{2,4})$
$V_{11} = (CR_{1,3}, WL_{2,1}\ ,CL_{2,2}\ ,CR_{2,3}\ ,WR_{1,4}\ ,WL_{3,1}, WR_{2,4}\ , CL_{3,3})$	$V_{23} = (WL_{2,1}\ ,CR_{1,3}, CL_{2,2}\ ,CR_{2,3}\ ,WR_{1,4}, WL_{3,1}\ ,WR_{2,4}\ , CL_{3,3})$
$V_{12} = (CR_{1,3}\ ,WL_{2,1}\ ,CL_{2,2}\ ,CR_{2,3,}\ WR_{1,4}, WL_{3,1}, CL_{3,3}\ , WR_{2,4})$	$V_{24} = (WL_{2,1}\ ,CR_{1,3}\ ,CL_{2,2}\ ,CR_{2,3}\ ,WR_{1,4}\ ,WL_{3,1}, CL_{3,3}\ , WR_{2,4})$

The first condition maintains the order of the trapezoidal centers, and the second condition preserves the trapezoidal shape. Fig. 4a and Fig. 4b shows two examples of unsuitable membership functions.



(a)

(b)

Fig. 4. Two sets of unsuitable membership functions.

By considering (15) and (16) the search space is greatly reduced. Additionally, (17) and (18) satisfy the complete coverage, and (19) satisfies the appropriate overlap.

$$wl_{i,1} \leq wl_{i,1} \leq cwr_{i-1,4} \tag{17}$$

$$wl_{i+1,1} \leq wr_{i,4} \leq wr_{i+1,4} \tag{18}$$

$$wl_{i,4} \leq wl_{i+2,1} \tag{19}$$

To reduce the domain of the search space and eliminate improper membership functions a new approach was proposed. The proposed approach is depicted below: Each $R_i,(1 \leq i \leq m)$ is illustrated by quadruple $(wl_{i,1}, cl_{i,2}, cr_{i,3}, wr_{i,4})$. Let $P = \{cr_{1,3}, wl_{2,1}, wr_{1,4}, \ldots, wl_{i,1}, wr_{i-1,4}, cl_{i,2}\ ,cr_{i,3}, wr_{i,4}, \ldots, wl_{m-1,1}, cl_{m-1,2}, cr_{m-1,3}, wl_{m,1}, wr_{m-1,4}, cl_{m,2}\}$ be the set of all TMFs parameters.

Various permutations can be considered from this set. However, many of these permutations will be invalid because they do not maintain the trapezoidal shape. For each R_i, any non-duplicate Cartesian product is determined as a suitable membership function which is shown in (20).

$$\{Wr_{i\text{-}1,4}\ ,cl_{I,2}\} \times \{wr_{i\text{-}1,4}\ ,cl_{I,2},cr_{I,3},wl_{i+1,1}\} \times \{wr_{i\text{-}1,4}\ ,cl_{I,2},cr_{I,3},wl_{i+1,1}\} \times \{cr_{i,3}\ ,wl_{i+1,1}\} \tag{20}$$

By considering the Cartesian product $R_1 \times R_2 \times \ldots \times R_m$ and removing non-repetitive sequences, the final suitable trapezoidal membership functions are determined. For example, for three TMFs, we have 12! = 479, 001,600 sequences. While the proposed method produces only 24 membership functions ($L_3 = \{v_1, v_2, \ldots, v_{24}\}$) as valid membership functions. Where each v_i shows the points of trapezoidal membership functions. In Table I all 24 valid membership functions are shown.

C. Representation of the Learning Automata

In Fig. 5, the representation of learning automata for the team of CALA is given. Let k_{max} be the maximum number of fuzzy regions which is determined by the user. To build a team of CALA, we require $k_{max} + \sum_{i=1}^{k_{max}} \times k_{max}$ number of LA. The first k_{max} LA is used to specify the active TMFs. These LA are labeled as $LA^1, LA^2, \ldots, LA^{k_{max}}$. The actions generated by each LA of $LA^i, (1 \leq i \leq k_{max})$ are considered as $\alpha^i(n) \in N(\mu^i(n), \varphi[\delta^i(n)])$, where $\alpha^i(n)$ is a normal distribution with mean $\mu^i(n)$ and standard deviation $\varphi[\delta^i(n)]$. Let $MF_i, (1 \leq i \leq k_{max})$ consist of i fuzzy regions (i linguistic terms), where $MF_i = \{R_1, \ldots, R_i\}$. Each MF_i includes i fuzzy regions, and each fuzzy region in TMFs is determined by four parameters. Consequently, to represent each MF_i, $i \times 4$ learning automata are required. Each fuzzy region R_j, $(1 \leq j \leq i)$ of MF_i is equipped with four LA which are labeled as $LA_k^{i,j}$, $(1 \leq i \leq k_{max}, 1 \leq j \leq i$ and $k \in \{wl, cl, cr, wr\})$. The corresponding actions for LA $LA_k^{i,j}$ are shown by $\alpha_k^{i,j}(n) \in N[\mu_k^{i,j}(n), \varphi_k^{i,j}(n)]$.

D. Defining the Action-Set and Generating Actions

In the proposed method, to increase the speed of convergence using the maximum value of the dataset, all values are normalized between 0 and 1. The action-set of $LA^i, (1 \leq i \leq k_{max})$, is considered as a $\alpha_i(n) \in N(0,1)$. For each automaton $LA_k^{i,j}$ the action-set is defined as $\alpha_k^{i,j}(n) \in N[0,1]$. Thus, to generate the actions at instant n, all LA of CALA, namely LA^i and $LA_k^{i,j}$, choose an action $\alpha^i(n) \in N[0,1]$ and $\alpha_k^{i,j}(n) \in N[0,1]$, respectively. The actions generated at instant n are denoted by (21). Also, using the generated $\alpha^i(n)$ and the mean value of each LA, $\mu(n)$ is determined as (22).

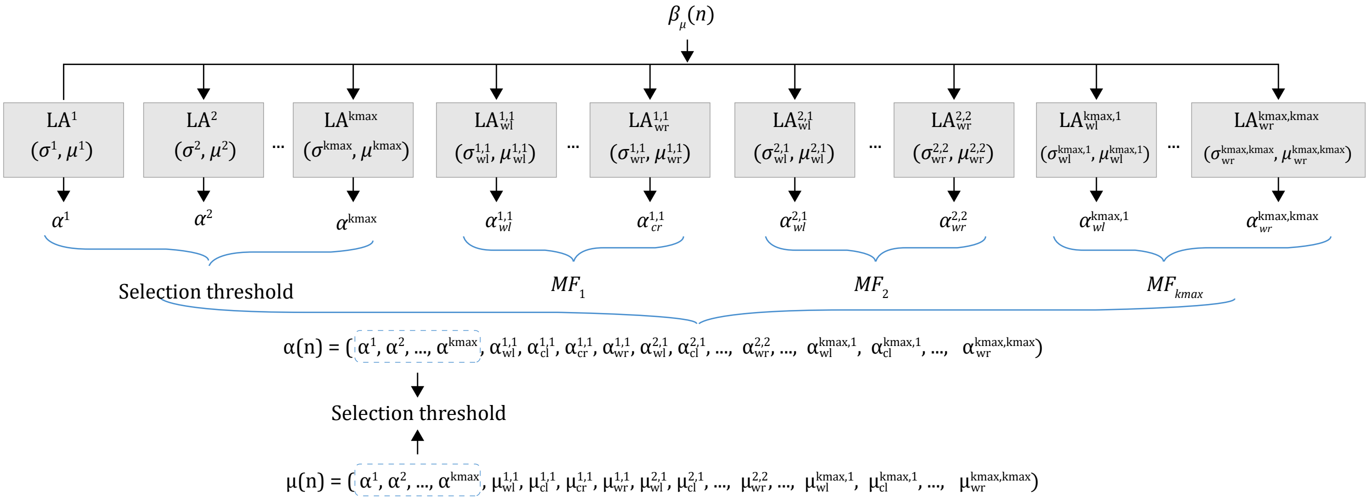

$\beta_\mu(n)$
LA^1 (σ^1, μ^1)
LA^2 (σ^2, μ^2)
...
LA^{kmax} $(\sigma^{kmax}, \mu^{kmax})$
$LA^{1,1}_{wl}$ $(\sigma^{1,1}_{wl}, \mu^{1,1}_{wl})$
...
$LA^{1,1}_{wr}$ $(\sigma^{1,1}_{wr}, \mu^{1,1}_{wr})$
$LA^{2,1}_{wl}$ $(\sigma^{2,1}_{wl}, \mu^{2,1}_{wl})$
...
$LA^{2,2}_{wr}$ $(\sigma^{2,2}_{wr}, \mu^{2,2}_{wr})$
$LA^{kmax,1}_{wl}$ $(\sigma^{kmax,1}_{wl}, \mu^{kmax,1}_{wl})$
...
$LA^{kmax,kmax}_{wr}$ $(\sigma^{kmax,kmax}_{wr}, \mu^{kmax,kmax}_{wr})$
α^1
α^2
α^{kmax}
$\alpha^{1,1}_{wl}$
$\alpha^{1,1}_{cr}$
$\alpha^{2,1}_{wl}$
$\alpha^{2,2}_{wr}$
$\alpha^{kmax,1}_{wl}$
$\alpha^{kmax,kmax}_{wr}$
Selection threshold
MF_1
MF_2
MF_{kmax}
$\alpha(n) = (\alpha^1, \alpha^2, ..., \alpha^{kmax}, \alpha^{1,1}_{wl}, \alpha^{1,1}_{cl}, \alpha^{1,1}_{cr}, \alpha^{1,1}_{wr}, \alpha^{2,1}_{wl}, \alpha^{2,1}_{cl}, ..., \alpha^{2,2}_{wr}, ..., \alpha^{kmax,1}_{wl}, \alpha^{kmax,1}_{cl}, ..., \alpha^{kmax,kmax}_{wr})$
Selection threshold
$\mu(n) = (\alpha^1, \alpha^2, ..., \alpha^{kmax}, \mu^{1,1}_{wl}, \mu^{1,1}_{cl}, \mu^{1,1}_{cr}, \mu^{1,1}_{wr}, \mu^{2,1}_{wl}, \mu^{2,1}_{cl}, ..., \mu^{2,2}_{wr}, ..., \mu^{kmax,1}_{wl}, \mu^{kmax,1}_{cl}, ..., \mu^{kmax,kmax}_{wr})$

Fig. 5. Representation of CALA in the proposed algorithm.

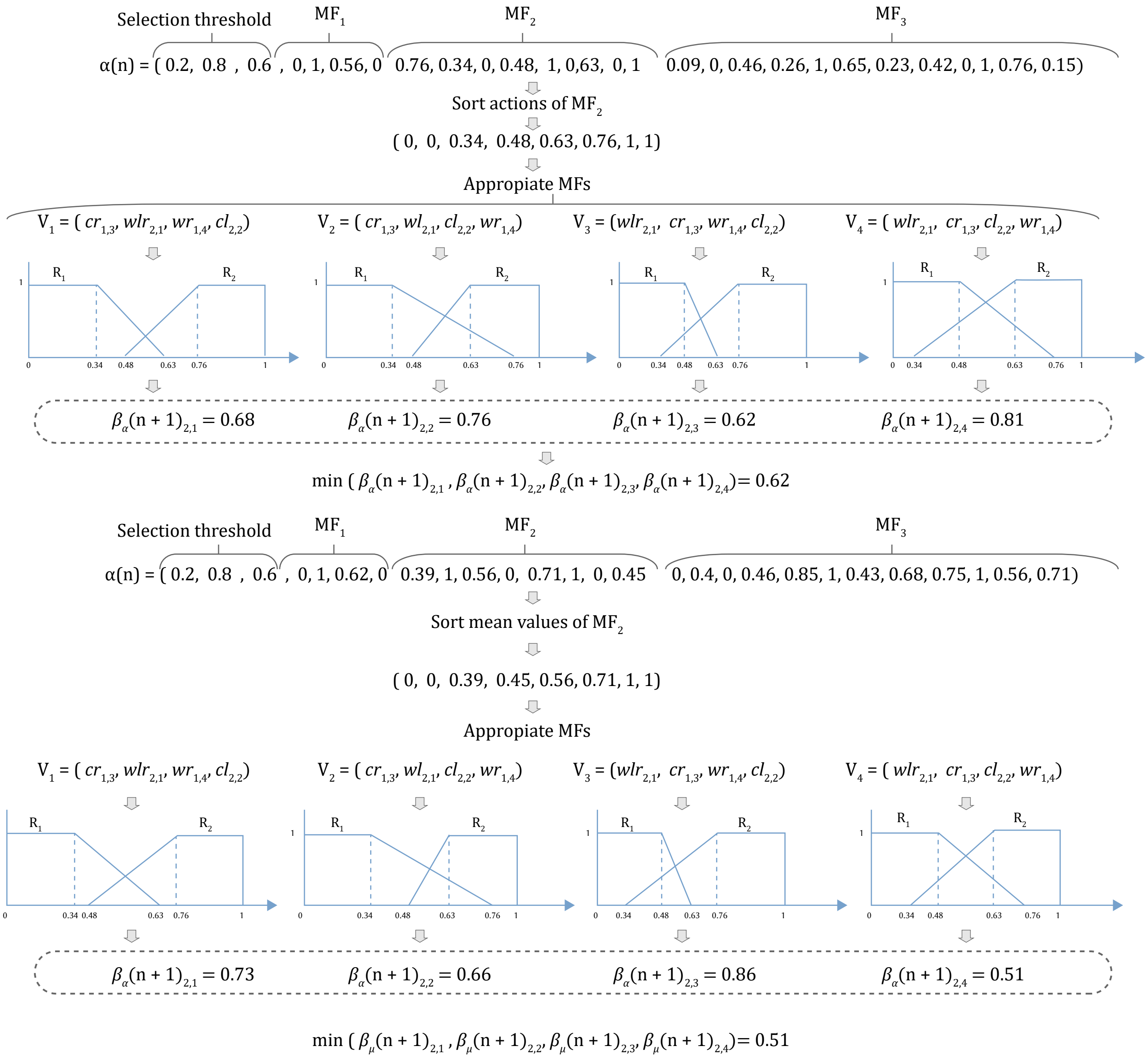

Selection threshold
MF_1
MF_2
MF_3
$\alpha(n) = (0.2, 0.8 , 0.6 , 0, 1, 0.56, 0 \; 0.76, 0.34, 0, 0.48, 1, 0,63, 0, 1 \; 0.09, 0, 0.46, 0.26, 1, 0.65, 0.23, 0.42, 0, 1, 0.76, 0.15)$
Sort actions of MF_2
(0, 0, 0.34, 0.48, 0.63, 0.76, 1, 1)
Appropiate MFs
$V_1 = (cr_{1,3}, wlr_{2,1}, wr_{1,4}, cl_{2,2})$
$V_2 = (cr_{1,3}, wl_{2,1}, cl_{2,2}, wr_{1,4})$
$V_3 = (wlr_{2,1}, cr_{1,3}, wr_{1,4}, cl_{2,2})$
$V_4 = (wlr_{2,1}, cr_{1,3}, cl_{2,2}, wr_{1,4})$
R_1
R_2
1
0
0.34
0.48
0.63
0.76
1
$\beta_\alpha(n + 1)_{2,1} = 0.68$
$\beta_\alpha(n + 1)_{2,2} = 0.76$
$\beta_\alpha(n + 1)_{2,3} = 0.62$
$\beta_\alpha(n + 1)_{2,4} = 0.81$
$\min (\beta_\alpha(n + 1)_{2,1} , \beta_\alpha(n + 1)_{2,2}, \beta_\alpha(n + 1)_{2,3}, \beta_\alpha(n + 1)_{2,4})= 0.62$
Selection threshold
MF_1
MF_2
MF_3
$\alpha(n) = (0.2, 0.8 , 0.6 , 0, 1, 0.62, 0 \; 0.39, 1, 0.56, 0, 0.71, 1, 0, 0.45 \; 0, 0.4, 0, 0.46, 0.85, 1, 0.43, 0.68, 0.75, 1, 0.56, 0.71)$
Sort mean values of MF_2
(0, 0, 0.39, 0.45, 0.56, 0.71, 1, 1)
Appropiate MFs
$V_1 = (cr_{1,3}, wlr_{2,1}, wr_{1,4}, cl_{2,2})$
$V_2 = (cr_{1,3}, wl_{2,1}, cl_{2,2}, wr_{1,4})$
$V_3 = (wlr_{2,1}, cr_{1,3}, wr_{1,4}, cl_{2,2})$
$V_4 = (wlr_{2,1}, cr_{1,3}, cl_{2,2}, wr_{1,4})$
R_1
R_2
1
0
0.34
0.48
0.63
0.76
1
$\beta_\alpha(n + 1)_{2,1} = 0.73$
$\beta_\alpha(n + 1)_{2,2} = 0.66$
$\beta_\alpha(n + 1)_{2,3} = 0.86$
$\beta_\alpha(n + 1)_{2,4} = 0.51$
$\min (\beta_\mu(n + 1)_{2,1} , \beta_\mu(n + 1)_{2,2}, \beta_\mu(n + 1)_{2,3}, \beta_\mu(n + 1)_{2,4})= 0.51$

Fig. 6. An example of finding noisy function values in the proposed algorithm.

$$\alpha(n) = \left(\begin{array}{c} \alpha^1, \alpha^2, \ldots, \alpha^{kmax}, \\ \alpha_{wl}^{1,1}, \alpha_{cl}^{1,1}, \alpha_{cr}^{1,1}, \alpha_{wr}^{1,1}, \\ \ldots, \alpha_{wl}^{kmax,1}, \alpha_{cl}^{kmax,1}, \alpha_{cr}^{kmax,1}, \\ \alpha_{wr}^{kmax,1}, \ldots, \alpha_{wl}^{kmax,kmax}, \alpha_{cl}^{kmax,kmax}, \\ \alpha_{cr}^{kmax,kmax}, \alpha_{wr}^{kmax,kmax} \end{array} \right) \tag{21}$$

$$\mu(n) = \left(\begin{array}{c} \alpha^1, \alpha^2, \ldots, \alpha^{kmax}, \\ \mu_{wl}^{1,1}, \mu_{cl}^{1,1}, \mu_{cr}^{1,1}, \mu_{wr}^{1,1}, \\ \ldots, \mu_{wl}^{kmax,1}, \mu_{cl}^{kmax,1}, \mu_{cr}^{kmax,1}, \\ \mu_{wr}^{kmax,1}, \ldots, \mu_{wl}^{kmax,kmax}, \mu_{cl}^{kmax,kmax}, \\ \mu_{cr}^{kmax,kmax}, \mu_{wr}^{kmax,kmax} \end{array} \right) \tag{22}$$

E. Modification of the Generating Actions

After generating the actions, if the value of all action $\alpha^i(n)$ and $\alpha_k^{i,j}(n)$ chosen by each automaton is less than 0 or greater than 1, reinitialize it with a value between 0 and 1. Also, if the values of all chosen actions of $\alpha^i(n)$ are less than 0.5, we randomly choose an action $\alpha^i(n)$ and reinitialize it with a value between 0.5 and 1.

F. Finding the Active TMFs

Using the generated actions for the first kmax automaton, the active TMFs at instant n are determined. If i is the number of actions with values greater than 0.5, then the membership functions MFi and the corresponding actions, $\alpha_{wl}^{i,1}, \alpha_{cl}^{i,1}, \alpha_{cr}^{i,1}, \alpha_{wr}^{i,1}, \ldots, \alpha_{wl}^{i,j}, \alpha_{cl}^{i,j}, \alpha_{cr}^{i,j}, \alpha_{wr}^{i,j}$ are activated.

G. Evaluation of the Noisy Functions

After computing α(n) and μ(n), the next step is to evaluate the values of the noisy function to find the values of $\beta_\alpha(n)$ and $\beta_\mu(n)$. CALA at instant n needs to compute two functions, namely, $\beta_\alpha(n) = f(\alpha(n))$ and $\beta_\mu(n) = f(\mu(n))$. To compute the value of $\beta_\alpha(n)$, perform the following steps:

1. Find the active TMFs (see Section F).
2. For active TMFs such as MF_i, the corresponding actions, $\alpha_{wl}^{i,1}, \alpha_{cl}^{i,1}, \alpha_{cr}^{i,1}, \alpha_{wr}^{i,1}, \ldots, \alpha_{wl}^{i,j}, \alpha_{cl}^{i,j}, \alpha_{cr}^{i,j}, \alpha_{wr}^{i,j}$ are assigned in a vector, and their value is sorted in the ascending order.
3. Using the appropriate list for each MF_i and considering the sorted actions, TMFs for each element of valid membership functions are constructed.
4. For each constructed trapezoidal membership function, the value of $\beta_\alpha(n)$ is determined using the cost function.
5. For active membership functions such as MFi, the minimum value of $\beta_\alpha(n)$ from among all appropriate membership functions is considered as the best solution at instant n.

$\beta_\mu(n)$ is calculated similarly to $\beta_\alpha(n)$, with the difference that μ(n) is used here.

H. Updating the Parameters of Each Learning Automaton

After determining the value of $\beta_\alpha(n)$ and $\beta_\mu(n)$, each LA of CALA updates its parameters, such as mean μ(n) and variance σ(n). The updating rule for the LA of CALA is as follows: Each LA^i, $(1, \leq, i, \leq k_{max})$ updates the μ(n) and σ(n). In this case, if $\mu^i(n)$ less than 0 or greater than 1, then, then its value is reinitialized with 0 and 1, respectively. In addition, each automaton of $LA_k^{i,j}$ updates its internal permeates of μ(n) and σ(n) using (4) and (5). In this case, the reinitialization of $\mu_k^{i,j}$ is not performed.

Fig. 6 shows an example of computing $\beta_\alpha(n)$ and $\beta_\mu(n)$. Assume that, at instant n, the action values for the two vectors α(n) and μ(n) are generated according to (4) and (5) which is a normal distribution with the mean value of 0 and standard deviation of 1. In this example, k_{max} is set at 3, so the team of CALA uses 27 learning automata. According to the number of actions for threshold values > 0.5, it is clear that MF2 can be active. According to the action values generated for MF2, we sort their values and then the TMFs are constructed for each vector by considering the appropriate membership functions that are specified by the four vectors v1to v4. Subsequently, the value of each vector is evaluated using the cost function. In this example, $\beta_\alpha(n)_{i,j}$ is used to represent the j-th appropriate membership function for MF_i. By considering the minimum value obtained among four membership functions, the value 0.62 will be selected as $\beta_\alpha(n)$. The procedure of computing $\beta_\mu(n)$ is similar to $\beta_\alpha(n)$, except that vector $\beta_\mu(n)$ is used instead of $\beta_\alpha(n)$. In this case, the value of 0.51 computed for $\beta_\mu(n)$. It is noteworthy that the value of $\beta_\mu(n)$ will be utilized as the reinforcement signal.

I. Pseudo-Code for the CALA-AFTM

In Algorithm 2, the pseudo-code of CALA-AFTM algorithm is presented. The CALA-AFTM algorithm uses four functions, as presented in Algorithms 3 to 6.

```
Algorithm 2: CALA-AFTM
  Input: $K_{max}$   //$K_{max}$ is the maximum number of fuzzy regions.
  Output: Number and positions of optimized trapezoidal
          membership functions.
  // Save appropriate membership functions.
1 for $i \leftarrow 1$ to $K_{max}$ do
2     $L_i = ValidMernbershipFunctios(i)$
3 end for
  // Build CALA and initialize parameters.
4 Build a tcam of CALA with thc sizc of $K_{max} + \sum_{i=1}^{K_{max}} (i \times K_{max})$ LA.
5 For each LA of CALA, initialize the parameters of C, $\sigma$, $\lambda$, $\sigma_L$ and $\mu$.
6 n ← 0, ActivcMF ← 0, TempAction[]
7 repeat
    // steps for computing $\beta_\alpha(n)$
    // Generate actions of $\alpha(n)$.
8   All LA of CALA, namely $LA^i$ and $LA_k^{i,j}$, ($1 \leq i \leq K_{max}$, $1 \leq j \leq i$ and
    $k \in \{w_l, c_l, c_r, w_r\}$) choose an action $\alpha^i(n) \in N(0, 1)$ and
    $\alpha_k^{i,j}(n) \in N(0, 1)$ respectively.
    // Check all selected actions are in the ranga of [0,1]
9   Modify Actions()
    // Count number of active membership functions.
10  ActiveMF ← FindActiveMF()
    // Copy all associated actions of ActiveMF.
11  TempAction ⟵ Copy all actions of $MF_{ActiveMF}$
12  TempAction ⟵ Sort(TempAction)
13  Call modifyActions($MF_{ActiveMF}$)
    /* assume, each item of any L; correspond with one
    appropriate membership function.                   */
14   for $i \leftarrow 1$ to $L_{ActiveMF}$.Size do
15       $v_i$ ← item(i)
16       Assign each value of TempAction to each elemcnt of $v_i$
         respectively.
17       TMF ← Build trapezoidal membership function using $v_i$.
18       $\beta_\alpha(n)_{ActiveMF,i}$ ← apply objective function to the TMF.
19   end for
20   $\beta_\alpha(n) = min\ (\beta_\alpha(n)_{ActiveMF,i}, 1 \leq i \leq L_{ActiveMF}.Size)$
    // steps for computing $\beta_\mu(n)$
21   The steps for computing $\beta_\mu(n)$ is similar to $\beta_\alpha(n)$ with
     difference that the steps (10) and (11) are eliminated and in
     step (13) instead of $\alpha(n)$ the $\mu(n)$ is used.
    // Update parameters of LA
22   Update()
23   $n \leftarrow n + 1$
24 until for each LA of CALA, the μ value docs not change noticeably
   and the value $\sigma$ converge to $\sigma_L$;
```

Algorithm 3: ValidMembershipFunctions

```
Input: m                    // m is the number of fuzzy regions.
Output: $L$      // $L$ is the list of valid membership functions.
$L = [\,]$
$R_1 = \{cr_{1,3}, wr_{1,4}\}$                   // $R_1$, is first fuzzy region.
$R_m = \{wl_{m1}, cl_{m,2}\}$                    // $R_m$, is last fuzzy region.
if $m = 1$ then
    $L \leftarrow R_1$
else if $m = 2$ then
    $L \leftarrow R_1 \times R_2$  // The product Cartesian of $R_1 \times R_2$ is assigned
                       in L.
else
    for $L \leftarrow 2$ to $m - 1$ do
        $R_i \leftarrow \{wr_{i-1,4}, cl_{i,2}\} \times \{wr_{i-1,4}, cl_{i,2}, cr_{i,3}, \cdot wl_{i+1,1}\} \times$
        $\{wr_{i-1,4}, cl_{i,2}, cr_{i,3}, \cdot wl_{i+1,1}\} \times \{cr_{i,3}, \cdot wl_{i+1,1}\}$
    end for
    // The product Cartesian of $R_1 \times R_2, ..., R_m$ is assigned in L.
    $L \leftarrow R_1 \times R_2 \times ... \times R_m$
end if
return L
```

Algorithm 4: FindActiveMF

```
Output: ActiveMF
// Count number of active membership functions.
ActiveMF ← 0
for $i \leftarrow 1$ to $K_{max}$ do
    if $\alpha^i(\mathrm{n}) \geq 0.5$ then
        ActiveMF ← ActiveMF + 1
    end if
end for
return ActiveMF
```

Algorithm 5: ModifyActions

```
// Check all selected actions are in the range of [0,1]
foreach chosen action $\alpha^i(\mathrm{n})$ and $\alpha_k^{i,j}(\mathrm{n})$ do
    if its value is lcss than 0 ar greater the 1 then
        reinitialize it with the value between [0,1].
    end if
end foreach
// at least one action greater than 0.5 is exist.
if value of all chosen actions of $\alpha^i(\mathrm{n}),(1 \leq i \leq K_{max})$ is less than 0.5
then
    randomly chose an action $\alpha^i(\mathrm{n})$ and reinitialize it with a value
    between [0.5,1].
end if
```

V. Experiments and Analysis of Results

A. Experimental Setting

We conducted several experiments to show the effectiveness of the proposed approach. All algorithms implemented in Java on a 1.80 GHz Intel Core i7 processor with Windows 10 and 16 GB RAM. Two real data sets were used to assess the proposed CALA-AFTM, namely the DePaul CTI dataset [75] and NASA dataset [76]. These datasets were selected because the time parameter can be represented as TMFs [29],[44]. All HTTP requests in the NASA dataset from 23:59:59 PM on August 3, 1995 to 23:59:59 PM on August 31, 1995 were collected by NASA Kennedy Space Center in Florida. This dataset includes 45464 user sessions and 863 page views. The CTI data set was collected by the users who visited the website during two weeks in April 2002. The CTI dataset after cleaning data contained 13745 sessions and 683 pages. Part of CTI dataset is depicted in Table II. In Table II, the page view and the related page are shown. In Table III, for each user the browsing sequences are represented.

TABLE II. Page View and Corresponding Pageview IDs on the CTI Dataset

Page view Id	Page view
0	/admissions/
1	/admissions/career.asp
2	/admissions/checklist.asp
3	/admissions/costs.asp
⋮	
681	/shared/404.asp?404; http://www.cs.depaul.edu/msoffice/cltreq.asp
682	/shared/404.asp?404; http://www.cs.depaul.edu/resources/grad_scholarships.asp

TABLE III. The Browsing Sequences on the CTI Dataset

Client ID	Browsing sequence
1	(679, 2) (574, 7) (585, 5) (604, 4)
2	(387, 37) (558, 20)
3	(387, 24) (400, 125) (71, 26) (228, 34)
…	…
13563	(54, 11) (358, 55)

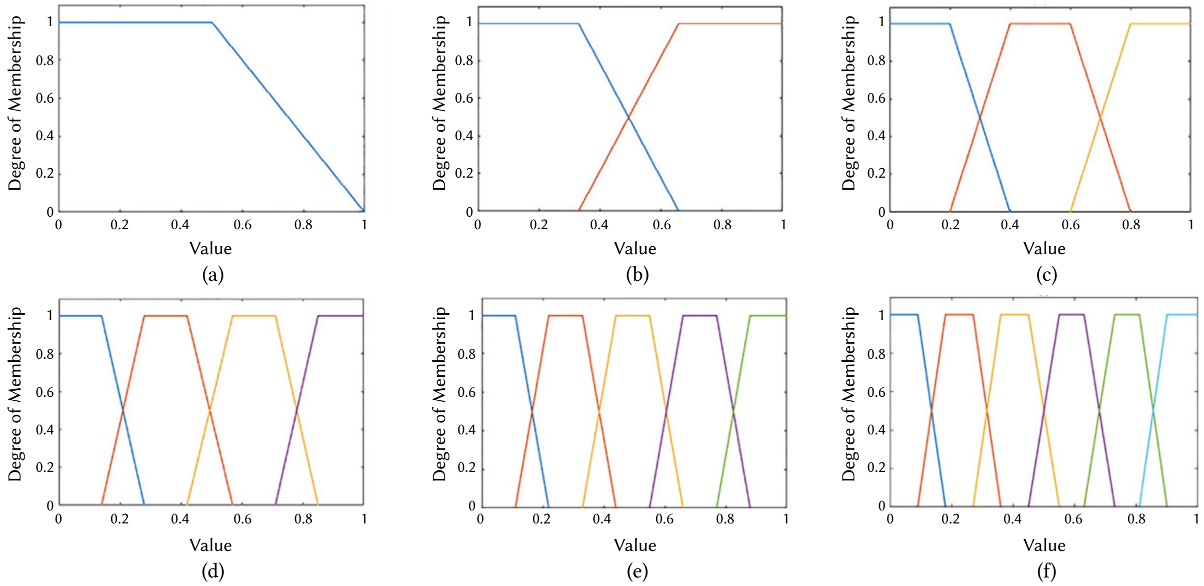
To improve the convergence speed and generate actions between 0 and 1, we used the maximum value of time duration stored in two datasets and normalized these datasets values. In the proposed algorithm, for CTI and NASA datasets, web page number 387 (/news/default.asp) and web page number 588 (/shuttle/resources/orbiters/challenger) are considered for finding their appropriate membership functions, respectively.

The proposed CALA-AFTM algorithm compared with fuzzy web mining algorithm (FWMA) [31] and VSLA-AFTM algorithm. FWMA is used for fuzzy web mining applications and uses a predefined TMFs. Additionally, we developed another algorithm called VSLA-AFTM which uses VSLA to evaluate the results. The details of the implementation of VSLA-AFTM and CALA-AFTM are similar, and their differences are described below.

All learning automaton of CALA-AFTM used the continuous action-set, whereas the learning automaton of VSLA-AFTM except for the learning automata used in the threshold section used the discrete action-set. Therefore, finding the active membership functions in both algorithms is similar. VSLA-AFTM randomly selects an action.

Let r shows the number of actions, then the chosen action $\alpha_i(1 \leq i \leq r)$ corresponding to the value of $\frac{i-1}{r-1}$. CALA-AFTM algorithm needs to compute $\beta_\alpha(n)$ and $\beta_\mu(n)$, while the VSLA-AFTM algorithm only requires to compute $\beta_\alpha(n)$. The final value for $\beta_\mu(n)$ in CALA-AFTM is regarded as the best value for the cost function, while in VSLA-AFTM, the final value for $\beta_\alpha(n)$ is taken as the best value for the cost function. In VSLA-AFTM, if the value of $\beta_\alpha(n)$ is smaller than the previous step, all chosen actions are rewarded; otherwise, they are penalized according to (4) and (5), respectively.

Fig. 7 shows the predefined membership functions used in the experiments. To assess the results, parameters such as overlap, suitability, fuzzy support, coverage, the average value of cost function (AVCF), number of large 1-sequences (L_1), and execution time were considered. Each algorithm was run independently 30 times, and the mean and standard deviation of these 30 runs were considered. Table IV represents the parameter settings used in these algorithms.



(a) (b) (c) (d) (e) (f)

Fig. 7. The fixed trapezoidal membership functions used in experiments.

TABLE IV. Parameter Settings

Parameter Name	Description	CALA-AFTM	VSLA-AFTM	FWMA
c	Penalty Constant	5		
σ_L	Lower bound for the variance	0.01		
λ	Step size for learning	0.02		
α	Minimum support	[0.002 – 0.01]	[0.002 – 0.01]	[0.002 – 0.01]
a	Reward		0.1	
b	Penalty		0.01	
r	Number of actions		[10, 20, 30, 40]	
k_{max}	Maximum number of MFs	6	6	6

TABLE V. Appropriate Number of TMFs, Coverage, Fuzzy Support, Suitability, Overlap, and Average Value of Cost Function Acquired From CALA-AFTM, VSLA-AFTM, and FWMA on the CTI Dataset

Algorithm	Optimal number of TMFs	Overlap	Coverage	Suitability	Fuzzy support	AVCF
CALA-AFTM	3.01±0.0100	0.0050±0.0023	1.0000±0.0000	1.0050±0.0023	1.4761±0.0864	0.6763±0.0389
VSLA-AFTM	4.13±0.3000	0.0674±0.0045	1.0000±0.0000	1.0674±0.0045	0.8667±0.0289	1.2321±0.0873
FWMA	3±0.0000	0.0000±0.0000	1.0000±0.0000	1.0000±0.0000	0.6524±0.0145	1.5328±0.0000

TABLE VI. Appropriate Number of TMFs, Coverage, Fuzzy Support, Suitability, Overlap and Average Value of Cost Function Acquired From CALA-AFTM, VSLA-AFTM, and FWMA on the NASA Dataset

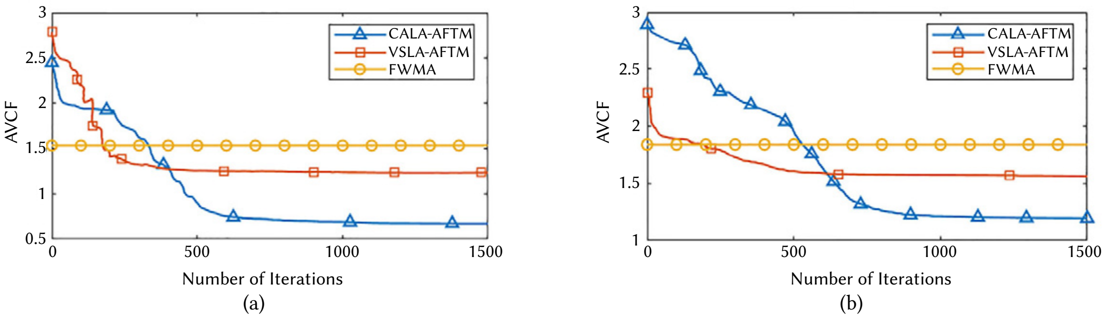
Algorithm	Optimal number of TMFs	Overlap	Coverage	Suitability	Fuzzy support	AVCF
CALA-AFTM	4.15±0.2101	0.0452±0.0053	1.0000±0.0000	1.0452±0.0053	0.8742±0.0367	1.1923±0.1132
VSLA-AFTM	2.20±0.3420	0.1476±0.0075	1.0000±0.0000	1.1476±0.0075	0.7566±0.0158	1.5652±0.1745
FWMA	3±0.0000	0.0000±0.0000	1.0000±0.0000	1.0000±0.0000	0.5416±0.0236	1.8463±0.0000

B. Experimental Evaluations

In this experiment, we evaluated the results obtained using the CALA-AFTM, VSLA-AFTM, and FWMA algorithms on the CTI and NASA datasets. Parameters such as the optimal number of TMFs, fuzzy support, overlap, coverage, suitability, and AVCF were used to evaluate the results. The results of Tables V and VI shows that the proposed CALA-AFTM for NASA and CTI datasets produced three and four TMFs, respectively. VSLA-AFTM for CTI and NASA datasets produced four and two TMFs, respectively. FWMA algorithm used a fixed number of TMFs. So, by checking each $k_{max} \in [2,6]$, it is found that the optimal number of TMFs is three.

By comparing the results, it can be seen that CALA-AFTM algorithm for the parameters AVCF, fuzzy support, and overlap has produced better results. CALA-AFTM in the CTI datasets improved the value of AVCF and fuzzy support by 51% and 49%, respectively. Additionally, in the NASA dataset, CALA-AFTM improved the value of AVCF and fuzzy support by 30% and 26% respectively. The AVCF for both CTI and NASA datasets is depicted in Fig. 8a, b. Additionally, the results show that CALA-AFTM generated the minimum value of AVCF in the two datasets. Additionally, since the VSLA-AFTM algorithm uses a small number of actions, the convergence speed of VSLA-AFTM is faster than CALA-AFTM.

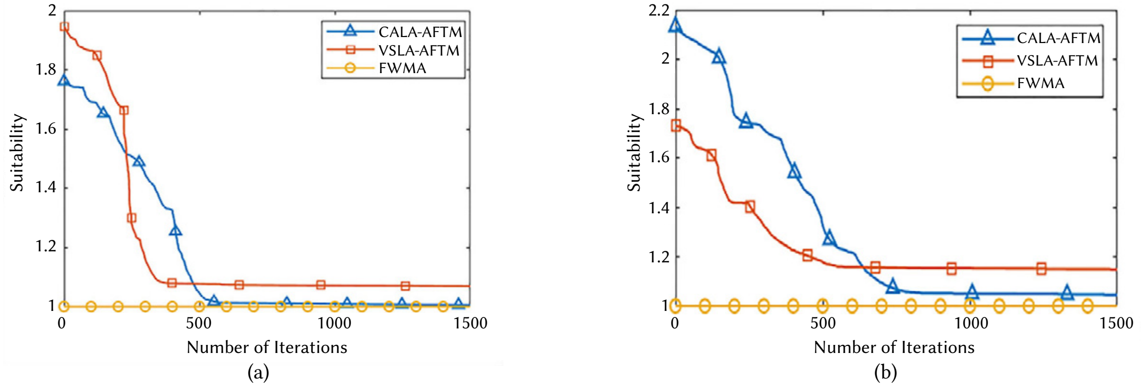
The suitability value for both CTI and NASA datasets is given in Fig. 9a, b. In the CTI dataset, the suitability value for FWMA and CALA-AFTM was almost the same. However, in the NASA dataset, FWMA produced better results.



(a)

(b)

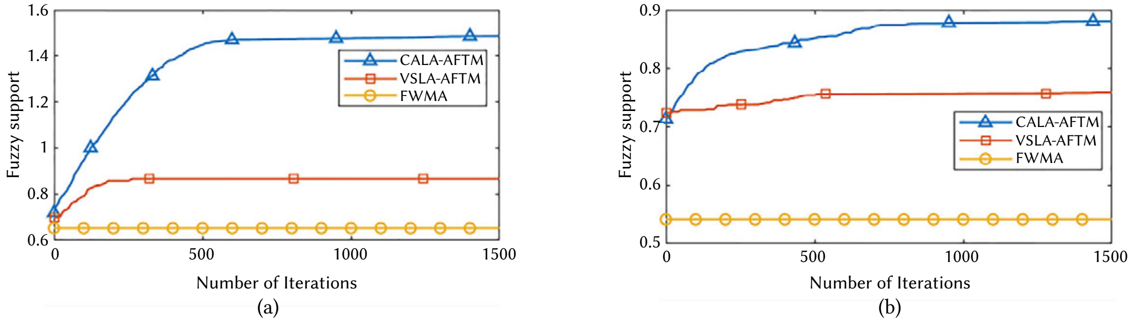
Fig. 8. The value of AVCF with different numbers of iterations on CTI (a) and NASA (b) datasets.



(a)

(b)

Fig. 9. The suitability value with different numbers of iterations on CTI (a) and NASA (b) datasets.



(a)

(b)

Fig. 10. The fuzzy support value with different numbers of iterations on CTI (a) and NASA (b) datasets.

The fuzzy support value for both CTI and NASA datasets is shown in Fig. 10a, b. The proposed CALA-AFTM produced a high fuzzy support. Therefore, the proposed cost function had a good performance. Figures 11 and 12 show the membership functions of the CTI and NASA datasets before and after 12,000 iterations of CALA-AFTM and VSLA-AFTM. Based on Figs. 11 and 12, the initial shape of membership functions are not appropriate. After optimization, the proper TMFs are produced. To assess the effect of dataset sizes on the efficiency of CALA-AFTM algorithm, another experiment was performed. The results of this experiment are shown in Tables VII and VIII. Tables VII and VIII compare the results for the overlap, coverage, fuzzy support, suitability, and the AVCF obtained from three test algorithms. By comparing the results between CALA-AFTM and VSLA-AFTM, we found that CALA-AFTM in the CTI and NASA datasets produces better results for AVCF, fuzzy support, and suitability parameters. VSLA-AFTM has a limited number of actions, so it cannot find the accurate value for the optimal parameters.

Additionally, the statistical significance of the AVCF was analyzed using unpaired t-test. The unpaired t-test results between CALA-AFTM and the other two algorithms for CTI and NASA datasets are given in Tables VII and VIII, respectively. In this experiment, datasets of different sizes were tested. Tables VII and VIII show the values obtained for AVCF. Also, by considering the value of 0.95 for the confidence level, the p-value and t-value are specified. Let X and Y represent two algorithms respectively. In this case, X is statistically better than Y if the t-test (X, Y) is less than zero and the positive p-value is less than 0.05. The results in Tables IX and X show that CALA-AFTM is statistically significant than other algorithms. To assess the effect of number of extracted large 1-sequences and rules, another experiment was performed. In this experiment, the minimum confidence value was assumed to be 0.1 and the results were tested with different minimum support values. The results in Figs. 13 and 14 show that the CALA-AFTM algorithm produces a large number of 1-sequences and more rules than other algorithms. To assess the effect of association rules, another experiment was performed with different values for the minimum support and minimum confidence. The results are shown in Fig. 15a, b respectively. The result in Fig. 15a show that when the minimum support value is greater than 0.002, the number of extracted rules decreases significantly. Also, these result show that when the minimum support is greater than 0.004, the number of extracted rules will be less than 500 rules. Additionally, the result in

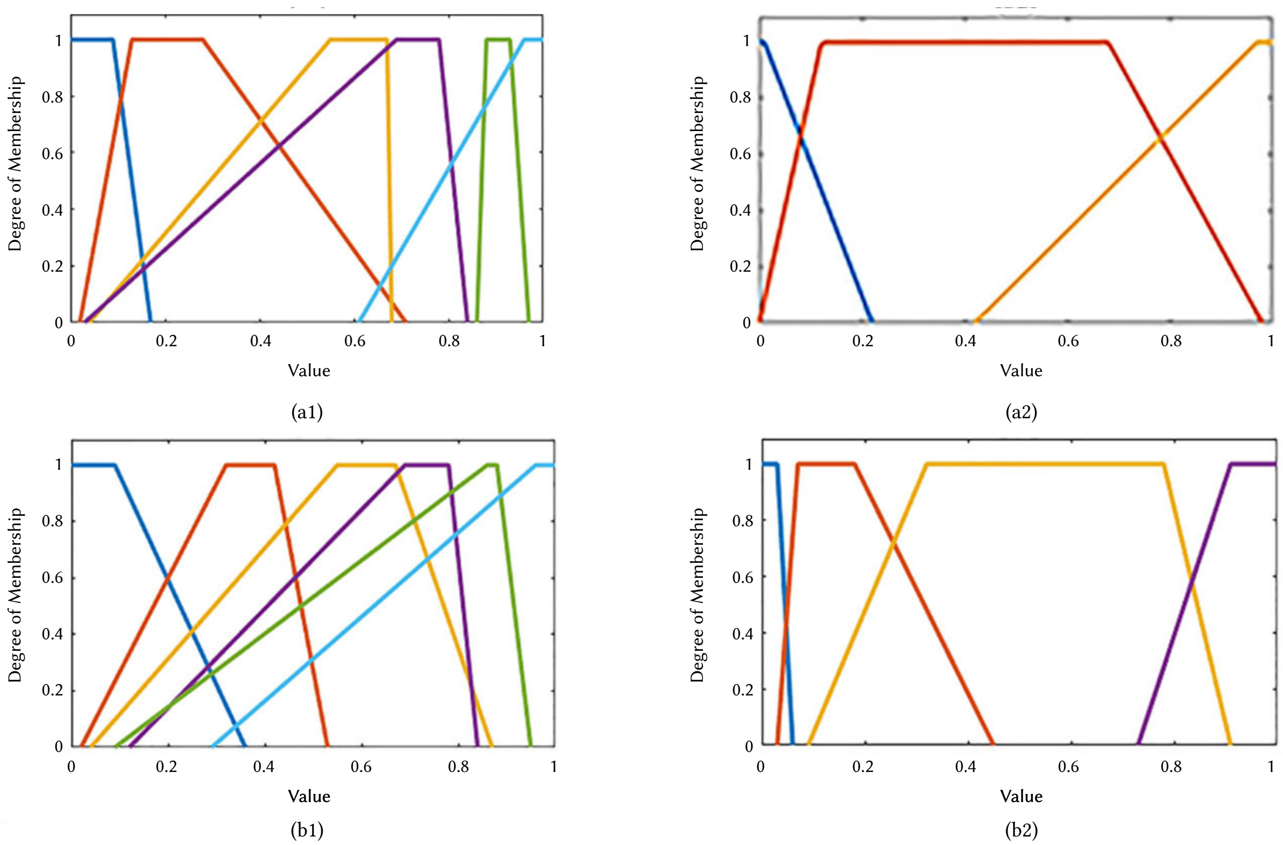


Fig. 11. The initial and optimized TMFs derived by CALA-AFTM and VSLA-AFTM on the CTI dataset. (a_1) Initial TMFs for CALA-AFTM, (b_1) Initial TMFs for VSLA-AFTM, (a_2) Optimized TMFs for CALA-AFTM, (b_2) Optimized TMFs for VSLA-AFTM.

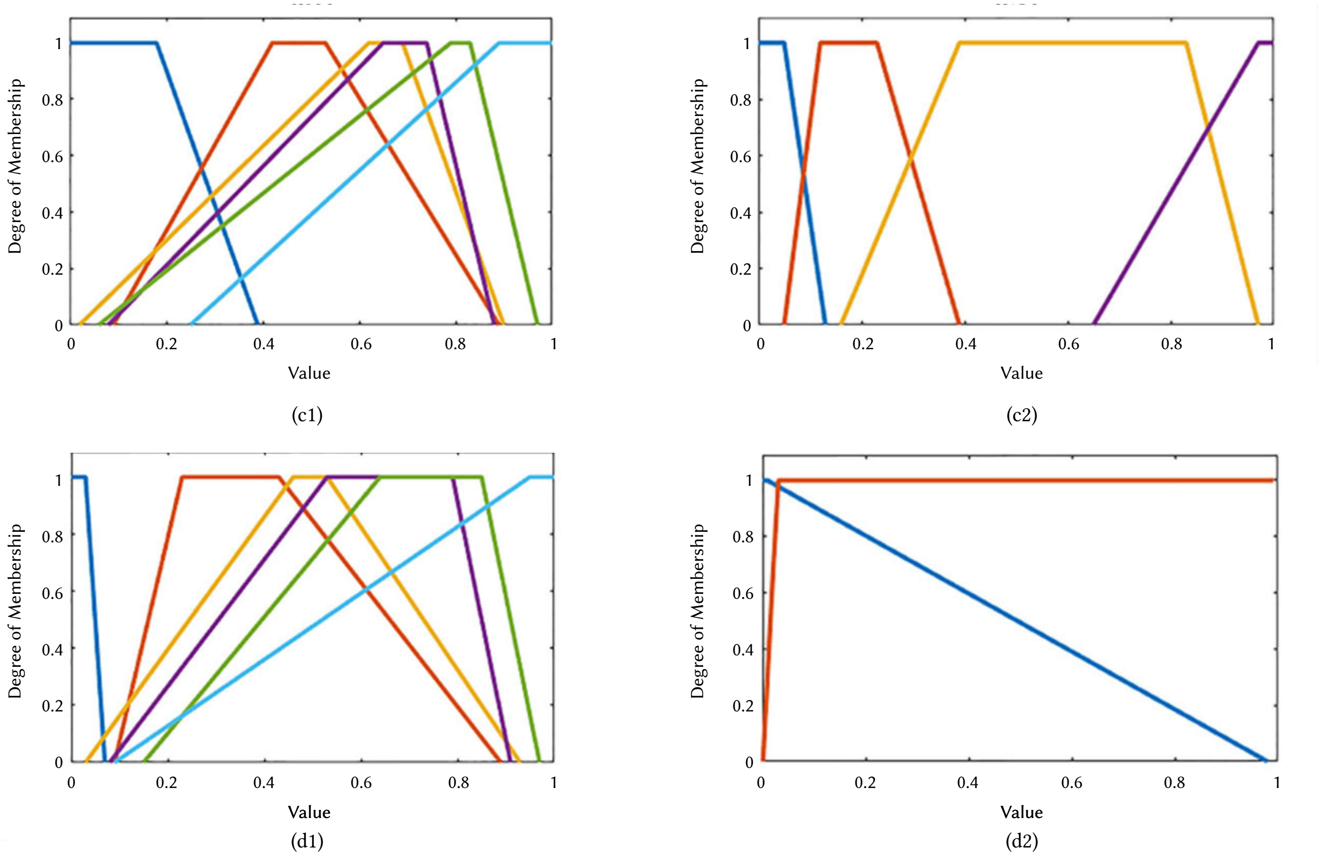


Fig. 12. The initial and optimized TMFs derived by CALA-AFTM and VSLA-AFTM on the NASA dataset. (c_1) Initial TMFs for CALA-AFTM, (d_1) Initial TMFs for VSLA-AFTM, (c_2) Optimized TMFs for CALA-AFTM, (d_2) Optimized TMFs for VSLA-AFTM.

TABLE VII. Comparison of the Results of Different Sizes of Data on the CTI Dataset

Data	Algorithm	Overlap	Coverage	Suitability	Fuzzy support	AVCF
50K	CALA-AFTM	0.0018	1.0000	1.0018	1.4918	0.6715±0.0232
	VSLA-AFTM	0.0245	1.0000	1.0245	0.8365	1.2246±0.0652
	FWMA	0.0000	1.0000	1.0000	0.6509	1.5362±0.0000
100K	CALA-AFTM	0.0025	1.0000	1.0025	1.5075	0.6650±0.0161
	VSLA-AFTM	0.04121	1.0000	1.04121	0.8547	1.2182±0.0580
	FWMA	0.0000	1.0000	1.0000	0.6594	1.5164±0.0000
150K	CALA-AFTM	0.0043	1.0000	1.0043	1.6156	0.6216±0.0672
	VSLA-AFTM	0.0386	1.0000	1.0386	0.8665	1.1986±0.1139
	FWMA	0.0000	1.0000	1.0000	0.6656	1.5023±0.0000
200K	CALA-AFTM	0.0039	1.0000	1.0039	1.5826	0.6343±0.0524
	VSLA-AFTM	0.0583	1.0000	1.0583	0.8784	1.2048±0.0943
	FWMA	0.0000	1.0000	1.0000	0.6632	1.5075±0.0000
250K	CALA-AFTM	0.0050	1.0000±0.0000	1.0050	1.4860	0.6763±0.0389
	VSLA-AFTM	0.0674	1.0000±0.0000	1.0674	0.8663	1.2321±0.0873
	FWMA	0.0000	1.0000±0.0000	1.0000	0.6524	1.5328±0.0000

TABLE VIII. Comparison of the Results of Different Sizes of Data on the NASA Dataset

Data	Algorithm	Overlap	Coverage	Suitability	Fuzzy support	AVCF
130K	CALA-AFTM	0.0165	1.0000	1.0165	0.8618	1.1794±0.0923
	VSLA-AFTM	0.0724	1.0000	1.0724	0.6968	1.5388±0.1235
	FWMA	0.0000	1.0000	1.0000	0.5514	1.8132±0.0000
260K	CALA-AFTM	0.0268	1.0000	1.0268	0.8673	1.1838±0.0856
	VSLA-AFTM	0.0952	1.0000	1.0952	0.7087	1.5452±0.1361
	FWMA	0.0000	1.0000	1.0000	0.5478	1.8252±0.0000
390K	CALA-AFTM	0.0325	1.0000	1.0325	0.8684	1.1889±0.1230
	VSLA-AFTM	0.0863	1.0000	1.0863	0.6974	1.5576±0.1971
	FWMA	0.0000	1.0000	1.0000	0.5460	1.8312±0.0000
520K	CALA-AFTM	0.0409	1.0000	1.0409	0.8813	1.1810±0.0762
	VSLA-AFTM	0.1153	1.0000	1. 1153	0.7236	1.5413±0.1426
	FWMA	0.0000	1.0000	1.0000	0.5496	1.8193±0.0000
650K	CALA-AFTM	0.0452	1.0000	1.0452	0.8766	1.1923±0.1132
	VSLA-AFTM	0.1476	1.0000	1.1476	0.7331	1.5652±0.1745
	FWMA	0.0000	1.0000	1.0000	0.5416	1.8463±0.0000

Fig. 15b show that the extracted rules reach to zero, when the value for the minimum support is greater than 0.05. The result in Fig. 16a, b shows that by increasing the confidence value the number of extracted rules decreases. Also, the results in Fig. 16b shows that when the value of minimum support is greater than 0.6 the curve becomes smoother. Additionally, when the value for the minimum confidence reaches to 0.9, the number of generated rules is zero. Another experiment was performed to evaluate the execution time of the proposed CALA-AFTM algorithm. In this experiment, different action sizes(r) with the values of 10, 20, 30, and 40 were considered for VSLA-AFTM algorithm. We also used values of 3 and 6 for maximum number of TMFs to evaluate the results.

To specify the number of actions in VSLA-AFTM, we employed the notation VSLA (r). The results of both CTI and NASA datasets are shown in Fig 17a, b and Fig18a, b. Figs. 17a, b illustrates average execution time increases when the maximum number of membership functions (kmax) was set at 3 for CTI and NASA dataset, respectively. The execution time in VSLA-AFTM is directly related to the number of actions. VSLA-AFTM with the number of actions 10 and 20 converges faster and has a lower execution time than CALA-AFTM. The execution time of CALA-AFTM is less than VSLA-AFTM with the number of actions 30 and 40. Figs. 18a, b shows average execution time increases when the maximum number of membership functions (kmax) was set at 6 for CTI and NASA dataset, respectively. It can be seen that by increasing kmax and the size of the datasets the average execution time dramatically increases.

VI. Conclusions

In this paper a continuous action-set learning automata-based approach named CALA-AFTM was proposed to automatically determine both the optimal number and position of trapezoidal membership functions in order to extract fuzzy association rules from quantitative transactions. In this method, a new representation

TABLE IX. The Results of T-value and P-value With Different Dataset Sizes on CTI Dataset

Size	AVCF		t-value		p-value		Significant
	Algorithm	value	Pairwise algorithms	value	Pairwise algorithms	value	
50k	CALA-AFTM	0.6715±0.0232	CALA-AFTM, VSLA-AFTM	43.7753	CALA-AFTM, VSLA-AFTM	<0.0001	Extremely statistically significant
	VSLA-AFTM	1.2246±0.0652	CALA-AFTM, FWMA	204.1447	CALA-AFTM, FWMA	<0.0001	Extremely statistically significant
	FWMA	1.5362±0.0000					
100k	Algorithm	value	Pairwise algorithms	value	Pairwise algorithms	value	
	CALA-AFTM	0.6650±0.0161	CALA-AFTM, VSLA-AFTM	50.3380	CALA-AFTM, VSLA-AFTM	<0.0001	Extremely statistically significant
	VSLA-AFTM	1.2182±0.0580	CALA-AFTM, FWMA	289.6466	CALA-AFTM, FWMA	<0.0001	Extremely statistically significant
	FWMA	1.5164±0.0000					
150k	Algorithm	value	Pairwise algorithms	value	Pairwise algorithms	value	
	CALA-AFTM	0.6216±0.0672	CALA-AFTM, VSLA-AFTM	23.8975	CALA-AFTM, VSLA-AFTM	<0.0001	Extremely statistically significant
	VSLA-AFTM	1.1986±0.1139	CALA-AFTM, FWMA	71.7826	CALA-AFTM, FWMA	<0.0001	Extremely statistically significant
	FWMA	1.5023±0.0000					
200k	Algorithm	value	Pairwise algorithms	value	Pairwise algorithms	value	
	CALA-AFTM	0.6343±0.0524	CALA-AFTM, VSLA-AFTM	28.9649	CALA-AFTM, VSLA-AFTM	<0.0001	Extremely statistically significant
	VSLA-AFTM	1.2048±0.0943	CALA-AFTM, FWMA	91.2732	CALA-AFTM, FWMA	<0.0001	Extremely statistically significant
	FWMA	1.5075±0.0000					
250k	Algorithm	value	Pairwise algorithms	value	Pairwise algorithms	value	
	CALA-AFTM	0.6763±0.0389	CALA-AFTM, VSLA-AFTM	31.8520	CALA-AFTM, VSLA-AFTM	<0.0001	Extremely statistically significant
	VSLA-AFTM	1.2321±0.0873	CALA-AFTM, FWMA	120.5975	CALA-AFTM, FWMA	<0.0001	Extremely statistically significant
	FWMA	1.5328±0.0000					

TABLE X. The Results for the P-value and T-value Parameters on NASA Dataset

Size	AVCF		t-value		p-value		significant
	Algorithm	value	Pairwise algorithms	value	Pairwise algorithms	value	
130k	CALA-AFTM	1.1794±0.0464	CALA-AFTM, VSLA-AFTM	19.0551	CALA-AFTM, VSLA-AFTM	<0.0001	Extremely statistically significant
	VSLA-AFTM	1.5388±0.0923	CALA-AFTM, FWMA	74.8161	CALA-AFTM, FWMA	<0.0001	Extremely statistically significant
	FWMA	1.8132±0.0000					
260k	Algorithm	value	Pairwise algorithms	value	Pairwise algorithms	value	
	CALA-AFTM	1.1838±0.0856	CALA-AFTM, VSLA-AFTM	12.9879	CALA-AFTM, VSLA-AFTM	<0.0001	Extremely statistically significant
	VSLA-AFTM	1.5452±0.1261	CALA-AFTM, FWMA	41.0408	CALA-AFTM, FWMA	<0.0001	Extremely statistically significant
	FWMA	1.8252±0.0000					
390k	Algorithm	value	Pairwise algorithms	value	Pairwise algorithms	value	
	CALA-AFTM	1.1889±0.1230	CALA-AFTM, VSLA-AFTM	8.6922	CALA-AFTM, VSLA-AFTM	<0.0001	Extremely statistically significant
	VSLA-AFTM	1.5576±0.1971	CALA-AFTM, FWMA	28.6018	CALA-AFTM, FWMA	<0.0001	Extremely statistically significant
	FWMA	1.8312±0.0000					
520K	Algorithm	value	Pairwise algorithms	value	Pairwise algorithms	value	
	CALA-AFTM	1.1810±0.0762	CALA-AFTM, VSLA-AFTM	12.2057	CALA-AFTM, VSLA-AFTM	<0.0001	Extremely statistically significant
	VSLA-AFTM	1.5413±0.1426	CALA-AFTM, FWMA	45.8807	CALA-AFTM, FWMA	<0.0001	Extremely statistically significant
	FWMA	1.8193±0.0000					
650k	Algorithm	value	Pairwise algorithms	value	Pairwise algorithms	value	
	CALA-AFTM	1.1923±0.1132	CALA-AFTM, VSLA-AFTM	9.8195	CALA-AFTM, VSLA-AFTM	<0.0001	Extremely statistically significant
	VSLA-AFTM	1.5652±0.1745	CALA-AFTM, FWMA	31.6440	CALA-AFTM, FWMA	<0.0001	Extremely statistically significant
	FWMA	1.8463±0.0000					

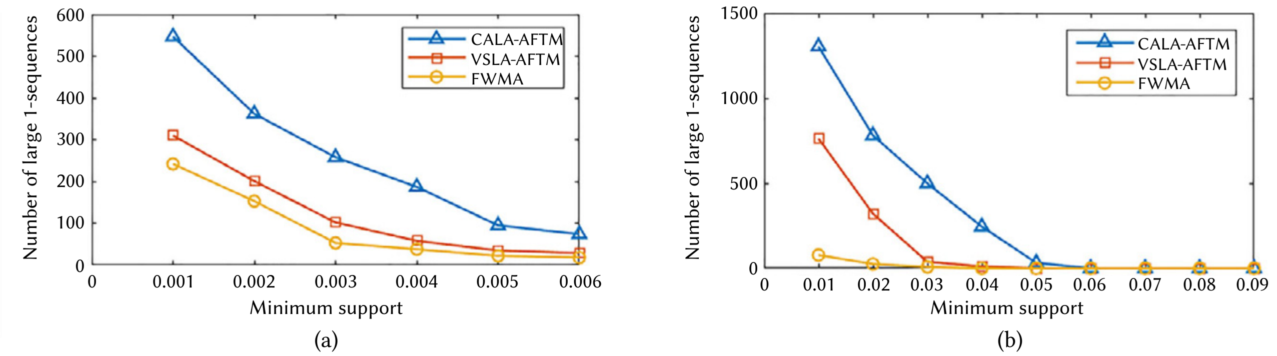


Fig. 13. The effect of number of extracted large 1-sequences on CTI dataset (a), NASA dataset (b).

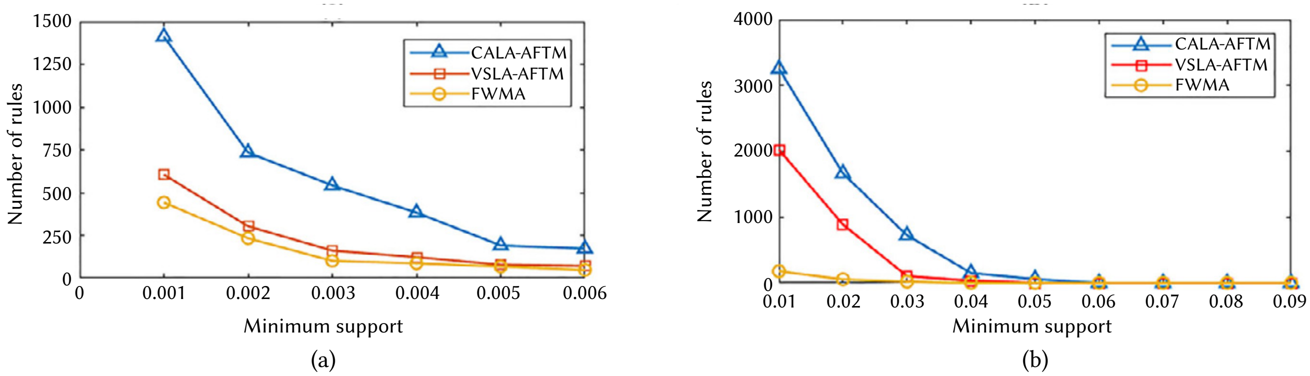


Fig. 14. The effect of number of extracted rules on CTI dataset (a), NASA dataset (b).

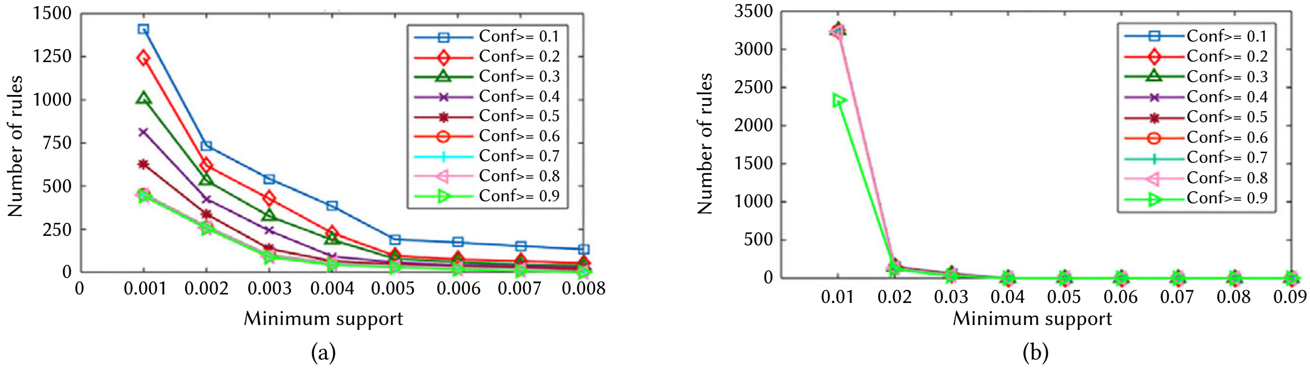


Fig. 15. The effect of numbers of extracted rules and the minimum support with different values for minimum confidence on CTI dataset (a), NASA dataset (b).

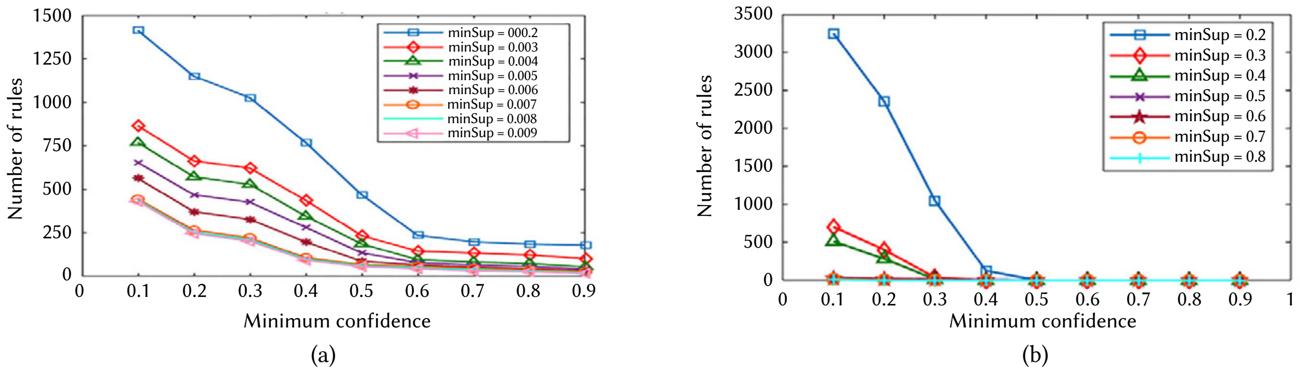


Fig. 16. The effect of numbers of extracted rules and the minimum confidence with different values for minimum support on CTI dataset (a), NASA dataset (b).

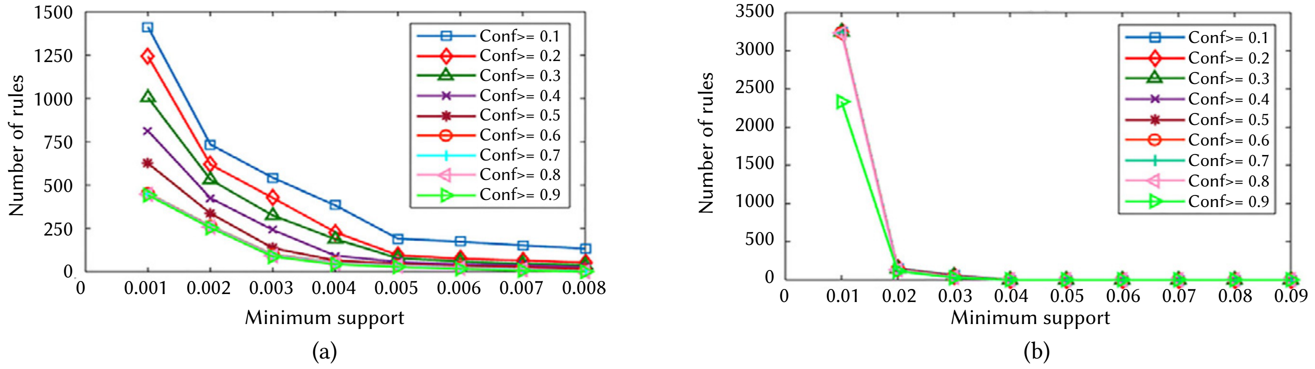


Fig. 17. The average execution time along with different dataset sizes with kmax=3 on CTI dataset (a), NASA dataset (b).

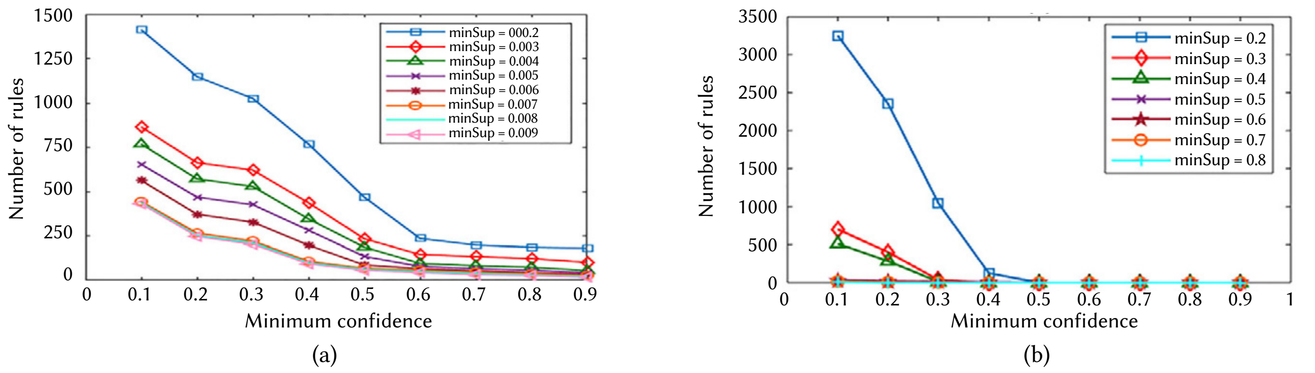


Fig. 18. The average execution time along with different dataset sizes with kmax=6 on CTI dataset (a), NASA dataset (b).

is introduced to construct a team of CALA for cooperating learning automata. The learning automaton of this team of CALA is composed of two parts. The learning automata of the first part try to determine the proper number of membership functions, and the relative learning automata in the second part optimize their positions. The proposed algorithm has been tested on web usage data, to find membership functions and significant rules. Therefore, the extracted rules represent the browsing behavior of users and can be used to give some proper suggestions to web-server administrators. In the proposed CALA-AFTM, to reduce the domain of the search space and remove unsuitable membership functions, two conditions were applied and a novel algorithm was proposed. Also, to check the effectiveness of the proposed CALA-AFTM, various experiments were performed using two real data sets. We showed that with increasing the different dataset sizes, compared with other algorithms, CALA-AFTM provides better results in terms of fuzzy support, overlap, number of large 1-sequences, coverage, number of rules, and cost function. In future work, our goal will be to develop the proposed method in topics such as multi-objective applications, fuzzy temporal rule mining, fuzzy generalized association rule mining, and 2-tuple fuzzy linguistic representation.

References

[1] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery in databases," *AI Magazine*, vol. 17, no. 3, pp. 37-37, 1996.

[2] M.-S. Chen, J. Han, and P. S. Yu, "Data mining: an overview from a database perspective," *IEEE Transactions on Knowledge and data Engineering*, vol. 8, no. 6, pp. 866-883, 1996.

[3] K. Wagstaff, C. Cardie, S. Rogers, and S. Schrödl, "Constrained k-means clustering with background knowledge," *in Proceedings of the 8th International Conference on Machine Learning, 2001*, vol. 1, pp. 577-584.

[4] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM transactions on intelligent systems and technology (TIST)*, vol. 2, no. 3, p. 27, 2011.

[5] T. Hastie, R. Tibshirani, J. Friedman, and J. Franklin, "The elements of statistical learning: data mining, inference and prediction," *The Mathematical Intelligencer*, vol. 27, no. 2, pp. 83-85, 2005.

[6] C. C. Aggarwal and C. Zhai, Mining text data, New York, NY: Springer, 2012.

[7] T.-P. Hong, C.-H. Chen, Y.-C. Lee, and Y.-L. Wu, "Genetic-fuzzy data mining with divide-and-conquer strategy," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 2, pp. 252-265, 2008.

[8] G. Sumathi and J. Akilandeswari, "Improved fuzzy weighted-iterative association rule-based ontology postprocessing in data mining for query recommendation applications," Computational Intelligence, vol. 36, no. 2, pp. 773-782, 2020.

[9] C. H. Cheng and C. H. Chen, "Fuzzy time series model based on weighted association rule for financial market forecasting," *Expert Systems*, vol. 35, no. 4, p. e12271, 2018.

[10] F. Chiclana, R. Kumar, M. Mittal, M. Khari, J. M. Chatterjee, and S. W. Baik, "ARM–AMO: an efficient association rule mining algorithm based on animal migration optimization," *Knowledge-Based Systems*, vol. 154, pp. 68-80, 2018.

[11] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo, "Fast discovery of association rules," *Advances in knowledge discovery and data mining*, vol. 12, no. 1, pp. 307-328, 1996.

[12] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," *in Proceedings of the 1993 ACM SIGMOD international conference on Management of data (SIGMOD'98), New York: ACM Press*, 1993, pp. 207-216.

[13] R. Agrawal, "Fast Algorithm for Mining Association Rules in Large Databases," in Proceedings of International Conference on Very Large Databases (VLDB '94), Santiago, Chile,1994, pp.487-499.

[14] J. Chen, A. Mikulcic, and D. H. Kraft, *An integrated approach to information retrieval with fuzzy clustering and fuzzy inferencing*, in Knowledge Management in Fuzzy Databases (pp. 247-260), Heidelberg, Germany: Physica-Verlag, 2000.

[15] H. Ishibuchi, T. Nakashima, and M. Nii, *Classification and modeling with linguistic information granules: Advanced approaches to linguistic Data Mining*, Springer Science & Business Media, Berlin, Germany: Springer-Verlag, 2004.

[16] W. Siler and J. J. Buckley, *Fuzzy expert systems and fuzzy reasoning*. John

Wiley & Sons, Hoboken, NJ, USA, p. 424.

[17] H. Zhang and D. Liu, *Fuzzy modeling and fuzzy control*, Boston, MA, USA: Birkhauser, 2006.

[18] A. Kandel, *Fuzzy expert systems*, Boca Raton, Florida, USA: CRC Press, 1991.

[19] L. A. Zadeh, "Fuzzy sets," *Information and control*, vol. 8, no. 3, pp. 338-353, 1965.

[20] K. C. Chan and W.-H. Au, "Mining fuzzy association rules," *in Proceedings of the sixth international conference on Information and knowledge management*, Las Vegas, NV, 1997, pp. 209-215.

[21] J. S. Yue, E. Tsang, D. Yeung, and D. Shi, "Mining fuzzy association rules with weighted items," *in proceedings of the ieee international conference on systems, man and cybernetics*, Nashville, TN, 2000, pp. 1906-1911.

[22] C. M. Kuok, A. Fu, and M. H. Wong, "Mining fuzzy association rules in databases," *ACM Sigmod Record*, vol. 27, no. 1, pp. 41-46, 1998.

[23] N. Marín, M. D. Ruiz, and D. Sánchez, "Fuzzy frameworks for mining data associations: fuzzy association rules and beyond," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 6, no. 2, pp. 50-69, 2016.

[24] R. Pierrard, J.-P. Poli, and C. Hudelot, "A Fuzzy Close Algorithm for Mining Fuzzy Association Rules," *in International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, Springer, Cham, 2018, pp. 88-99.

[25] C. Molina, M. D. Ruiz, and J. M. Serrano, "Representation by levels: An alternative to fuzzy sets for fuzzy data mining," *Fuzzy Sets and Systems*, vol. 401, pp. 113-132, 2019.

[26] E. Cuevas, J. Gálvez, and O. Avalos, *Fuzzy Logic Based Optimization Algorithm*, in Recent Metaheuristics Algorithms for Parameter Identification, pp. 135-181, Springer, Cham, 2020.

[27] S. G. Matthews, M. A. Gongora, A. A. Hopgood, and S. Ahmadi, "Web usage mining with evolutionary extraction of temporal fuzzy association rules," *Knowledge-Based Systems*, vol. 54, pp. 66-72, 2013.

[28] R. Wu, "Mining generalized fuzzy association rules from Web logs," *in 2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery*, 2010, vol. 5, pp. 2474-2477: IEEE.

[29] T.-P. Hong, C.-M. Huang, and S.-J. Horng, "Linguistic object-oriented web-usage mining," *International Journal of Approximate Reasoning*, vol. 48, no. 1, pp. 47-61, 2008.

[30] C. Chai and B. Li, "A novel association rules method based on genetic algorithm and fuzzy set strategy for web mining," *Journal of Computers*, vol. 5, no. 9, pp. 1448-1455, 2010.

[31] Y.-H. Tao, T.-P. Hong, W.-Y. Lin, and W.-Y. Chiu, "A practical extension of web usage mining with intentional browsing data toward usage," *Expert Systems with Applications*, vol. 36, no. 2, pp. 3937-3945, 2009.

[32] C.-H. Chen, G.-C. Lan, T.-P. Hong, and S.-B. Lin, "Mining fuzzy temporal association rules by item lifespans," *Applied Soft Computing*, vol. 41, pp. 265-274, 2016.

[33] C.-K. Ting, T.-C. Wang, R.-T. Liaw, and T.-P. Hong, "Genetic algorithm with a structure-based representation for genetic-fuzzy data mining," *Soft Computing*, vol. 21, no. 11, pp. 2871-2882, 2017.

[34] T.-P. Hong, Y.-C. Lee, and M.-T. Wu, "An effective parallel approach for genetic-fuzzy data mining," *Expert Systems with Applications*, vol. 41, no. 2, pp. 655-662, 2014.

[35] J. C.-W. Lin, T. Li, P. Fournier-Viger, T.-P. Hong, J. M.-T. Wu, and J. Zhan, "Efficient mining of multiple fuzzy frequent itemsets," *International Journal of Fuzzy Systems*, vol. 19, no. 4, pp. 1032-1040, 2017.

[36] G. Santharam, P. Sastry, and M. Thathachar, "Continuous action set learning automata for stochastic optimization," *Journal of the Franklin Institute*, vol. 331, no. 5, pp. 607-628, 1994.

[37] H. Beigy and M. Meybodi, "A new continuous action-set learning automaton for function optimization," *Journal of the Franklin Institute*, vol. 343, no. 1, pp. 27-47, 2006.

[38] K. S. Narendra and M. A. Thathachar, Learning automata: an introduction, Prentice-Hall, Englewood Cliffs, NJ: Prentice Hall, 1989.

[39] M. A. Thathachar and P. S. Sastry, Networks of learning automata: Techniques for online stochastic optimization, Dordrecht, The Netherlands: Kluwer, 2004.

[40] S. Sanyal, M. Bansal, S. Banerjee, and P. Kalra, "On learning shapes from shades," *in Proceedings of 4th Indian Conference on Computer Vision Graphics and Image Processing*, Calcutta, India, 2004, pp. 275–282.

[41] H. Beigy and M. R. Meybodi, "An adaptive call admission algorithm for cellular networks," *Computers & Electrical Engineering*, vol. 31, no. 2, pp. 132-151, 2005.

[42] B. Anari, J. A. Torkestani, and A. M. Rahmani, "Automatic data clustering using continuous action-set learning automata and its application in segmentation of images," *Applied Soft Computing*, vol. 51, pp. 253-265, 2017.

[43] T.-P. Hong, M.-J. Chiang, and S.-L. Wang, "Mining weighted browsing patterns with linguistic minimum supports," *in IEEE International Conference on Systems, Man and Cybernetics*, 2002, Yasmine Hammamet, Tunisia, pp. 635-639.

[44] T.-P. Hong, M.-J. Chiang, and S.-L. Wang, "Mining fuzzy weighted browsing patterns from time duration and with linguistic thresholds," *American Journal of Applied Sciences*, vol. 5, no. 12, pp. 1611, 2008.

[45] R. Wu, W. Tang, and R. Zhao, "Web mining of preferred traversal patterns in fuzzy environments," *Springer-Verlag, Berlin Heidelberg Lecture Notes in Artificial Intelligence*, vol. 3642, 2005, pp. 456-465.

[46] S.-L. Wang, W.-S. Lo, and T.-P. Hong, "Discovery of fuzzy multiple-level Web browsing patterns," in Proceedings of the international conference on fuzzy systems and knowledge discovery: Springer, Singapore, 2005, pp. 251-266.

[47] W.-S. Lo, T.-P. Hong, and S.-L. Wang, "A top-down fuzzy cross-level Web-mining approach," *in: The 2003 IEEE International Conference on Systems, Man and Cybernetics*, 2003, vol. 3, pp. 2684-2689.

[48] T.-P. Hong, K.-Y. Lin, and B.-C. Chien, "Mining fuzzy multiple-level association rules from quantitative data," *Applied Intelligence*, vol. 18, no. 1, pp. 79-90, 2003.

[49] T.-P. Hong, C.-H. Chen, Y.-L. Wu, and Y.-C. Lee, "A GA-based fuzzy mining approach to achieve a trade-off between number of rules and suitability of membership functions," *Soft Computing*, vol. 10, no. 11, pp. 1091-1101, 2006.

[50] C.-H. Chen, V. S. Tseng, and T.-P. Hong, "Cluster-based evaluation in fuzzy-genetic data mining," *IEEE transactions on fuzzy systems*, vol. 16, no. 1, pp. 249-262, 2008.

[51] C.-H. Chen, Y. Li, and T.-P. Hong, "Type-2 genetic-fuzzy mining with tuning mechanism," *in 2015 Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, 2015, pp. 296-299: IEEE.

[52] J. Alcalá-Fdez, R. Alcalá, M. J. Gacto, and F. Herrera, "Learning the membership function contexts for mining fuzzy association rules by using genetic algorithms," *Fuzzy Sets and Systems*, vol. 160, no. 7, pp. 905-921, 2009.

[53] W. Wang and S. Bridges, "Genetic algorithm optimization of membership functions for mining fuzzy association rules," *in: International Joint Conference on information Systems, Fuzzy Theory and Technology Conference*, Atlantic City, NY, 2000, pp. 1–4.

[54] C.-H. Chen, Y. Li, T.-P. Hong, Y.-K. Li, and E. H.-C. Lu, "A GA-based approach for mining membership functions and concept-drift patterns," *in 2015 IEEE Congress on Evolutionary Computation (CEC)*, 2015, pp. 2961-2965: IEEE.

[55] C.-H. Chen, T.-P. Hong, Y.-C. Lee, and V. S. Tseng, "Finding active membership functions for genetic-fuzzy data mining," *International Journal of Information Technology & Decision Making*, vol. 14, no. 06, pp. 1215-1242, 2015.

[56] A. M. Palacios, J. L. Palacios, L. Sánchez, and J. Alcalá-Fdez, "Genetic learning of the membership functions for mining fuzzy association rules from low quality data," *Information Sciences*, vol. 295, pp. 358-378, 2015.

[57] M. Kaya and R. Alhajj, "Genetic algorithms-based optimization of membership functions for fuzzy weighted association rules mining," *in Proceedings of International Conference Symposium on Computers and Communications ISCC*, 2004, vol. 1, pp. 110-115.

[58] T.-P. Hong, Y.-F. Tung, S.-L. Wang, M.-T. Wu, and Y.-L. Wu, "An ACS-based framework for fuzzy data mining," *Expert Systems with Applications*, vol. 36, no. 9, pp. 11844-11852, 2009.

[59] M.-T. Wu, T.-P. Hong, and C.-N. Lee, "A continuous ant colony system framework for fuzzy data mining," *Soft Computing*, vol. 16, no. 12, pp. 2071-2082, 2012.

[60] C.-K. Ting, R.-T. Liaw, T.-C. Wang, and T.-P. Hong, "Mining fuzzy association rules using a memetic algorithm based on structure representation," *Memetic Computing*, vol. 10, no. 1, pp. 15-28, 2018.

[61] J. Botzheim, C. Cabrita, L. T. Kóczy, and A. Ruano, "Fuzzy rule extraction

by bacterial memetic algorithms," *International Journal of Intelligent Systems*, vol. 24, no. 3, pp. 312-339, 2009.

[62] A. Song, J. Song, X. Ding, G. Xu, and J. Chen, "Utilizing bat algorithm to optimize membership functions for fuzzy association rules mining," *in International Conference on Database and Expert Systems Applications*, 2017, pp. 496-504.

[63] M. A. Chamazi and H. Motameni, "Finding suitable membership functions for fuzzy temporal mining problems using fuzzy temporal bees' method," *Soft Computing*, vol. 23, no. 10, pp. 3501-3518, 2019.

[64] F. Alikhademi and S. Zainudin, "Generating of derivative membership functions for fuzzy association rule mining by Particle Swarm Optimization," in 2014 International Conference on Computational Science and Technology (ICCST), 2014, pp. 1-6.

[65] F. Rudziński, "A multi-objective genetic optimization of interpretability-oriented fuzzy rule-based classifiers," *Applied Soft Computing*, vol. 38, pp. 118-133, 2016.

[66] B. Minaei-Bidgoli, R. Barmaki, and M. Nasiri, "Mining numerical association rules via multi-objective genetic algorithms," *Information Sciences*, vol. 233, pp. 15-24, 2013.

[67] H. R. Qodmanan, M. Nasiri, and B. Minaei-Bidgoli, "Multi objective association rule mining with genetic algorithm without specifying minimum support and minimum confidence," *Expert Systems with applications*, vol. 38, no. 1, pp. 288-298, 2011.

[68] U. M. Patil and J. Patil, "Mining fuzzy association rules from web usage quantitative data," *Computer Science & Information Technology*, vol. 89, 2016.

[69] M. Ghavipour and M. R. Meybodi, "An adaptive fuzzy recommender system based on learning automata," *Electronic Commerce Research and Applications*, vol. 20, pp. 105-115, 2016.

[70] M. A. Thathachar and P. S. Sastry, "Varieties of learning automata: an overview," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 32, no. 6, pp. 711-722, 2002.

[71] K. S. Narendra and M. A. Thathachar, "On the behavior of a learning automaton in a changing environment with application to telephone traffic routing," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 10, no. 5, pp. 262-269, 1980.

[72] N. Kumar, J.-H. Lee, and J. J. Rodrigues, "Intelligent mobile video surveillance system as a Bayesian coalition game in vehicular sensor networks: Learning automata approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 3, pp. 1148-1161, 2014.

[73] A. Helmzadeh and S. M. Kouhsari, "Calibration of erroneous branch parameters utilising learning automata theory," IET Generation, Transmission & Distribution, vol. 10, no. 13, pp. 3142-3151, 2016.

[74] B. Anari, J. Akbari Torkestani, and A. M. Rahmani, "A learning automata-based clustering algorithm using ant swarm intelligence," *Expert systems*, vol. 35, no. 6, p. e12310, 2018.

[75] http://www.cs.depaul.edu.

[76] https://ita.ee.lbl.gov/html/traces.html.

Zohreh Anari

Zohreh Anari is currently Associate Professor in Computer Science at Department of Computer Engineering and Information Technology, Payame Noor University, Iran. She also received the Ph.D. degree in Computer Engineering from Islamic Azad University, Urmia, Iran, in 2020. She joined the faculty member of Computer Engineering Department at Payame Noor University, Shabestar, Iran in 2008. Her research interests are learning systems, data mining, web mining, fuzzy systems and soft computing. Recently, she is working in the field of fog computing and Blockchain.

Abdolreza Hatamlou

Abdolreza Hatamlou is Associate Professor of the Department of Computer Science at Islamic Azad University, Khoy Branch where he has been a faculty member since 2002. He completed his Ph.D. at National University of Malaysia in 2012. His research interests lie in the area of optimization, meta-heuristic and nature-inspired algorithms, data mining and clustering. He has collaborated actively with researchers in several other disciplines of computer science. Abdolreza Hatamlou has served on roughly twenty conference and workshop program committees.

Babak Anari

Babak Anari is currently is Assistance Professor in Department of Computer Science at Islamic Azad University, Shabestar Branch. He completed his Ph.D. at Science and Research Branch, Islamic Azad University (IAU), Tehran, in 2017. His research interests are in the areas of Learning Automata, Cellular Automata, fog computing, Blockchain, and IoT.

Research on the Application of Computer Graphic Advertisement Design Based on a Genetic Algorithm and TRIZ Theory

Yang Song*

School of Academy of fine arts, Baotou Teachers College; Baotou Inner Mongolia, 014030 (China)

Received 2 November 2020 | Accepted 17 April 2021 | Published 5 August 2021

Abstract

In view of the shortcomings of the traditional thinking of computer graphic advertising design, this paper introduces TRIZ innovative thinking to design computer advertising. First of all, combined with specific cases of computer creative print advertising, this paper analyzes the creative methods of stimulating divergent thinking, aggregation thinking and transformation thinking from the innovation principle of TRIZ theory as the origin, and applies them to the creative mechanism and application program of print advertising creativity. The whole process is led by rational principles of perceptual thinking, driven by specific principles of abstract imagination, to explore the thinking source of creative design essence of print advertising. The theory and its application mechanism become a new thinking method and application attempt in the creative field of print advertisement. Then, based on the TRIZ innovation theory, the business model of advertising content arrangement is constructed, and the mathematical model is constructed according to the planning business media resource planning on the business model to realize the multi-objective optimization of efficient use of orders and precise delivery of time. Finally, a multi-objective optimization mathematical model of parallel genetic algorithm is designed to solve the advertisement content arrangement. The innovative thinking of TRIZ and the application of genetic algorithm in content arrangement of computer graphic advertisement design are verified by experiments.

Keywords

Content Arrangement, Genetic Algorithm, Innovative Thinking, Print Advertising, TRIZ Theory.

DOI: 10.9781/ijimai.2021.08.007

I. The Application of TRIZ Theory in Computer Graphic Advertisement Design

A. An Overview of TRIZ Theory

1. The Concept of TRIZ Theory

TRIZ (theory of the solution of innovative problems from Russian) is translated into "theory of solving invention problems." TRIZ provides comprehensive theory and method tools for people to solve problems creatively. The three basic theoretical bases are as follows [1]:

1. In the practice of solving problems, various contradictions and corresponding solutions are always repeated;
2. There are not many innovative principles and methods to solve problems thoroughly rather than compromise, which can be learned and mastered by ordinary scientific and technological personnel;
3. The most effective principles and methods to solve this field's problems often come from scientific knowledge in other areas [2]. After more than half a century of development, TRIZ theory has been widely used in many fields worldwide, it has not been applied to the creative design of print advertising.

* Corresponding author.

E-mail address: songyang1008@yeah.net

2. The Relevance Between TRIZ's Core Idea and Creative Thinking of Print Advertisement

In recent years, the research on TRIZ theory in the field of innovation is on the rise. At present, TRIZ has penetrated from the initial engineering technology field to social science, management science, and other fields. Interdisciplinary knowledge acquisition is significant [3]. When people encounter difficult problems, they can seek knowledge of their discipline and expand outside the discipline to find solutions in other professional fields. The scientific principles of innovative design often belong to other fields. This core idea is in line with this study's theme, applying TRIZ theory in the creative thinking of print advertising. Computer graphics is the computer science branch that uses a computer to create images. Today's dominant technology is computer graphics in digital imaging, animation, video games, cellular and computer displays, and various specialist applications. Computer graphics can be a method for growing student knowledge of science topics like mathematics. Through real-time, interactive, and visual input, students can easily see the impact of mathematics on creating successful designs. Graphic designers prepare, evaluate, and create visual solutions for communication issues using color, form, illustration, photography, animation, various printing and layout technology to distinguish print and electronic media messages. Graphic designers produce and manufacture multiple magazines, newspapers, journals, business reports, and other publications as a whole and work in various ways in advertising. Graphic designers have some responsibility in advertising, including the manufacture of promotional panels, product and service packaging and marketing brochures, logos design for goods and organizations, and the creation of signs and signage systems for businesses and governments. Graphic designers often create computer and internet materials, including webpage, digital, and virtual media projects.

B. Combination of TRIZ Innovation Principle and Traditional Creative Thinking Method

1. The Innovative Principle of TRIZ Theory

The principle of TRIZ innovation is to use scientific discovery principles and methods to analyze and solve problems rationally and thoroughly. It contains the common principles followed by human innovation. It is the earliest, most basic, most core, and the highest practical utilization rate of TRIZ theory. It is useful and easy to learn and master, as shown in Table I.

TABLE I. Key Innovation Principles of TRIZ Theory

Serial Number	Innovation Principle	Specific Description
1	Segmentation Principle	(1) Divide the whole into separate parts (2) Divide the total into parts that are easy to assemble and easy to disassemble (3) Improve the whole, separable ability, realize the entire transformation
…	…	…
5	Combination Principle	(1) Spatial dimension, the combination of the same or similar objects (2) Time dimension, which merges the same or related operations
6	Versatility Principle	(1) An object has many different functions (2) Trim objects that do not have necessary functions
13	Reverse Action Principle	(1) To turn an object upside down or inside out (2) To change the motion of an object or environment to rest
…	…	…
14	Composite Materials Principle	Replace the homogeneous component with a composite component

2. Traditional Creative Thinking Method

1. Divergent thinking: divergent thinking is a thinking form in which the brain presents a multidimensional diffusion state. Starting from a thinking starting point, it puts forward rich ideas and seeks multiple ways to solve specific problems [4]. Authors should not stick to traditional practices and create more possibilities.
2. Aggregation thinking: aggregation thinking refers to the sublimation of logical conclusions from the existing representation and broad ideas' aggregation into a focus. It is a convergent thinking mode with scope, direction, and order [5].
3. Conversion thinking: conversion thinking is to observe objects from different aspects and angles with the perspective of connection and development, change new attitudes, avoid thinking stereotypes, and then get a comprehensive understanding of objects and complete solutions.

3. Suitability Analysis of the Combination of TRIZ Theory and Traditional Creative Thinking Method

1. The improvement of logicality and efficiency is the focus of existing innovative thinking methods in graphic design, whose procedures, steps, and measures are mostly based on overcoming creative psychological barriers to stimulate creative thinking [6]. Its methods are highly abstract and generalized and tend to be formalized [7]. TRIZ reveals the internal law and principle of creation. Compared with the traditional creative thinking method, it uses scientific methods to attribute special problems to TRIZ's general problems and solve the problems. TRIZ is more rapid, accurate, and efficient in the technique and process of solving problems than the traditional creative thinking method.
2. To stimulate creative thinking from the original point: TRIZ is a controllable and effective method and powerful tool to generate innovative thinking. Authors should break the inertia and one-sided restriction of thought and avoid the blindness and limitation of the traditional creative process. TRIZ theory affirms that the basic principles of creativity exist objectively. Software graphic design provides visual ideas and an innovative advertisement and marketing template. It is a pattern of the way your advertising influences your marketing. The graphics designer plays an important role in publishing information and making more enticing pictures of a product to raise the demand. It uses technology to control and combine words, images, color, and typography to create emotional feelings and transmit messages. It involves different tools, which makes the design time-efficient and straightforward. The method of images gives you many items that you can easily find where you have them to be put. Once again, when talking about coloring, then a single click fills up the region, and editing is too quick.

These principles can be sorted out and summarized into targeted design creative theories, which can shorten the creative design cycle and improve the success rate. Fig. 1 shows the case of computer graphic advertisement design under the traditional innovative thinking method.

Fig. 1. Computer graphic advertisement design case under the traditional creative thinking method.

To sum up, the existing creative thinking methods have apparent deficiencies, and TRIZ can make up for them to generate more suitable innovative methods for design. However, TRIZ can solve problems creatively and achieve design innovation through scientific and rational ways. To better grasp the breakthrough point of TRIZ theory in creative thinking, it is necessary to define the innovation principle of TRIZ and traditional creative thinking. Among them, the TRIZ innovation principle focuses on the microanalysis of the expression law of creativity. In contrast, traditional creative thinking focuses on analyzing the general law of invention in the operation process from a macro perspective [8]. Computer graphics is the computer science industry that uses computers to generate images. Here Computer graphics is a central technology in digital imaging, video games, and many advanced applications. Computer graphics are the primary technology. Much specialized hardware and software have been developed, with computer graphics hardware-driven displays on most devices. The field of computer science is vast and recently established. Graphic designers are involved in professional design services such as ads, printing, related support, journals, journal books, publishers' directories, and of computer systems graphics for design companies. The emphasis is on graphical designers with website design expertise and animation experience due to increased interest in interactive media projects. As advertising companies produce prints and web marking and promotional materials for more products and services, especially internet advertising, graphic design demand in advertising will rise. A broad liberal education in the art and experience in marketing and management, such as the liberal advertisement and graphic design program, makes candidates more suitable for positions that work on communications strategies.

In graphic advertising design and the known traditional creative thinking methods, Authors need to explore further the thinking origin that can induce it. This paper attempts to introduce TRIZ theory into graphic advertising design, focusing on modern creative thinking methods based on TRIZ theory.

C. Creative Thinking Method of Print Advertisement Based on TRIZ Theory

This paper mainly introduces the combination principle, multi-purpose principle, and reverse action principle of TRIZ innovation principle to stimulate the divergent thinking, aggregation thinking, and conversion thinking in traditional creative thinking and produce the method suitable for the creative thinking of print advertisement [9]. Digital graphics are the technology for design and pictures on a computer screen. Computer graphics reflect the data on a computer display visually. Computer graphics are being used to design, create, model, and catalog videos and computer programs. Nowadays, almost every machine can render such graphics, and people even anticipate controlling their device with icons and pictures instead of just typing. Design is related to areas of architecture and engineering. One of the key characteristics of most design concerns is the absence of a unique solution. The developer would then evaluate and then potentially modify a potential project to find a better solution.

Rational criteria guide the whole process, and abstract imagery is driven by specific principles to seek the thinking origin of the creative design essence of print advertisement Point. Fig. 2 shows the case design of computer print advertising under the TRIZ innovation theory thinking method.

Fig. 2. Computer graphic advertisement design case under TRIZ innovative method.

1. The Method of Stimulating Divergent Thinking With the Principle of Multi-function

Hegel, a famous German philosopher, once said that "creative thinking needs rich imagination." Divergent thinking is listed as the first creative thinking in design. According to the multifunctional principle of TRIZ theory, the principle that an object has many different functions is used to stimulate divergent thinking. A theme has multiple interpretations, and an item has multiple uses, which is an essential manifestation of the principle of multi-function. When the object has versatility, it can make it have more cooperation and value-added effect at any time, place, and environment. Most of its facilities were used by computer graphics in the film and gaming industries. It has applied to movies, TV shows, cartoon animated music video, and production. In the spaces where focus and interactivity are the key players, computer graphics help provide these functions effectively. Specialized training techniques, such as simulators, may prepare applicants to be understood better within a limited period. It is easy and very useful to develop computer graphics training modules.

2. The Method of Using the Combination Principle to Stimulate Aggregation Thinking

The combination principle of TRIZ is to combine the same objects in space or time or objects that complete a similar operation. The deconstruction and reorganization of design elements and the introduction of new colors, new textures, and new materials into the old objects are important means of operation of the combination principle, leading to aggregation thinking through the combination principle. If the complete image is divided into a single visual element and repeatedly distributed on the screen, the same or similar methods are used to "process" one by one to make the main body repeat repeatedly and set a certain order to appear, adjust the size of specifications or color texture, etc., the final effect is often creative and has excellent visual impact. The representative's graphic instruments are used as teaching aids to educate students in the classroom. In many applications for special purposes, the computer graphics industry is commonly used. The performance of computer graphics in business applications has been seen in several ways. In its implementations the importance of computer graphics lies. The physician will view this large volume of data in new and useable ways through interactive computer graphics. Computer graphics has extended the parameters of art and entertainment.

3. The Method of Using the Opposite Principle to Stimulate the Transformation of Thinking

On the contrary, the principle is a different way of thinking. When using the opposite principle to stimulate the change of thinking, poster design is often concise and powerful, can lead the audience into profound thinking. Specific principles include:

1. Attribute conversion: the objects with opposite attributes are exchanged to generate new images, such as size, speed, weight, etc.;
2. The transformation of position and structure: to arrange things in the place that should not appear and change the typical structure of something;
3. Process transformation: reverse the natural law of development of things;
4. Theoretical conversion: convey the profound significance and arouse the resonance and reflection of the audience.

II. Content Arrangement of Computer Print Advertisement Based on a Genetic Algorithm

A. A Business Model of Print Advertising Design

The basic description and relationship of the business model [10] of print advertising design are shown in Fig. 3. Graphic designer projects in these areas can be made, including billboards, posters, logos, ads, brochures, magazines, book covers, newsletters, product packaging, website, T.V. Graphics, ads, show, video and film graphics, and graphical computer graphics. When dealing with text, graphical designers may collaborate with copywriters to fit the image of the creators. They collaborate with art managers, design directors, creation managers, account managers, printers, images, illustrators, and web developers.

Service: basic information of media channel, including number, name, description, URL or FTP address of receiving content, etc. Large advertising agencies, magazine, or design companies hire graphic designers and frequently operate in cost, lighted environments. Due to production schedules, graphics designers employed in advertisements like printers and publishers can work evenings or weekends, with shorter and more frequent deadlines. Graphic designers usually work full time in these environments. Some are self-employed.

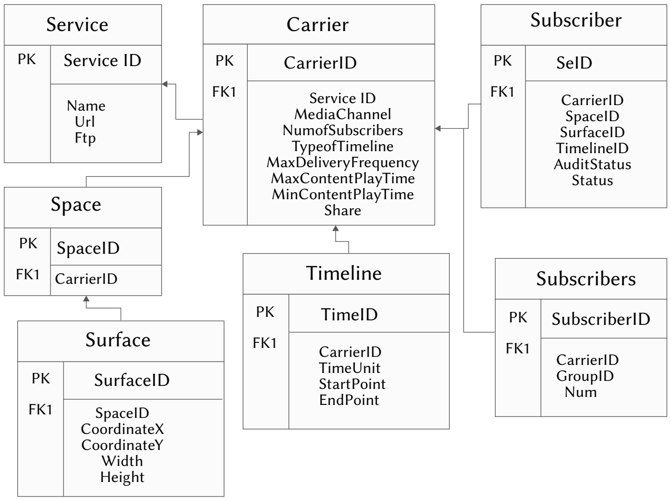


Fig. 3. Business model entity relation diagram of computer graphic advertisement design.

Carrier: the content carrying resource management that business or media channels (such as MMS Mobile News) can provide is called "carrier." All carriers can be expressed through investigation and analysis as three dimensions: space, time sequence, timeline, and subscribers. Besides, the carrier includes the basic requirements of the content arrangement, such as the maximum number of times that the same content is allowed to be put in the carrier's time axis interval, the maximum and minimum time sequence length of a single content the carrier. Web design is the expertise in creating content presentations typically delivered to the end-user through a Web browser, using hypertext or hypermedia. Different disciplines, such as animation, imagination, design of communication, business identity, graphical creation, human-computer interaction, the architecture of the content, interaction design, marketing, photography, optimization of the search engine, and typography, can be used for developing Web pages, web sites or multimedia applications.

Space: space resources are all templates and split-screen layouts created by the business administrator under a particular business. The plane is split into multiple pieces and split-screen styles. For example, Flipboard applications use rich template styles to render content.

Timeline: time-varying temporal resource description, mainly for streaming media such as focus media, digital TV, or multi-frame sequence of MMS (limited to one day). The time interval with the same business characteristics on the time axis is defined as a timeline object. Virtual Reality (VR) is a technology that allows a user to connect in a simulated computing world. For instance, simulation for pilot or combat training may be close to the real world. Virtual Reality is widely used to describe a broad range of applications often associated with its realistic, high visual 3D environment. For gaining insight into or track these processes' function, computer simulations have become a valuable part of mathematically modeling many natural systems in physical (computer physics), chemical and biologically, human systems in economies, psychology, and socio-science, as well as modern technology.

Subscribers: the audience category group of the business synchronized with the precision system, and each group of users is orthogonal and non-overlapping. In the application of CMS and its support, the audience groups can be distinguished by certain descriptive names, such as "business office building group," "family residential building group," etc., or certain user category labels such as gender, age, occupation, income range, etc. all combinations and user list binding are based on the domain values of each tag. The influence of digital technology has changed traditional activities, such as painting, drawing, and sculpture. In contrast, creative activities have been recognized for new forms such as network art, digital installer art, and virtual reality.

Subcarrier: a subcarrier is a resource block that combines specific spatial resources and material resources in a carrier. Generally, it is an independent content bearing resource block created by business administrators, which is mapped in the mobile advertising system, that is, advertising space, such as the rolling caption box at the bottom of the video terminal from 8:00 to 10:00.

Order: order is a transaction unit in CMS. It is initiated by CP or business administrator. It indicates the binding relationship between content and subcarrier and transaction price (input for order arrangement) and records transaction approval and operation status information.

B. Mathematical Model of Content Arrangement Model

For describing and simplifying the model solution with a mathematical language [11], this paper makes the following model assumptions and symbolic representation for the content arrangement problem.

1. The carrier is the resource object to be planned in the media layout model. In SMS / MMS, e-magazine, or advertisement, the space resource of carrier has the first dimension's priority. Therefore, multiple layout models can be constructed according to a specific location. It is simplified as a multi-group planning problem of the time axis and audience resources based on the spatial dimension.
2. Time axis resources are more important and valuable than audience resources in the content arrangement. So, the time axis is regarded as the second dimension, and the audience resources are considered to be the i-dimension resources based on the time axis.
3. Order scheduling input (*Input*) is the model's input data, which is defined as positive I_i, $i \in [1, N]$ according to the order of entering the model, where N is the total number of inputs in the model; the transaction price is P_i, $P_i \in R^+$.
4. The media industry attaches great importance to the same timeliness and strong publicity effect of the whole audience when making the advertising layout table. For example, an advertisement is mainly broadcast on multiple TV stations at 11:59. This model supports the advertising layout's professional requirements; that is, the same order content is released to all selected audiences in the same period.
5. The model supports the accurate delivery of content by audience groups, which can be arranged and delivered to the user list associated with multiple audience groups. The number of users in the audience group can be expressed as the vector $\vec{S} = \{s_j | j \in [1, G]\}$, where G is the total number of audience groups, S_j is the number of users in the group. A positive input audience group selection $Input\ I_i$ is described as $\vec{S_i} = \{s_{ij} | j \in [1, G]\}$ in which $s_{ij} = \begin{cases} 1, select\ j\ for\ Input\ I_i \\ 0, otherwise \end{cases}$.
   If $Input\ I_i$ does not accurately select the audience group; then it does not exist. The model supports the imprecise selection of audience number, i.e., the number of audience for content delivery can be specified during order arrangement input, which is defined as $Total_i$, $i \in [1, N]$. If the target audience I_i is precisely selected, the number of audiences I_i is defined as:
   $$Total_i = \vec{S_i} \bullet \vec{S} = \sum_{j=1}^{G} s_{ij} \bullet s_j \tag{1}$$
6. The time axis resources of the carrier can be continuous or discrete. In the actual content filling and advertising sales, the transaction

unit of time sequence often has time granularity. For example, the advertising space of G3 media requires that the content playtime should be multiple of 5 seconds [12]. The time series is divided into the smallest granularity in the model, and the continuous-time periods are discretized. The planning of time axis resources is based on a discrete sequence. Therefore, to simplify the algorithm and model description, it is assumed that there is a one-time axis resource of the carrier, which is defined as $[0, T]$. Suppose the primetime interval is defined as: 11:45-12:15, with 5 seconds as the time granularity division unit. In that case, there are 360 value sets in the time axis discrete sequence, that is is $T = 360$, the time interval is described as $[0, 360]$.

7. The model supports the content delivery at the precise time; the content sequence length is defined as l_i, $l_i \leq T$; the expected precise timing point d_i, $d_i \in [1, T]$ $d_i \in Z$. The penalty factor for the advance or delay of order scheduling in unit price unit time unit is α, $\alpha \in$ [0, 1].

C. Definition and Objective Function of Model Variables

1. Definition of Model Variables

$X_{ij}(t)$, where t is a time sequence point on the timeline resource, which means that the content arrangement starts at the t time point and ends at the $t + l_i$ time point:

$$X_{ij}(t) = \begin{cases} 1, I_i \ start \ at \ t \ from \ s_j \\ 0, otherwise \end{cases} \tag{2}$$

2. Objective Function

Suppose the time sequence resources are occupied by the historical order input too early. In that case, the remaining complete time sequence for the whole audience is too few, which will affect the sales of the carrier resources in the advertising system. Therefore, after the order arrangement input is arranged, the order arrangement input of the audience group with no conflict or inaccurate audience is arranged for the same period. It will save time axis resources as shown in Fig. 4, when *Input*2 does not need to refine the audience group, the blank time axis resources of the whole audience. Therefore, the larger the frequency of different orders sharing, the better.

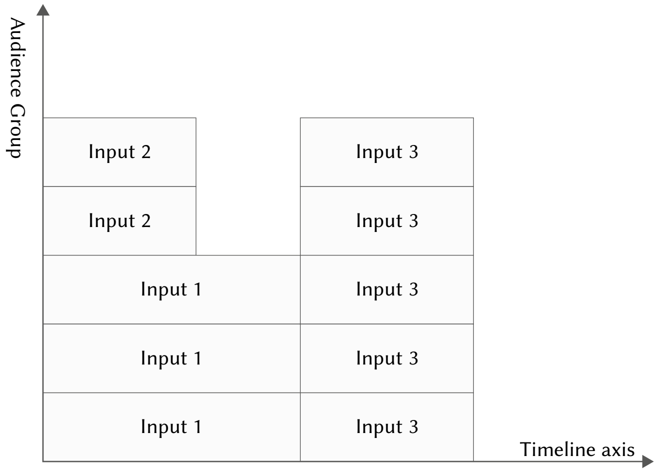


Fig. 4. Time axis and audience resource planning.

The objective is expressed by a mathematical formula as follows:

$$\max \sum_{t=0}^{T} \left[\sum_{r=t}^{t+l_i} X_{ij}(r) \right]^2 \tag{3}$$

The arrangement results should satisfy the number of audience and group selection in the order. When the number of audiences is specified in the order, the system needs to arrange specific audience groups for it [13]. In most cases, the audience group set's total audience cannot be precisely equal to the older audience's number of days. It is necessary to reduce the deviation ratio of the number of audiences as far as possible. It is because if the number of arranged audiences is less than the number of orders required, then the revenue of billing and settlement based on the actual delivery results will be lost; if the number of arranged audiences exceeds the standard, then the more released audience resources do not belong to the billing and settlement scope of the order. There is no benefit to the system. It is undoubtedly a waste of audience resources. The objective is expressed by a mathematical formula as follows [14]:

$$\min \frac{\left| \sum_{j=1}^{G} X_{ij}(t) \bullet s_j - Total_i \right|}{Total_i} \times 100\% \tag{4}$$

When the order scheduling input has the requirement of precise time delivery, if the scheduling result does not meet the requirement, it will cause a penalty in advance or delay. Therefore, the penalty amount should be minimized

$$\sum_{i=0}^{N} \sum_{j=0}^{G} \alpha P_i \bullet \left[\sum_{t=0}^{d_i - 1} (d_i - t) \bullet X_{ij}(t) + \sum_{t=d_i+1}^{T} (t - d_i) \bullet X_{ij}(t) \right] \tag{5}$$

Any two different order scheduling inputs I_i, I_k, $\forall i \neq k$, $i, k \in [1, N]$ cannot occupy the same user group in the same period. Otherwise, it will cause a resource conflict, which belongs to the category of infeasible solution. The mathematical expression of the constraint is expressed as follows:

$$X_{ij}(r) = 0, \forall r \in [t, t + l_k], if \ X_{kj}(t) = 1 \tag{6}$$

As the basis of the order price, the certainty required in the arrangement and input of order type demand must be met or guaranteed as far as possible. The results of the agreement should meet the requirements of the number of audiences for each order arrangement input

$$\sum_{j=1}^{G} X_{ij}(t) \bullet s_j \geq Total_i \, , \forall i \in [1, N] \tag{7}$$

For the order arrangement input with accurate audience group requirements, the arrangement results should meet the needs of audience group selection, which is expressed by a mathematical formula as follows:

$$\begin{gathered} \vec{X}_i(t) = \left(X_{i1}(t), \ldots, X_{ij}(t), \ldots, X_{iG}(t) \right) = \vec{S}_i \\ if \ \vec{S}_i \neq 0, \forall i \in [1, N] \end{gathered} \tag{8}$$

The resource planning model of the multi-objective optimization problem is constructed.

III. A genetic Algorithm for Solving the Mathematical Model of TRIZ Theory-based Advertisement

A. Solving Problems

Generally speaking, there is no an optimal solution for multi-objective optimization problems. All possible solutions are called non-inferior solutions, known as Pareto solutions [15]. To improve the performance of one sub-target, the understanding of the other sub-targets may decrease. Therefore, multiple sub objectives cannot achieve optimal results at the same time. It can coordinate the compromise among the sub-objectives, and finally, get the better solution.

The solution to the classical combinatorial optimization problem mainly depends on constraints. As long as the conditions are sufficient, the optimal combination scheme can be uniquely determined. As the

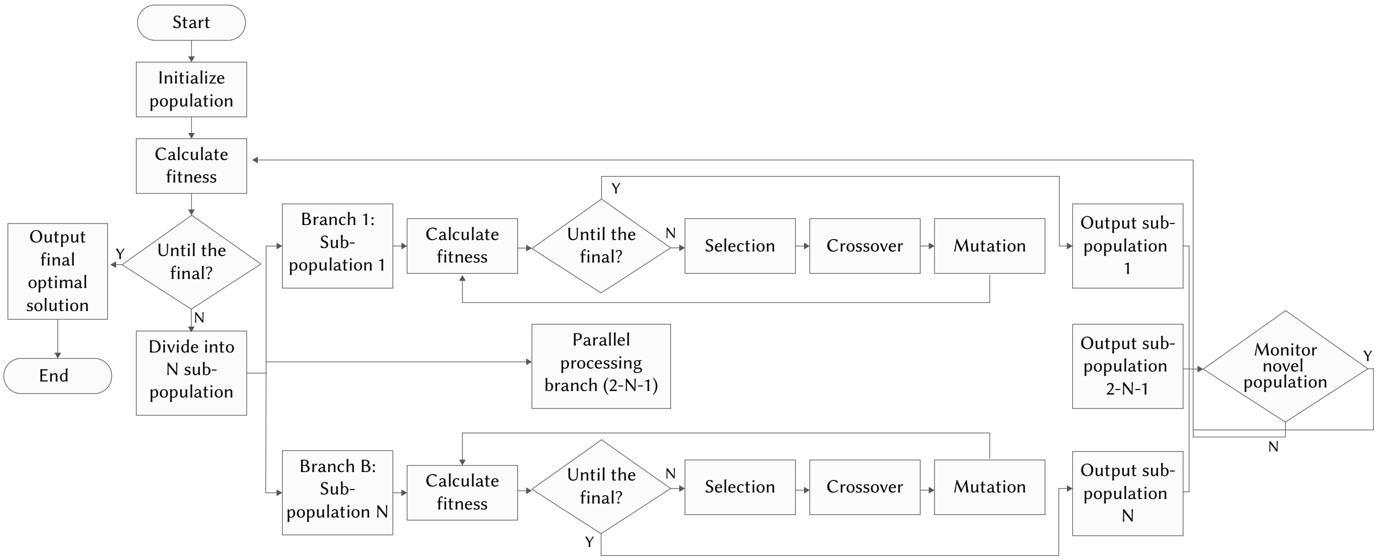


Fig. 5. Mathematical model of TRIZ theory based advertisement solved by the genetic algorithm.

number of composite objects increases rapidly, the solution becomes very complicated. It is called a "combined explosion." The classical programming method is usually feasible in theory for solving such problems. It is not suitable for practical issues. Therefore, how to take appropriate measures to suppress the "combination explosion" from the actual situation and to adopt the proper algorithm to reduce the search space has become a key problem in the arrangement and solution.

B. *Design of Parallel Genetic Algorithm*

A genetic algorithm (GA) [16] is an imitation of the mechanism of selection, crossover, and mutation in the process of biological genetics and evolution. For completing the search algorithm for the optimal solution of the problem, a genetic algorithm has the characteristics of parallel search, population optimization, and strong robustness. It has been widely used to solve nonlinear programming problems. At present, several commonly used multi-objective genetic algorithms are parallel selection method, non-inferior hierarchical genetic algorithm, a genetic algorithm based on objective weighting method, multi-objective particle swarm optimization algorithm. In this model, because the range of the objective function cannot be normalized, it is not suitable to use the accurate weighting method. Therefore, the parallel selection method with relatively simple selection steps is called "vector estimation multi-objective genetic algorithm [17]".

The core idea of genetic algorithms based on parallel selection to realize the content arrangement model is: using the gene coding value of chromosome to express the arrangement result; dividing the initial population into several subpopulations according to the number of targets; each population solves the whole space of genetic algorithm under a single objective function assigned by each population and the constraint conditions, and focuses on the local search with high performance. It is not easy to fall into the minimal local solution space, as to generate new subpopulations: merge all the new subpopulations, select cross mutation, recycle the previous processing to the termination condition, and then obtain the Pareto optimal solution of the problem. The flow chart is shown in Fig. 5.

C. *The Solving Process of the Parallel Genetic Algorithm*

1. Chromosome Coding

In this paper, the binary coding strategy is used to map the problem's solution space to a binary string space {0, 1}. The selection principle of bit string length is based on the problem's precision and range of solution. The value of the bit string should cover all values of uniform discretization in the variable definition domain. Considering the definition of model variables, three types of gene atoms should be declared: the starting time point of order content, the end time point of order content, and the audience group. Three gene atoms were concatenated to form a gene representing the arrangement result. Then, *N* genes were sequentially connected according to the number sequence of N orders, forming chromosomes.

Suppose three orders for the problem to be solved in this paper, with 5 audience groups. Then, the number of digits of chromosome can be defined as 45, the number 1-5 represents the starting time point of order No.1, bits 6-10 represent the end time of order No.1, and bits 11-15 describe the selection of order No.1 to five audience groups (multiple groups can be selected). The second to fifth-order genes were organized according to the first gene structure and then continued successively, forming 45 chromosomes. In this way, the solution space is mapped to the gene space of chromosome. As long as the chromosome is decoded according to the inverse process of tissue structure, the solution can be obtained.

2. Initial Population Selection

The initial population has diversity. It can be obtained by the random method and selective experience method. Considering that some orders have specific requirements to meet, such as selecting audience groups, precise time delivery, etc., this paper uses the selection experience method to generate the initial population meeting these deterministic requirements. Therefore, the constraint condition (F) and objective function (C) of the model degenerate and disappear (except when the gene mutation operation occurs), the solution space is converged in the initial population selection. The generated population is closer to the optimal solution space than the random population, improving the solution speed. Because of the parallel selection genetic algorithm, the initial population size is divided into three populations.

3. Fitness Function

Individual fitness is the basis of a genetic algorithm for the survival of the fittest. The most commonly used fitness evaluation method, namely "original fitness function, " directly uses the problem's objective function as the fitness function. The three constraints in the model are constructed into penalty functions, which need to be attached to three objective functions. As the fitness function of the three populations, F_1, F_2, F_3.

4. Selection, Crossover, and Variation

Selection is a survival process of the fittest, and those with high fitness can be inherited and copied to the next generation with high probability [18]. There are many selection operators, such as elite individual retention strategy, tournament selection method, etc. Because of the complexity of the layout model, this paper chooses the roulette method. The selection probability of individuals is as follows:

$$P_i = \frac{f_i}{\sum_{j=1}^{N} f_j} \tag{9}$$

Among them N is the population size and f_j is the fitness value of the $j-th$ individual in the population. The realization process is as follows: random number r is generated in the interval [0, 1] if the conditions are satisfied:

$$\sum_{j=0}^{i-1} P_j \leq r \leq \sum_{j=0}^{i} P_j \tag{10}$$

Then the $i-th$ individuals were selected, where $P_0 = 0$.

Crossover [19] is when some genes of two parents replace each other to produce new individuals. In this paper, Authors use the uniform crossover method: randomly generate binary bit string of chromosome length as the crossover template, where 0 means no exchange, 1 means exchange; according to the template, Authors can get new individuals. Generally, the crossover probability is [0.4, 0.99], and the crossover probability used in this paper is 0.6.

Mutation [20] is to change the algorithm's local search ability and maintain the population's diversity to change some genes of individuals on a binary coded chromosome. The mutation is reversing the gene value of some genes, that is, 1 to 0, 0 to 1. The method of uniform variation is used in this paper. The mutation probability was set as 0.05. Generally, the evolutionary algebra is 100-1000 times. Due to the parallel selection genetic algorithm, each population adopts 200 times of evolutionary algebra.

IV. Simulation Experiment and Result Analysis

A. Simulation Results

In the general microcomputer environment, Python software's parallel genetic algorithm is implemented [21]. Suppose the current mobile advertising business. The carrier resources are composed of 1 spatial location, five audience groups [1,5], and 10-time units [1,10]. Due to media resources limitation, it is necessary to plan orders in these timelines and audience resources. The number of audience groups is set, as shown in Table II.

TABLE II.The Number of People in the Audience Group

Audience group j	1	2	3	4
Number of humans S_j	180	120	230	280

The number of input data of order arrangement increases by 5 at a time. The values are {5, 10, 15, ..., 45, 50}. The input data is shown in Table III. Assuming the penalty coefficient $\alpha = 0.3$, the parallel genetic algorithm proposed in this paper is used to solve the problem.

TABLE III. Input Data of Partial Order Arrangement

Order input I_i	Price P_i	Audience Group $\vec{S_i}$	Audience Number $Total_i$	Sequence start d_i	Time length l_i
1	290	{1,1,1,0,0}	480	0	2
2	240	-	420	-	1
3	480	{1,1,1,1,1}	780	-	3

Solve the model variables based on $X_{ij}(t)$, and transform them into natural language description results, as shown in Fig. 5. Table IV shows part of the operation results, and the simulation results show a better solution based on the model objectives and constraints. It is verified that the model can solve the problem of content arrangement.

TABLE IV. Partial Results of a Parallel Genetic Algorithm Based on TRIZ Theory

Order input I_i	Audience Group $\vec{S_i}$	Audience Number $Total_i$	Sequence start d_i	Time length l_i	Order input I_i
1	{1,1,1,0,0}	476	0	2	1
2	{0,0,0,1,1}	418	0	1	2
3	{1,1,1,1,1}	778	2	3	3

B. Analysis of Experimental Results

In this paper, the weighted average method is used to set each objective's weight coefficient based on the importance of each objective, and the multi-objective is transformed into a single objective. The classical dynamic programming method is used to solve the problem. The simulation efficiency is shown in Fig. 6. With the increase of the problem's scale, the number of iterations and the running time of the dynamic programming method [22] increase exponentially; that is, there is a combinatorial explosion problem. In contrast, the parallel genetic algorithm's time performance is relatively stable, and the average convergence time is shorter than that of the dynamic programming method. When the order size is more than 30, the parallel genetic algorithm is suitable for solving the problem.

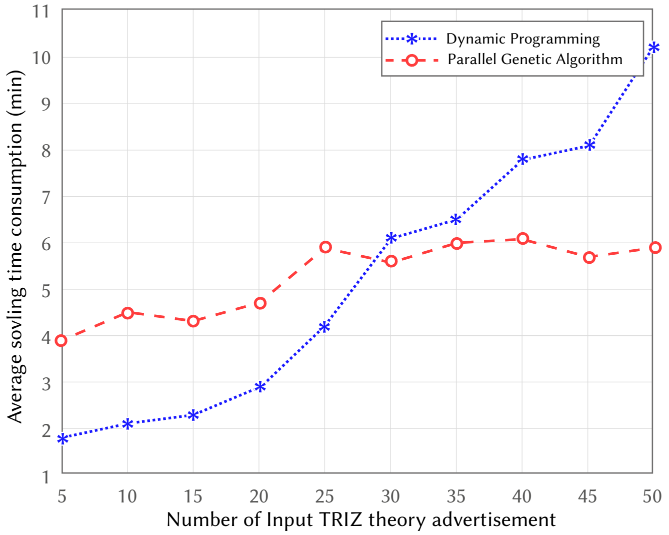


Fig. 6. Comparison results of dynamic programming algorithm and parallel genetic algorithm.

When there are a few orders, a relatively simple dynamic programming method can be used. Therefore, the parallel genetic algorithm has a high application value in engineering with more demands, complex scheduling requirements, and non-real-time request-response. When the problem's scale is enlarged and the time precision is improved, the chromosome's binary code string will overgrow. The real coded genetic algorithm can be used to design chromosomes and algorithms, reducing a genetic algorithm's search space. The increasing algorithm's evolutionary algebra method is used to increase the number of individuals in the solution space to improve the Pareto optimal solution's probability. Designers work for smaller develop outsourcing companies or others who work freelance work on the job, or the contract basis adapts their working day to the customers' schedules and deadlines. Freelance designers have to satisfy customers and look for new ones to sustain their stable incomes. In this setting, graphic designers do freelance work full-time or part-time in addition to a paying job in the design or other workplace. In advertising, the graphic designers use details such as a customer's desires, planned design messages, and appeal before producing a new design for consumers or users. Graphic designers collect and conduct their research on their projects through interactions with buyers, creative or art directors. When graphical designers acquire this knowledge, they create designs or layouts to output their design visions by hand or using a computer and include colors, music, artwork, photography, animation, design styles, and other visual objects. Graphic designers then choose an element to view on a website or screen, create graphs and diagrams of data to be released and consult copywriters on text accompanying the design. Clients or art/creative directors are then presented with completed designs.

V. Conclusion

This paper verifies the feasibility of the combination of TRIZ theory and creative thinking through logical analysis, analyzes the process of stimulating divergent thinking, aggregation thinking, and transformation of innovative thinking methods from the innovation principle of TRIZ theory as the origin, and applies it to the creative mechanism and application program of print advertising creativity. The specific principles and rational principles can stimulate perceptual thinking and abstract imagination, seek breakthroughs based on original creative thinking methods, explore the source of thinking, and provide a more feasible and practical creative thinking method of print advertising. It gives the application basis for the development of TRIZ theory in many fields and provides detailed theoretical guidance for the construction of print advertising's creative thinking method. Based on the TRIZ innovation theory, computer print advertisement design needs to complete the arrangement of content. In this paper, the media resources of content media arrangement are constructed into a three-dimensional resource model that takes the carrier as the core and describes the template screen space, time sequence resources, audience grouping, and other three-dimensional resources model. A multi-objective resource optimization mathematical model content arrangement is constructed based on the order demand constraints, high resource utilization, and accurate delivery time goal. Finally, a multi-objective optimization mathematical model of a parallel genetic algorithm is designed to solve the advertisement content arrangement. Experiments verify the innovative thinking of TRIZ and the application of genetic algorithm in content arrangement of computer graphic advertisement design.

References

[1] C. H. Lee, C. H. Chen, F. Li, A. J. Shie, "Customized and knowledge-centric service design model integrating case-based reasoning and TRIZ," *Expert Systems with Applications*, vol.143, pp.113062, 2020, https://doi.org/10.1016/j.eswa.2019.113062.

[2] F. Li. "A Study on Innovative Design of Rotary Pile Foundation Drilling Machine Based on TRIZ Theory," *In International Conference on Mechanical Design, Springer, Singapore*, 2019, pp. 302-309.

[3] N. Şen, Y. Baykal. "Development of car wishbone using sheet metal tearing process via the theory of inventive problem-solving (TRIZ) method", *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol.41, no.10, pp.390,2019, https://doi.org/10.1007/s40430-019-1884-7.

[4] F. Jiang, J. Shen, T. Zhu, J. Wen. "Design of Flagstone Transport Device Based on TRIZ Theory", *In International Conference on Mechanical Design*, Springer, Singapore, 2019, pp. 253-266.

[5] D. Russo, C. Spreafico, M. Spreafico. "A Simplified TRIZ Approach Involving Technology Transfer for Reducing Product Energy Consumption", *In Sustainable Design and Manufacturing*, Springer, Singapore, 2020, pp. 129-138.

[6] H. D. García-Manilla, J. Delgado-Maciel, D. Tlapa-Mendoza, Y. A. Báez-López, L. Riverda-Cadavid. "Integration of design thinking and TRIZ Theory to assist a user in the formulation of an innovation project", *In Managing innovation in highly restrictive environments*, Springer, Cham, 2019, pp. 303-327.

[7] K. Hmina, M. Sallaou, A. Arbaoui, L. Lasri. "A preliminary design innovation aid methodology based on energy analysis and TRIZ tools exploitation", *International Journal on Interactive Design and Manufacturing*, vol.12, no. 3, pp. 919-928, 2018, https://doi.org/10.1007/s12008-017-0455-3

[8] G. Bersano, P. E. Fayemi. "Application of TRIZ and Innovation Management Theory on Decision Support for Transport Infrastructure", *In International TRIZ Future Conference*, Springer, Cham, 2019, pp. 486-493.

[9] P. Livotov, A. P. C. Sekaran, R. Law, Reay. D. "Systematic Innovation in Process Engineering: Linking TRIZ and Process Intensification*", In Advances in Systematic Creativity*, Palgrave Macmillan, Cham, 2019, pp. 27-44.

[10] S. Purnamawati, E. B. Nababan, B. Tsani, R. Taqyuddin, R. F. Rahmat. "Advertisement scheduling on commercial radio station using genetics algorithm", *2nd International Conference on Computing and Applied Informatics*, Medan, Indonesia, 2017, pp. 28–30.

[11] Q. Madera, O. Castillo, M. Garcia, A. Mancilla. "Interactive evolutionary computation with adaptive mutation for increasing the effectiveness of advertisement texts", *In 2016 IEEE Symposium Series on Computational Intelligence*, IEEE, 2016, pp. 1-6.

[12] G. T. Reddy, M. P. K. Reddy, K. Lakshmanna, D. S. Rajput, R. Kaluri, G. Srivastava. "Hybrid genetic algorithm and a fuzzy logic classifier for heart disease diagnosis", *Evolutionary Intelligence*, vol. 13, no. 2, pp. 185-196, 2020, https://doi.org/10.1007/s12065-019-00327-1.

[13] L. Qin, W. Huang, Y. Du, L. Zheng, M. K. Jawed. "Genetic algorithm-based inverse design of elastic gridshells", *Structural and Multidisciplinary Optimization*, vol.62, pp. 2691-2707, 2020, https://doi.org/10.1007/s00158-020-02639-8.

[14] M. Guo. "A Study on Data Mining of Digital Display Performance of Brand Advertisement", *Wireless Personal Communications*, vol. 102, no. 2, pp. 1243-1253, 2018, https://doi.org/10.1007/s11277-017-5180-5.

[15] K. M. Hamdia, X. Zhuang, T. Rabczuk. "An efficient optimization approach for designing machine learning models based on genetic algorithm", *Neural Computing and Applications*, vol.33, pp. 923–1933, 2021, https://doi.org/10.1007/s00521-020-05035-x.

[16] Z. Zhou, F. Li, H. Zhu, H. Xie, J. H. Abawajy, M. U. Chowdhury. "An improved genetic algorithm using greedy strategy toward task scheduling optimization in cloud environments", *Neural Computing and Applications*, vol. 32, no. 6, pp. 1531-1541, 2020, https://doi.org/10.1007/s00521-019-04119-7.

[17] H. Chung, K. S. Shin. "Genetic algorithm-optimized multi-channel convolutional neural network for stock market prediction", *Neural Computing and Applications*, vol. 32, no.12, pp. 7897-7914, 2020, https://doi.org/10.1007/s00521-019-04236-3.

[18] J. Tian, M. Gao, G. Ge. "Wireless sensor network node optimal coverage based on improved genetic algorithm and binary ant colony *algorithm*", *EURASIP Journal on Wireless Communications and Networking*, 2016:104, 2016, doi: 10.1186/s13638-016-0605-5.

[19] T. L. Chen, C. Y. Cheng, Y. H. Chou. "Multi-objective genetic algorithm for energy-efficient hybrid flow shop scheduling with lot streaming", *Annals of Operations Research*, vol. 290, pp. 813-836, 2020, https://doi.org/10.1007/s10479-018-2969-x.

[20] N. Malarvizhi, P. Selvarani, P. Raj. "Adaptive fuzzy genetic algorithm for multi biometric authentication", *Multimedia Tools and Applications*, Vol.79. no.13, pp. 9131-9144, 2020, https://doi.org/10.1007/s11042-019-7436-4.

[21] C. Ma, W. Hao, F. Pan, W. Xiang. "Road screening and distribution route multi-objective robust optimization for hazardous materials based on neural network and genetic algorithm", *PLoS One*, vol. 13, no. 6, pp. e0198931, 2018, https://doi.org/10.1371/journal.pone.0198931.

[22] A. Mondal, J. M. Young, T. A. Barckholtz, G. Kiss, L. Koziol, A. Z.

Panagiotopoulos. "Genetic algorithm driven force field parameterization for molten alkali-metal carbonate and hydroxide salts", *Journal of Chemical Theory and Computation*, vol. 16.no. 9, pp.5736-5746, 2020, http://dx.doi.org/10.1021/acs.jctc.0c00285.

Yang Song

Yang Song, graduated from the Beijing Institute of Graphic Communication of Design art in 2008.Working in School of Academy of fine arts, Baotou Teachers College. Her research interests include Digital media art Design.

Computer Entertainment Technologies for the Visually Impaired: An Overview

Manuel López Ibáñez, Alejandro Romero-Hernández, Borja Manero, María Guijarro *

Complutense University, Madrid (Spain)

Received 17 October 2020 | Accepted 18 March 2021 | Published 23 April 2021

LA UNIVERSIDAD
EN INTERNET

Abstract

Over the last years, works related to accessible technologies have increased both in number and in quality. This work presents a series of articles which explore different trends in the field of accessible video games for the blind or visually impaired. Reviewed articles are distributed in four categories covering the following subjects: (1) video game design and architecture, (2) video game adaptations, (3) accessible games as learning tools or treatments and (4) navigation and interaction in virtual environments. Current trends in accessible game design are also analysed, and data is presented regarding keyword use and thematic evolution over time. As a conclusion, a relative stagnation in the field of human-computer interaction for the blind is detected. However, as the video game industry is becoming increasingly interested in accessibility, new research opportunities are starting to appear.

Keywords

Accessibility, Computer Entertainment, Human-computer Interaction, Video Games, Visual Impairments.

DOI: 10.9781/ijimai.2021.04.008

I. Introduction

Throughout the last decades, accessibility in interactive applications has been profoundly improved, especially when it comes to dealing with the experiences of blind and visually impaired people while performing everyday tasks. For example, screen readers [1]–[4] have greatly eased the process of interacting with computers, and nowadays, capable digital assistants abound [5]–[7], making simple, day-to-day interactions more accessible than ever for people with visual impairments, even though there is still much room for improvement in this area [8], [9].

However, entertainment technologies, and particularly video games, are still primarily visual, and often disregard the importance of universal access in this context. Even so, recent advancements, such as the notably deep accessibility measures that players can find in *The Last of Us: Part II* [10], have gathered a considerable amount of attention and praise from both users and the media. Adaptations of this quality are still difficult to achieve, however, as most companies will not allocate enough resources to cover the needs of a relatively small portion of their clients. This is the reason why research in the field of accessibility for the blind and visually impaired revolving around computer entertainment is still relatively scarce, but necessary in order to achieve much more welcoming virtual environments for everyone.

The goals of this article are two: on one hand, to thoroughly describe the state of the art in the field of accessibility for the blind or visually impaired; on the other, to detect trends in this area of knowledge.

* Corresponding author.

E-mail addresses: manuel.lopez.ibanez@ucm.es (M. López Ibáñez), alerom02@ucm.es (A. Romero-Hernández), bmanero@ucm.es (B.Manero), mguijarro@ucm.es (M. Guijarro).

II. Research Methodology

A. Search Terms

In order to collect relevant publications in this field, we combined the following search terms or expressions: "blind", "visually impaired", "visual impairment", "accessibility", "accessible technologies", "disability", "video games", "audio games" and "computer entertainment". These were used in Google Scholar, IEEE Xplore and ACM Digital Library, and results were limited to articles published from 1993 to 2020. A total of 102 articles were selected throughout this process, by considering their relevance to the subject. Due to the relatively small number of publications available in this area, articles were only rejected when their focus was not accessible video and audio games, or simply when they did not appear in any search engine after inputting the already-mentioned search terms.

B. Categorisation

As a means to classify all selected works, we started a preliminary review which consisted of reading their abstracts and keywords. After this, four general categories were created, and articles were distributed between them as Table I shows.

Thematic connections between articles were also explored during the classification process. Fig. 1 focuses on the distribution of each article according to this taxonomy. Coloured entries evince a relevant relationship between an article and a different category marked with the same colour; each of these articles still pertain to their main categories, but include references to subjects or methods present in a different one. Subcategories have also been included, in order to reach more precision regarding thematic classification.

C. Contributions

As a second step, articles were read and analysed one by one in order to fully understand their content, while extracting metrics from them.

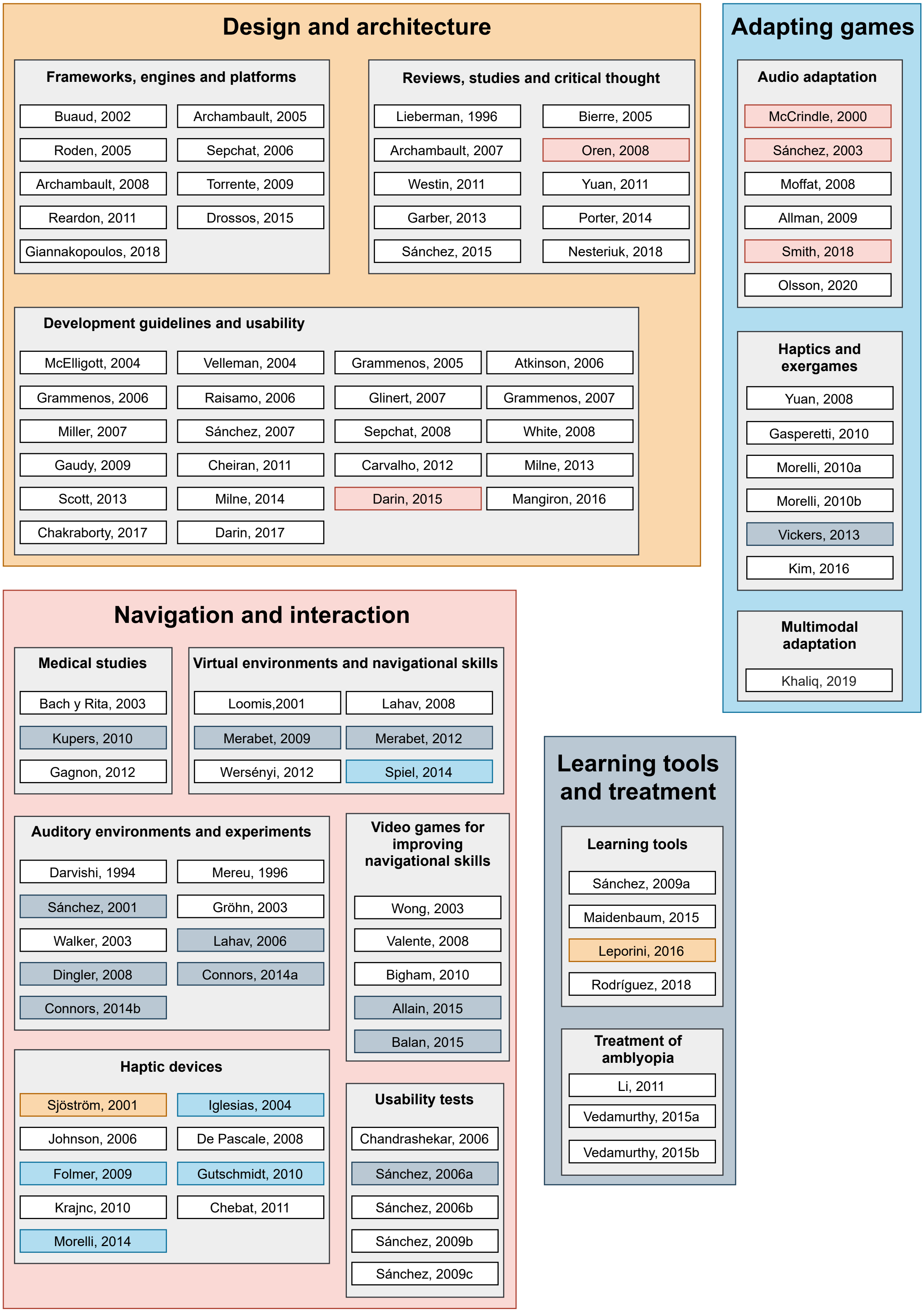


Fig. 1. Article classification depending on subject. Coloured references mean there is a significant relationship between said reference and another, secondary category, marked with that same colour.

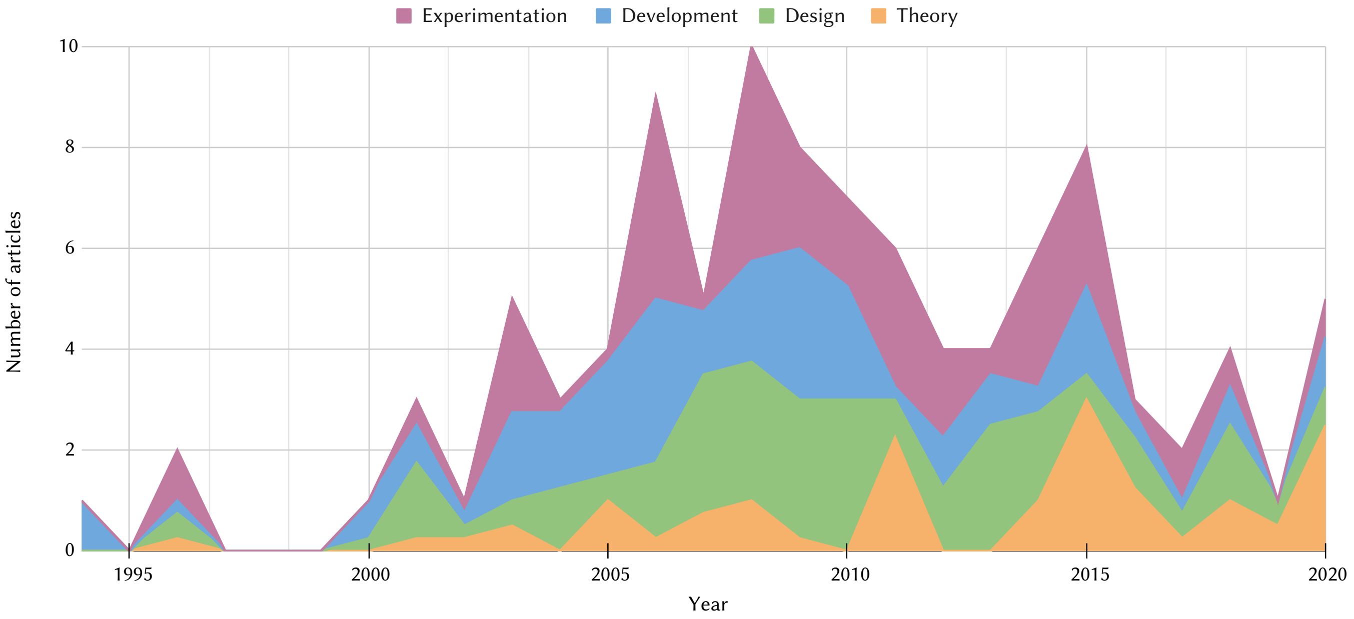


Fig. 2. Evolution of research approaches, in terms of relevant contributions, over time.

A survey was carried out to categorise the types of contributions made by each of them. To do this, we selected the four main contributions we were looking for in our analysis: 1) a theoretical contribution to the field, 2) the design of a tool, 3) the development of a tool, or 4) the development of an experimental study.

Then, we developed a system which allowed for assigning a numerical array score to every article depending on its contributions. A number between 0 and 1 was assigned to every mentioned contribution type so that every article's aggregated value was always exactly 1. Each category could have 5 different values: 0, 0.25, 0.5, 0.75 and 1. Valid distributions would be, for example: [0, 0.25, 0.75, 0] or [1, 0, 0, 0] The results of this analysis can be appreciated in Table II, where both the number of articles and their distribution in terms of contribution types are represented.

TABLE I. Number of Entries in Each Category

Category	Number of articles
Design and architecture of video games for blind or visually impaired players	43
Adapting games for the visually impaired	13
Video games as a learning tool or as treatment for people with visual impairments	9
Navigation and interaction of visually impaired players in virtual environments	37
TOTAL	**102**

Fig. 2 represents the research trends (experimentation, development, design, and theory) over the years. 66 papers present a tool design process, 61 delve into development, and 65 include experiments. However, only 21 articles offer a theoretical contribution.

D. Summaries

Lastly, contributions were summarised and explained in Sections III to VI.

Section III contains references to specific video games, design aids or software architectures designed with accessibility in mind, and focuses on final products, coming from both academic and industrial environments. Some of these games and interactive applications solve specific problems detailed in different sections, but were kept in this category due to them being final products or systems aimed towards facilitating the design of accessible video games. Systematic reviews or compilations addressing the design and architecture of accessible games are also included in this category.

Section IV delves into how non-accessible video games can be adapted to become usable by blind or visually impaired players. This is a narrow field, as it is generally considered suboptimal to adapt already-existent games instead of creating them from scratch with accessibility in mind. However, some of the ideas presented in this section are very interesting, and can be applied to the field of accessible game design.

Section V captures a new trend in this field: designing serious games as learning tools or as a treatment for the blind or visually impaired. This category includes both medical and educational applications, and presents technologies which could be applied to the design of accessible serious games.

Lastly, section VI covers the development of technologies aimed towards improving non-visual navigation and human-computer interaction in virtual environments. This field relies largely upon haptic or auditory user interfaces, some of which have already been integrated in accessible video games with positive results. This category also summarises different experiments carried out to test the effectiveness of a variety of tools created either to improve navigational skills or to determine how accessible an application or content is. This section also contains some studies about the effect of navigating virtual environments in the brain of non-sighted users.

III. Design and Architecture of Video Games for Blind or Visually Impaired Players

Most articles reviewed during this research process address everyday problems associated with developing accessible video games for the blind or visually impaired. For example, Archambault [11] proposes a game platform, named Blindstation, useful for solving technical problems revolving around accessibility and interactive experiences. Several games are presented, adapted and evaluated by blind children, and the conclusion is that these games constitute more than accessible applications, and can be considered a valid entertainment product by users.

TABLE II. Number of Research Articles per Type of Contribution. One Article May Appear in More Than One Category

Category	Theory	Design	Development	Experimentation
Design and architecture of video games for blind or visually impaired players	15	28	24	19
Adapting games for the visually impaired	2	12	11	10
Video games as a learning tool or as treatment for people with visual impairments	3	3	4	7
Navigation and interaction of visually impaired players in virtual environments	6	23	22	29
TOTAL	**26**	**66**	**61**	**65**

After his previous contribution, the same author [12] delves on the very idea of game accessibility through a systematic review. Academic articles published in the last decade are taken into consideration. Besides, an architecture for a framework allowing mainstream game accessibility is also included.

Eventually, this author produced an accessibility framework [13] allowing game developers to design accessible games, as well as assistive game interfaces.

The TiM project [14] presents a similar approach, in order to facilitate the development of accessible games. This project has helped design and adapt computer games for children with visual disabilities. A software architecture is defined, containing a proposed interaction design based on common auditory and tactile resources.

On the other hand, Roden and Parberry [15] describe a framework for designing 3D audio games aimed at wider markets, instead of exclusively made for the blind. The system is able to represent the world in a very simple and restrictive manner, so that players know exactly where they are at all times. Instead of using true 3D audio, the authors propose a 2D virtual stage which can support surround sound configurations of up to 6 speakers (5.1), but does not take into account sound source height.

The AGRIP project [16] focuses on the creation of modern first-person shooter games which are accessible to the blind and visually-impaired. As a result, the low quality of the accessibility infrastructure utilised in the video game *AudioQuake* is exposed. The use of this technology in educational settings is also criticised.

Reardon [17] presents a technology called AbES (Audio-Based Environment Simulator), which acts as a navigation aid for the blind. The idea behind the system is to help non-sighted individuals to learn routes and "paint" navigation maps in a similar way sighted people do. This kind of technology has been used to map real environments, such as a university campus or an underground station.

By now, it should be clear that options for blind people are scarce in the world of interactive entertainment. Thus, Carvalho, Guerreiro, Duarte et. al. [18] present an iterative and participatory design of an audio-based puzzle game, along with a preliminary evaluation with 13 blind participants. The outstanding results achieved highlight that the game is both fun and challenging for its target audience.

Another article by Bierre, Chetwynd, Ellis et al. [19] proposes a study about different commercial video games which implement accessibility. The authors conclude that there exists a clear need for accessibility in commercial games.

Velleman, Van Toi, Huiberts et al. [20] introduce a series of video games specifically made for the blind, as a result of a collaboration between the Utrecht School of the Arts and the Bartiméus Accessibility Foundation. These games include *Drive*, *The Curb Game*, *Hall of Sound*, *Powerchords*, *Wow* and *Demor*, among other titles. A detailed explanation is given on the design principles followed by each project, as well as their goals.

White, Fitzpatrick and McAllister [21] face the issues of navigation and feedback in 3D environments for the visually impaired by interviewing 8 expert users and letting them guide design proposals. The study is focused on multi-user virtual environments, and particularly on Second Life, a persistent virtual world. Multimodal approaches, including audio and haptics, are suggested as a solution for most accessibility issues in video games.

Defining non-functional software requirements is a critical task for the success of any information system. In the field of accessibility and usability, non-functional requirements are considered especially important when developing information systems. The work presented in [22] aims to better understand the non-functional software requirements needed to address accessibility and usability challenges for people with visual disabilities in video games. To achieve this, a study of the different analysis, results and recommendations derived from previous studies is detailed. Starting from a 2D arcade video game from the 1980s, these authors develop a new version where they incorporate hardware controls and sound effects, making it possible for people with visual disabilities to use it. Through interviews with six visually impaired users, they also determine that the experience with this new version is positive.

Sepchat, Monmarché, Slimane et al. [23] claim that the main senses involved in making a video game accessible for the blind are hearing and touch, and present a semi-automatic system which can help developers design accessible games controlled by tactile surfaces and described through audio. All of these games must be based on a 2D grid over which a character moves. Two examples are included: a maze game, in which the player has to find an exit while avoiding obstacles and enemies, and a snake game, in which the player is a snake that grows by eating apples and dies when hitting a wall or its own tail.

By considering the impact on human development that video games have, this work [24] proposes a 3D game that is based on the perception of sound and haptic feedback, leaving aside any visual interaction. This game has been tested with blind and blindfolded people, thus highlighting several interaction problems which needed to be solved. The software was then redesigned by taking into account these results.

A video game and a functional game engine aimed at children with visual disabilities are presented in [25]. On one hand, the game is implemented using binaural technology that allows players to listen and navigate through the game space by adding location information to audio. On the other hand, the engine presented by these authors enables a quick game development process for people with visual disabilities, this contribution being a good starting point for future developments in the field.

Gaudy, Natkin and Archambault [26] present the design of an audio game in which interaction is based on very simple actions, so that no instructions are needed before starting to play. Users are not supposed to fully understand all game mechanics from the beginning, though. Instead, they discover the intricacies of gameplay as the game progresses. An experiment aimed towards a usability validation is also presented in this article. Its results indicate that, even though all players managed to progress in the game, not everyone understood how specific mechanics worked.

Other authors [27] present a different approach for developing video games which are universally accessible. This way, people with visual and hearing impairments, multiple variants of color-blindness and reduced vision problems can be able to play and learn through them. These same authors also carried out simulation tests to prove the functionality of this perspective.

A set of inclusive and accessible games are presented in [28]. In all of these, accessibility is achieved through auditory representations of virtual environments. A development platform, named Memor-i studio, is also presented. It allows non-technical users (including blind people under supervision) to create inclusive games. Both the games and the platform have been tested by blind or visually impaired users with positive results.

AudiOdyssey, a video game prototype designed to be used by both sighted and visually impaired people, is presented in [29]. Said prototype is a rhythm-based game with fully accessible menus and game levels. It also allows for a variety of control schemes, so that it can adapt to every type of player.

Another research [30] describes the user interface design for the accessible and multiplayer game *Access Invaders*. This game simultaneously adapts to the needs of different player types by defining parallel game universes, thus allowing visually impaired people to play simultaneously with other users.

Sánchez and Hassler [31] build a virtual environment that contains a navigation aid for the blind and does not rely on 3D audio to function. Instead, these authors propose a system called AudioMUD, which uses spoken text as the only way to describe both virtual environments and player interactions. The system was evaluated through a cognitive testing process, concluding that collaboration skills and leadership can be developed while playing with this kind of technology.

UA-Chess, a fully accessible chess game which can be played through a standard web browser, is presented in [32]. Its main feature is its accessible nature: it can be played simultaneously by people with different disabilities, including visual impairments. Finally, this research analyzes how accessibility is supported in *UA-Chess* for different categories of users through its graphical user interface, its adaptiveness and the variety of input and output modes.

VBGhost [33] is presented as an educational, multiplayer game for blind or visually impaired people, which reinforces concepts related to reading and writing in Braille. Its main contribution to the field of game development is its interface, which uses six touch areas with a circular shape, enabling players to easily input letters in Braille.

These same authors [34] propose a compilation of their accessible and educational games, called *BraillePlay*, which includes four different titles: *VBReader*, *VBWriter*, *VBHangman* and the already-mentioned *VBGhost*. A formal evaluation of said games is also included, focusing on accessibility, engagement and effectiveness. The results show that engagement is the only issue when trying to get children to interact with these games for a period of time longer than four weeks.

Torrente, Del Blanco, Moreno-Ger et al. [35] present accessibility as a means to maintain inclusivity in e-learning processes. They review a variety of methodologies, tools and design patterns for accessible video games, and propose a series of guidelines for designing inclusive games for e-learning. A platform called <e-Adventure> is also presented, focusing on the generation of point-and-click adventure games. It contains a game authoring editor and a game engine. The article also concludes that accessibility is an often overlooked matter in the game development business, and that there is a need for tools which make adapting games to everyone easier and less expensive for professional game developers.

Other authors [36] present audio games as a low-cost alternative to portable video games for visually impaired players. An accessible sudoku game is described and implemented through RockBox, a custom firmware for popular MP3 players. Even though this article was written just before the popularization of smartphone games, it predicts the impact they would have in blind players.

McElligott and Van Leeuwen [37] propose a methodology for designing toys and computer games for the blind which involves taking into account their abilities, instead of their disabilities. Three different scenarios are provided as examples of inclusive design, and all of them are validated through experiments with small children. After these validation sessions, the authors conclude there is a need for more varied and balanced sensory stimulation in games for visually impaired children. Thus, video games that only use auditory stimuli have room for improvement if other communication channels are used, such as tactile surfaces.

Nesteriuk [38] defends the importance of audio games for general accessibility, and explains how they can help visually impaired people stay connected in a digital world. A survey about current advancements in the field is the main contribution of this author's work; however, he also introduces a series of design guidelines for accessibility in video games, based on the reviewed literature.

In a first approach towards improving game-related skills in people with visual disabilities, Darin [39] proposes a methodology to evaluate usability and cognitive impact in multimodal video games. This work is concluded with a list of guidelines for the design of interface elements.

Through further research, the same author [40] presents guidelines for the evaluation of usability in audio-based and haptic video games for blind people. A guide is also included, based on the comparison of usability evaluation methods (EMU) in this context, as well as an analysis on the evaluation of audio as a mode of interaction.

Grammenos, Savidis and Stephanidis [41] also describe the basic steps to adapt and apply a unified design method in the development of accessible video games. Key differences between turn-based strategy game design and action games are outlined.

Scott and Ghinea [42] describe the result of providing a list of accessibility guidelines to the participants at the Global Game Jam (GGJ), an event where small games are made in only a 48-hour development cycle. The authors conclude that designing games with accessibility in mind is possible even in a short time-frame, and that events like the GGJ can raise awareness about these issues among game developers.

Leporini and her colleagues explored another example of the potential use of serious games in rehabilitation in [43], with a particular focus on skills and retraining for blind and partially sighted people. The authors present a theoretical framework for developing this type of video games, and propose some design guidelines to follow when using these games in rehabilitation scenarios.

Lieberman [44] writes an article in an attempt to motivate our society to design universally accessible activities and video games.

Another article [45] analyses the current state of game accessibility, with a particular focus on the blind and visually impaired. They also discuss the barriers these groups face, as well as their options in the video game industry, such as audio games and "video-less" games. They conclude by suggesting the application of audio description techniques to video games, which could potentially improve accessibility.

Miller, Parecki and Douglas [46] provide a formal description of an auditory game specifically designed for blind people. The game requires players to match a rhythm with keystroke patterns, and includes an audio-based menu and a data collection tool that allows for gameplay analysis. This kind of software can also be used as an experimental tool for studying how blind players behave in a virtual auditory environment, and provides an unbiased performance variable (called "score"), which can potentially prove useful when judging the utility of certain game design decisions.

Oren, Harding and Bonebright [47] analyse and compare mental images created by both sighted and non-sighted players of a 2D, side-scrolling audio game. User descriptions of the virtual environment were retrieved and transferred to a map, and they were assigned a "mapping score" by judging how closely this mental image resembled the actual layout of the game. The authors conclude that there is not a significant difference in accuracy of mental maps created by sighted and visually impaired players, and that audio can convey enough information to allow for accurate virtual environment representation in video games.

Porter [48] takes into account mainstream gaming when addressing accessibility issues present in interactive software. He outlines a future research plan focusing on working together with actual game developers in order to improve how commercial products are accessed by people with disabilities.

Raisamo, Hippula, Patomaki et al. [49] describe how several haptic devices were used to generate feedback in multimodal applications. A method to conduct usability testing with visually impaired children is also explained.

Garber [50] presents a study where the need to make video games accessible is highlighted. To achieve this, a review of academic literature is included, pointing out the benefits and barriers found in software design, and concluding that the current trend is to include some sort of accessibility measure in new developments.

Sánchez, Darin and Andrade [51] conduct a systematic review of multimodal games which concludes that audio is a very common interface element, but other communication channels remain underused. For example: adaptation of GUI element sizes or the use of colours which are discernible by colour-blind people are uncommon practices in video game design. Besides, most games in this review do not take cognitive impact into account, and this approach would be needed for the development of cognitive skills in players who are blind.

A systematic review by Westin, Bierre, Gramenos et al. [52] covers general progress in the field of game accessibility from 2005 to 2010. Visual impairments prove to be the most researched topic in this context. The authors propose exploring more ways of employing haptics or tactile interfaces as an addition to audio in video games for the blind, which is still a very relevant topic for current research. Another survey about general game accessibility [53] defines visually impaired players as "unable to perceive primary stimuli", and propose modifying essential communication elements as the only way to improve video games on this matter. The most common adaptation techniques observed in reviewed literature are replacing visuals with audio or haptic feedback, and enhancing visuals for players with partial eyesight.

Lastly, Brown and Anderson [54] also evaluate the current state of accessibility in video games, specifically in terms of designing for disability. They evaluate 50 games, chosen using objective criteria, such as total sales, critical reception, awards, etc., in order to examine the widest possible sample of the most prominent games released in 2019. The results highlight design pitfalls and innovations regarding accessibility in four key areas: audio, vision, player movement and difficulty.

IV. Adapting Games for the Visually Impaired

Other authors, however, explore ways to adapt already existent video games in order to make them more accessible, and audio plays a very important role in this process. The subject of how audio can be used to make games accessible is explored by [55]. These researchers add sound to a shooting game with audio aids in order to provide information about direction and distance of moving targets, and then test it with users. The result is that both blind and sighted players can play this kind of game, and that sighted people also improve their performance with the help of audio.

An interactive, audio-based virtual environment called AudioBattleShip is presented in another research article by Sánchez, Baloian, Hassler et al. [56]. It aims to enhance collaboration and cognition in blind learners, and focuses on using spatialized sound to develop collaborative skills in blind learners.

[57] introduces the Racing Auditory Display (RAD), an audio-based user interface that allows players who are blind to play the same kind of racing games that sighted players can play with a similar efficiency and sense of control. Thanks to 2 empirical studies, these researchers find that players preferred the RAD's interface over that of *Mach 1*, a popular blind-accessible racing game.

Another research by [58] results in the game *Audio Space Invaders.* This software focuses on demonstrating how a traditional space invasion game can be made accessible using a 3D audio (Ambisonics) environment. The authors also present a multimodal game where sighted and blind users can share a gameplay experience by combining audio and visual interfaces with force feedback joystick movement.

Rock Band is a commercial video game which combines audio with sensors. The work by [59] presents a modification called *Rock Vibe*, which allows to create visual representations using haptic and audio feedback, thus granting access to this game for visually impaired people. A usability evaluation of this project is also included.

Through two exergames (*Dance Revolution Extreme 2* and *EyeToy Kinetic*), Gasperetti et. al. [60] present a list of modifications to be added in order to make a game accessible to visually impaired users.

A similar approach is taken by [61], who propose a badminton game called *Sonic-Badminton*, in which users can practice this sport by means of a virtual shuttlecock, using audio cues as navigational references. To verify its usability, the authors present a study carried out on both blind and sighted users, and state that all participants were able to play and enjoyed the game.

VI Tennis is another exergame presented by [62]. It is a modified version of a popular motion sensing game that explores the use of vibrotactile and audio cues. This application was tested by 13 blind children and the authors were able to observe how they improved their physical activity.

These same authors present, in another article [63], an adaptation of *VI Bowling*, a haptic and auditory exergame which can be played using a motion detection controller. This software makes use of a novel technique to perform sensory-motor space challenges, which can be applied to motor learning. *VI Bowling* was evaluated by six blind adults, obtaining very positive results.

Another research by Vickers, Istance and Heron [64] aims towards a comprehensive software framework which allows for dynamic adaptation of computer games to different levels of physical and cognitive abilities. The authors describe the principles by which games can be adapted, both during construction and during gameplay, in order to accommodate different abilities and disabilities, all while using the Game Accessibility Development Framework.

Khaliq and Torre [65] also aim for a comprehensive approach, and study how to include accessibility in games for people with visual impairments in a variety of ways. They carry out a study on current technologies, and offer three general adaptation techniques, classified into visual, auditory and tactile. These approaches are justified with the results of experiments conducted on different groups of the visually impaired population.

Olsson [66], on the other hand, proposes a study about the different barriers to accessibility still existing in video games. In order to do so, a heuristic tool aimed towards accessibility evaluation is presented. The tool was validated by applying it to four different games. This research also explores the possibility of incorporating the tool into a game development process. The results constitute a relevant contribution about systematic examination of accessibility features and problems experienced in shooter games by disabled people.

Lastly, visual replacement is explored by [67] in an article that proposes replacing visual stimuli with haptic stimuli as a viable strategy to make *Guitar Hero*, a popular rhythm game, accessible. The authors call this new approach *Blind Hero*, and present a haptic glove which translates visual stimuli into haptic stimuli. They also conduct a study with this modified video game involving sighted and non-sighted players, as well as performance measurements and comparisons in order to decide if players get a similar gameplay experience. The results indicate that the haptic glove can successfully translate visual stimuli into haptic stimuli despite having to compromise some elements of gameplay. Except for these compromises, all participants considered playing *Blind Hero* a fun and enjoyable experience.

V. Video Games as a Learning Tool or as Treatment for People With Visual Impairments

In the reviewed literature, several researchers have spoken about how technology can improve people's lives with amblyopia. As an example, [68] present an experiment to determine whether playing video games can induce plasticity in the visual system of adults with amblyopia. Researchers divided 20 participants into three groups depending on the content they played: an action video game, a non-action video game, and a crossover control group. Results showed that playing video games (both action and non-action games) for a short period using the amblyopic eye results in a substantial improvement in various fundamental visual functions. Finally, they claimed that the recovery in visual acuity is at least five times faster than expected from typical occlusion therapy in childhood amblyopia.

In the same line, Dr. Indu Vedamurthy and his colleagues [69] explore the connection between reduced suppression and improved visual function following treatment in adult amblyopia. The authors gathered twenty-three adults with amblyopia, ten with anisometropia (unequal refractive error), and thirteen with strabismus. They played 40 hours of a dichoptic action video game designed to reduce suppression, promote fusion, and increase attention by the amblyopic eye under binocular conditions. Contrary to the investigation above, the results showed no significant correlation between decreased suppression and improved visual function. This finding challenged the prevailing view and suggested that while dichoptic training improves visual acuity and stereopsis in adult amblyopia, reduced suppression is unlikely to be at the root of visual recovery. The same authors [70] also present the evaluation of a dichoptic action video game especially designed to improve amblyopia in adults. Thirty-eight adults with unilateral amblyopia participated. The experimental intervention was compared to a conventional method of supervised occlusion while watching movies. Results showed that the game group's visual acuity improved, on average, 28 % more than in the movie group.

Some researchers have explored the use of serious games to train visually impaired people in different skills. For example, a short paper by [71] presents some games for learning to use the EyeCane device. EyeCane expands the traditional WhiteCane functionalities with distances and angles. Also, following the same research line, [72] present and test *CPRforBlind*, a video game prototype composed of 13 mini-games. Researchers designed it to introduce the main steps of the cardiopulmonary resuscitation protocol to visually impaired people. The player acts as a helper who has to save the victim by using tactile interaction to solve the challenges assigned to each mini-game. Sixteen participants (8 blind and 8 sighted) tested the game. The authors claim a significant improvement in CPR knowledge after playing the game; besides, blind participants found it fun and easy to play.

A similar approach is taken by [73] who present the design and developing process of *AudioLink*, a role-playing video game for learning scientific concepts and reasoning only through audio. Besides, they evaluated its usability and cognitive impact with 20 students (13 of them with low vision). Results indicated that users considered the software appealing, challenging, engaging, and encouraging as a science learning tool. Besides, subjects rated the usability of *AudioLink* highly.

Other authors [74] opt for a more analytical perspective, and propose applying a combined method for evaluating accessibility in educational video games, taking into account the Web Content Accessibility Guidelines (WCAG) 2.1. The authors evaluated 82 video games and reached the conclusion that no serious games among those selected have reached an acceptable level of accessibility. Therefore, serious game developers should make significant efforts to improve accessibility in the future.

A systematic literature review by these same authors [75] highlights current trends and prominent issues in accessibility and provides guidelines for designing serious inclusive games. The study is based on the results of existing studies published between 2000 and 2020. The authors extract a total of 476 studies, and after a screening process with the help of the PRISMA flowchart, focus on only 47 of them. The authors conclude that developers rely on assistive technologies (both software and hardware) to achieve greater accessibility in serious games. Besides, they highlight the need for research on accessibility policies, guidelines, and practices for serious games, and reflect on the threats of not taking these measures into account.

VI. Navigation and Interaction of Visually Impaired Players in Virtual Environments

Virtual environments present numerous issues to visually impaired or blind people. That may well be the reason why this subject has attracted so much attention among the scientific community researching accessibility. We will start delving into this subject by summarising some studies on different navigational strategies, used either in virtual environments or in the real world.

Teaching navigational strategies to people with vision issues has caught the attention of the scientific community. This fact should not be surprising since there are medical studies [76] that find neural correlations while blind subjects navigate virtual environments, or experiments [77] to determine the neural mechanisms underlying spatial navigation in congenitally blind individuals. The act of comparing between sighted and non-sighted people has been a recurrent theme in medical studies, and is also present in studies involving specific technology. This is the case of [78], who introduced the concept of path integration. Path integration is form of navigation in which perceived self-motion is integrated over time to obtain an estimate of current position and orientation. In their study, non-sighted subjects, who were passively guided over the outbound path, made significant mistakes when attempting to return to the original position, but were nevertheless sensitive to turns and segment lengths in the stimulus path. Previous research by these authors had shown that optic flow constitutes a weak input to the path integration process.

Some researches focus on creating technologies that help people perceive objects around them. For example, VizWiz::LocateIt [79] is a mobile system that enables blind people to locate objects in their environment by combining remote human computation and local automatic computer vision. The authors claim this application to be ground-breaking when solving many useful problems for blind people by changing the paradigm of search problems. It is inspired by how blind people overcome many accessibility issues today by asking a sighted person: the application first asks people to outline the object, and then enables efficient and accurate automatic computer vision to guide users in an interactive manner. Only a commercial smartphone is needed for this to work. Finally, the authors present a two-stage algorithm which aims to guide users to the object they were looking for.

Virtual environments are remarkable tools when it comes to teaching and developing orientation. Merabet and Sánchez [80] describe a series of interactive audio-based computer software and virtual environments designed to serve as rehabilitative approaches to improve spatial navigation, problem-solving skills, and overall confidence in individuals with visual impairments. The authors find substantial differences in behavioral gains obtained through virtual navigation, especially when compared to navigating the real world. Even so, they advocate for a mixed strategy, instead of replacing current rehabilitative techniques. Finally, they highlight the importance of understanding how the brain creates spatial cognitive maps as a function of learning modality and over time in order to improve rehabilitation techniques. Dr. Merabet et al. also created the Audio-based Environment Simulator (AbES) [81], a novel approach to train navigation and spatial cognition skills in adolescents who are blind within a virtual reality application. In this paper, two experiments developed with said tool are presented, involving seventeen early blind persons. The authors conclude that the application significantly encouraged blind users to explore the virtual environment actively. Besides, it generated a definite sense of large-scale three-dimensional space and facilitated learning and transferring navigational skills to the physical world.

Subsequently, [82] present an experiment with AbES in which the authors measured the ability and efficacy of adolescents with early-onset blindness when acquiring spatial information gained from the exploration of a virtual indoor environment. Success in the transfer of navigation skill performance was markedly high, suggesting that interacting with AbES leads to the generation of an accurate spatial mental representation. Furthermore, there was a positive correlation between success in gameplay and navigation task performance. Finally, these same authors [83] delve further into the AbES as a way to explore the layout of an unfamiliar, complex indoor environment. The researchers carried out an experiment during which they compared two modes of interaction with AbES. In one group, blind participants implicitly learned the layout of a target environment while playing an exploratory, goal-directed video game. A second group was explicitly taught the same layout following a standard route and instructions provided by a sighted facilitator. Additionally, a control group interacted with AbES while playing the same kind of game. Results showed that participants from both modes of interaction were able to transfer spatial knowledge gained during gameplay. This transfer was not present in the control group.

In a previous study, [84] used a virtual environment to achieve a twofold objective: discovering the structural components and relationships that participants included in their cognitive map, and knowing how the constructed cognitive map contributes to orientation when navigating a limited space in the real-world. The author focused on the ability to recall and perform orientation tasks using long-term memory, and included a long-term experiment involving 4 participants (1 congenitally and 3 late blind). A point to highlight of the study is that participants were able to recall general properties of a virtual room which they visited twice, 16 months apart.

When dealing with visually impaired users, usability and accessibility while navigating are key factors to decide whether an application is worth it or not. In this regard, Jaime Sánchez has led many studies. The first one [85] presented a usability evaluation study of a haptic device designed *ad hoc* (Digital Clock Carpet), and a 3D video game (*MOVA3D*), which is based on audio in order to allow for the development of orientation and mobility skills in children when walking through closed and unfamiliar spaces. The results of this study showed that both the haptic device and the videogame were usable. The authors also described in detail the evaluation process, the re-design of some parts of the tool, and the final end-user usability evaluation. In the second one, [86] presented a design, development, and usability evaluation for the already-mentioned Audio-Based Environments Simulator (AbES) software. The environment this software creates, in this case, is a virtual representation of a real space, developed using user-centered design methodologies. As in the first study, mentioned above, results showed that users feel satisfied with the proposed interface. Researchers claimed that they are also using these simulators to study brain changes and adaptations by incorporating AbES within a neuroimaging environment.

In a previous study, the same author [87] presented, together with other colleagues, the design and usability evaluation of three-dimensional interactive environments for children with visual disabilities: *AudioChile* and *AudioVida.* 3D audio aims to orient players, avoid obstacles, and identify the position of characters and objects within the environment. The authors also described the process of testing with children with residual vision and improving their graphical user interfaces. Finally, they concluded that the use of this software allows children to differentiate and identify surrounding sounds that helped them to orient themselves. Finally, Sánchez [88] also researched the problems encountered when implementing awareness in collaborative software for blind people. The study had three phases: the design of a collaborative application for learning musical instruments; the usability testing of this software with five learners who evaluate the interfaces for both interaction and collaboration; and the identification of some unique problems in implementing awareness in collaborative software for blind learners. The most important conclusion was that awareness mechanisms for people with disabilities imply more complex problems that must be considered when designing collaborative software.

Other studies present experiments designed to determine the efficiency of virtual environments to help in navigation. [89] carried out an experiment to determine the efficiency of auditory and visual cues to find gates in a game-like experience. The authors conclude that audio-visual navigation was the most efficient, followed by visual navigation, and audio-only navigation revealed itself as the least effective method. The importance of auditory stimuli in virtual reality applications is, however, highlighted, as in this scenario many essential features of the virtual environment are located outside the main field of view.

Besides, Lahav and Mioduser [90] presented and experiment, involving 31 blind participants, to prove the effectiveness of a virtual environment as a helper for visually impaired people who navigate the real world. By using haptic and audio feedback in order to ease the task of exploring unknown spaces, they examined the ability of all participants to apply cognitive maps created in a virtual environment when accomplishing tasks in the real world. The results attained suggest that working within the virtual environment gave participants a stimulating, comprehensive, and thorough acquaintance with the target space. Evidence of a transfer from the exploration within the virtual environment to the cognitive map built in subjects' minds is also provided. In the same vein, Wersényi [91] used a virtual environment to help the blind community to use personal computers. The author focused on technical and hearing-related questions, as well as human factors. He conducted an experiment to compare the performance of twenty eight blind users to 40 sighted subjects in a virtual auditory environment.

Auditory environments have also been tested as helpers to visually impaired people. [78] collected a series of studies on auditory distance perception in outdoor environments. These show a systematic underestimation of sound source distance. As a conclusion, the authors describe a navigation system for the visually impaired that uses three recent technologies: the Global Positioning System (GPS), Geographic Information Systems (GIS), and virtual acoustics. They also state that these three technologies show a potential to help visually impaired individuals to navigate and learn about unfamiliar environments without the assistance of human helpers.

In order to evaluate three different auditory environments, other authors [92] presented an experiment, undertaken by 8 visually impaired users (a similar, previous study with 25 sighted participants is also described in this work). The main conclusion is that sound can help both sighted and visually impaired users in 3D applications (sighted users prefer music, while visually impaired users prefer simple tones). Also, they argue that, while sound-only applications are possible for both groups, visually impaired users are more proficient while using them.

In auditory environments, there are different sonification techniques. The most known include: auditory icons (brief sounds representing objects, functions, and actions), earcons (abstract, synthetic and mostly musical tones or sound patterns that can be used in structured combinations) and speech (a voice announcing the name of the object). On the other hand, other newer techniques include a combination of the ones described above, such as spearcons (spoken phrases sped up until they may no longer be recognized as speech) and hybrids (a combination of different sound types). Dingler et al. [93] present an experiment to determine the degree of learnability for different sonification techniques aiming to represent standard environmental features. Participants were 39 undergraduate students who reported normal or corrected-to-normal hearing and vision. Authors conclude that spearcons are comparable to speech, at least regarding learnability. Also, they leave the speech channel open and are very brief, which reinforces their potential as a sonification methodology.

Following the same line, some authors [94] use spatialised sound to help blind learners to construct cognitive spatial structures. This short paper presents the design process of an interactive software. The authors also claim to have conducted experimental studies in two different Chilean schools for blind children, which included thirty learners, for two years. They researched exposure to acoustic environments, corporal exercises, and experiences with sand, clay, styrofoam, and Lego bricks. The results of these studies revealed that blind children can build mental structures only with sound and that spatial imagery is not purely visual by nature, but can be constructed and transferred through spatialised audio.

A very different and original approach is taken by [95], who present a computer application, the Audio Abacus, designed to help users navigate sequences of numbers by transforming them into tones, following the analogy of an abacus. At the time of the paper, authors declared to be in the middle of an ongoing process, during which several parameters of the application will be evaluated. Thus, they presented early results from an initial experiment, which showed that users were able to perform relatively well with this system. However, we want to point out that no later related research has been found.

It seems clear that sounds represent a basic tool to help visually impaired users to navigate virtual or real environments. A study by [96] delves into automatically generating impact sounds, based on a physical modelling method. The kind of modelling presented in this article can facilitate the implementation of non-speech auditory cues representing objects and interactions in non-visual interfaces. Also, the authors discuss the results of comparing recorded impact sounds with their abstract representations. Finally, they claim to be able to automatically generate sounds for "spherical objects falling onto a beam or plate" with fast algorithms that implement the described physical models. In spite of the promising results, no more recent work on the subject has been found.

Video games have also proven useful when comparing navigational skills between sighted and non-sighted people. [97] present a study to compare the navigational strategies that sighted and blind subjects tend to use while playing. This work is based on studies on navigation that show spatial mental models are encoded differently across both groups of players. Whereas sighted players experience higher immersion in the allocentric system (that is: they tend to view the environment as a general representation), blind players experience higher immersion in the egocentric system. The experiment was passed to 6 blind and 6 sighted players, and used a text-based video game. Results support the hypothesis that egocentric directions are more natural to use and more immersive for non-sighted players, while sighted players perform better with allocentric directions.

This whole section has made evident that audio is the most common communication channel when creating accessible navigation systems or interfaces for the blind and visually impaired. Due to the amount of work in this area, in 2015, [98] conducted a systematic review of navigational audio-based games. The authors took conceptual and technological approaches, and also gave special attention to accessibility, as well as to user interaction efficiency. A series of audio games and virtual environments are presented; they are designed to help visually impaired people to improve their ability while performing exploration and orientation tasks. They are also aimed towards helping to develop spatial navigation and problem-solving skills. Two important conclusions were reached: on one hand, audio games represent an effective means toward progression in spatial contextual learning for visually impaired people; on the other, the challenging nature of audio games promotes a high level of engagement, immersion, and active user interaction for both sighted and visually impaired people.

Additionally, it is possible to find different examples that show the benefits of audio games when addressing navigational problems. Recently, [99] presented an audio game designed to teach blind children navigation skills. It consists of a 3D exploration game, which uses a head-mounted display and a realistic soundscape to create an immersive experience. Besides, a qualitative experiment was performed in a small group of blind teenagers. The authors informed that their reactions were quite positive, making the experience engaging and fun. Finally, the authors agree on the advantages of incorporating this game into educative programs for blind people.

A different short paper [100] presented an improvement for an already existent application (it is worth mentioning that no further information on that application was found) : NAVI (Navigation Aid for Visually Impaired). The authors created this system in 2001 and improved the tool through the process that is described in this paper.

The improvements made in NAVI are discussed, and a preliminary experimental approach is also described. This experiment focuses on developing a navigational aid for the blind through the transformation of stereo image to stereo sound.

Sound also presents advantages when interacting with mobile phones. In 2008, [101] published a study about designing non-visual games for phones. An example called *The Audio Flashlight* was provided. The interface of this video game is based on audio, vibrations, and gestural input. The authors tried to avoid auditory overloading by representing collisions through touch. A preliminary evaluation indicated that players enjoyed the experience, as it seemed fun and unusual to them, especially when compared to traditional games. These authors also include an interesting discussion on the difficulties that blind people have when playing games of this type. [102] also present a study about developing an application for mobile devices, especially aimed towards people with special needs such as the blind and visually impaired.

Command-based interfaces for virtual worlds can also ease the access of visually impaired users to video games. This is the case of TextSL, a client for *Second Life* that can be accessed with a screen reader. The authors [103] conducted an experiment involving 16 participants (8 were using a screen reader and 8 were sighted) to prove its effectiveness. They concluded that a command-based interface is a feasible approach to explore *Second Life*, communicate with other avatars, and interact with objects. However, command-based exploration and object interaction are significantly slower in TextSL. Two significant problems were identified in the conclusions: on the one hand, objects in virtual worlds such as *Second Life* often lack metadata, which makes it challenging to provide an accurate textual description; on the other, virtual worlds have large numbers of objects that may easily overwhelm a user when turned into audio feedback.

While command-based interfaces seem to be an interesting way to improve accessibility, there are studies [104] on neuroscience that propose the use of technology to expand human perception and to gain a deeper understanding of brain plasticity and cognitive processes. The authors of this research review current tools for different kinds of sensory substitution. Finally, they conclude by pointing out the need for robust and cheap technology for patients suffering from any kind of sensory loss. This study supports further research on the usefulness of tactile (haptic) devices. These devices have raised a lot of expectations amongst the scientific community due to their potential for non-sighted users.

[105] make use of a haptic device in order to develop a technique called real-time sensory substitution (RTSS). It allows players who are blind to play gesture-based video games without having to make any modifications to the game itself. RTSS uses real-time video analysis to detect the presence of a particular visual cue, which is then substituted with a vibrotactile cue that is provided by an external controller. The authors also detailed a study taken by 28 sighted subjects playing *Kinect Hurdles*. Results found no significant difference in performance (taking into account accuracy and completion time) when interchanging visual and haptic cues in versions of the game which already make use of a semantic sound system.

Haptic technologies are also applicable to traditional games. A good example is the implementation of an accessible Sudoku by [106]. The authors expose the benefits of using a planar, touch-sensitive and refreshable haptic display and compare it with other means of interaction with blind people.

By combining haptic and audio devices, [107] present and test the GRAB project. It included a haptics and audio system provided with a set of utilities and applications which granted visually impaired people access to 3D graphics. The project included audio synthesis, speech recognition, and haptic interaction to allow for object exploration and manipulation. To validate this system, the authors tested it with an adventure game, a city map explorer, and a chart data explorer. Participants in this validation process had different degrees of blindness. Authors claimed that the results confirmed the validity and potential of the GRAB system for these applications, although some features required some adjustments in order to create usable tools.

Haptic devices have also been used for other purposes. Johnson and Higginns [108] presented the building process and preliminary validation of a compact, wearable navigation device which aids visually impaired users to navigate everyday environments in real-time. Off-the-shelf components are used to create a device capable of gathering visual inputs (through webcams), extracting relevant information about the environment (where the nearest objects are in three-dimensional space), and finally translating it into a tactile signal. Authors claim that preliminary results suggest that this device is useful for object avoidance in simple environments.

[109] compiled some recommendations and guidelines for using haptic devices. These rules are grouped under the categories of “navigation”, “finding objects”, “understanding objects” and “haptic widgets”. The author claims that these recommendations help to improve haptic program efficiency. Other authors [110] have researched potential applications of haptic feedback in virtual worlds. In order to reach that goal, they proposed a haptic-enabled version of the *Second Life* game client (the same tool as [103]), which includes two haptic-based input modes to help visually impaired people while navigating and exploring the virtual 3D environments contained in the game. Additionally, the results of an experiment with temporarily blind-folded users show that haptic-based input modalities allow to easily find and reach avatars in virtual worlds.

A different experiment conducted by Chebat et al. [111] drew bolder conclusions. They tried to determine the ability of congenitally blind participants to detect and avoid obstacles using a sensory substitution device: the tongue display unit. 16 congenitally blind and 11 sighted subjects participated in the study. Results showed that blind participants outperformed sighted ones when detecting and avoiding obstacles using the substitution device. Besides, they demonstrated that bigger obstacles were better detected than smaller ones, and that step-around obstacles were more avoidable than step-over ones. Finally, the authors claimed to have produced the first study to show that congenitally blind participants outperform sighted ones in spatial tasks when equipped with a sensory substitution device.

Finally, [112] present a short paper including an experiment where 6 visually-impaired students evaluate a website. 4 of them were blind, and 2 had low vision. The conclusion they reach is that TAP (Think Aloud Protocol), in its simplified form as a concurrent verbal protocol method, may not be effective when applied to scenarios where blind people use a screen reader to access websites. Besides, they observed that blind users did not offer as many comments as users with low vision, even when prompted regularly. Finally, they claim that the TAP protocol should be modified in order to better adapt it to blind users.

VII. Discussion

Analysing thematic variations of published research in accessibility and video games provides essential information on how the area of knowledge has evolved over the last 27 years. Fig. 3 illustrates the evolution in the number of published works (classified by theme) over the period considered.

Navigation and interaction works were the pioneers (the first work appears in 1994) and have dominated the field until 2015. The design and architecture of video games (either by describing final products or

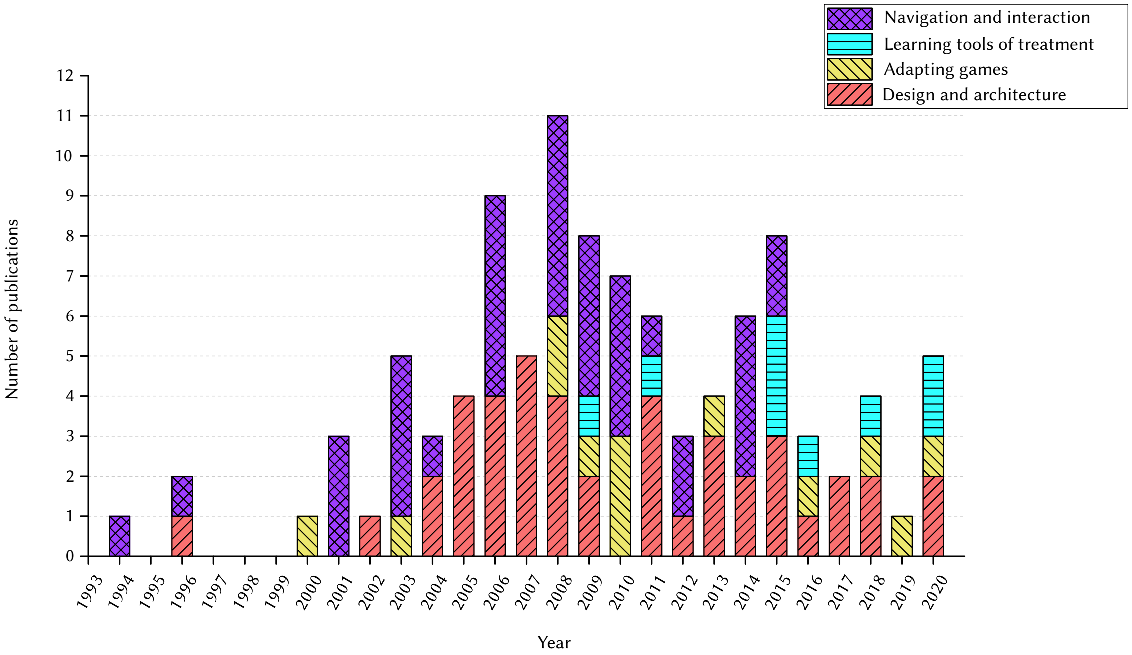


Fig. 3. Bibliography distribution over time. Bars are stacked and show the total amount of articles in this field for every year, classified by theme.

software development supporting accessible game design tasks) began to attract the researchers' interests in 2004 (the first work related to this subject was undertaken in 1996) and remains stable until today. However, academic papers that conceive video games as learning tools or treatments for the visually impaired have become far more frequent in the latest years, particularly since 2014. It seems reasonable to expect a consolidation of this tendency in the near future, as well as further growth in this field.

As for adapting games to make them more accessible, the graph shows a scarce repercussion of this subject throughout the years, and the amount of publications shows a trend to diminish. During this research process, we noticed that most authors agree that adapting video games to make them accessible is inconvenient; on the contrary, accessibility must be tackled since the beginning of the design process.

We have also detected an increment in the total number of publications on accessible entertainment since 2000. This field, however, achieved peak popularity in 2008 and, since then, the number of relevant academic papers has been reduced. The relatively low amount of recent articles found during the selection process confirms that there is still room for growth in this area.

In order to determine the trend followed by analysed papers, we included Fig. 4. It gathers the most popular keywords (catalogued and normalised to avoid repetitions) throughout the reviewed literature. Similar keywords are grouped under the same global name. For example: "video game", "video games", "games" and other similar concepts were all unified as a single term: "video-games". Additionally, keywords with only one appearance were discarded from this list to keep it relevant, as they do not allow to establish any relationship with other articles.

The first four terms in this list ("video-games", "blindness", "visual-impairment" and "accessibility") define the subject of this systematic review and are related to complementary concepts, such as "virtual-environments", "virtual-reality", "interaction", "3D", "game-engine", etc. "Sound" is the first non-thematic keyword in this list. It shows a clear trend in the field: sound is the preferred channel when implementing sensory substitution systems. Besides, it is a fundamental piece for the design and development of accessible games and interactive applications. In fact, "sound" becomes more important if we consider its tight relationship with other tangential concepts such as "audio-games", "auditory-interfaces", "spatialisation", "sound-generation" or "music".

"Navigation" also stands out as an essential concept; many articles reference this term in various ways. The most common approaches used to spatial navigation are related to either orienting players in virtual environments or using them to guide the real world. "User-interface" and "human-computer-interaction" are also strictly related concepts in this field; most authors aim to create accessible interactive applications by easing the interfaces interactions. Lastly, it is also common to find articles where sensors help to make video games accessible. The "haptic" and "touch" keywords illustrate that category.

The rest of the keywords do not make enough appearances to be considered relevant enough for this analysis, and most of them act as supplementary concepts for the already-mentioned terms.

VIII. Conclusions

This paper reviews the most relevant works published in the last 27 years. The lack of very recent articles in our review can be explained by the research methods utilised —recent works have not accumulated enough citations to be well-positioned in popular academic search engines. As was already mentioned, this field has recently experienced a significant evolution, but numerous issues still need improvement. One of the main concerns amongst consulted authors is the lack of mainstream content. High-budget (AAA) titles do not usually take any accessibility measures beyond the inclusion of closed captions and are often unplayable for blind users. Additionally, most accessible products and technologies, like the ones mentioned in Section 3, are tightly related to academic environments, which reduces their social impact due to their inability to reach the general public.

After analysing this group of research articles, we detected a certain stagnation in human-computer interaction for the blind and visually impaired. Authors agree that there is a need to improve how non-visual interfaces work and develop current technologies further to adapt better to every scenario. According to the reviewed literature, the best way to improve accessibility would be to combine haptics and audio technologies to create better and more intuitive user interfaces.

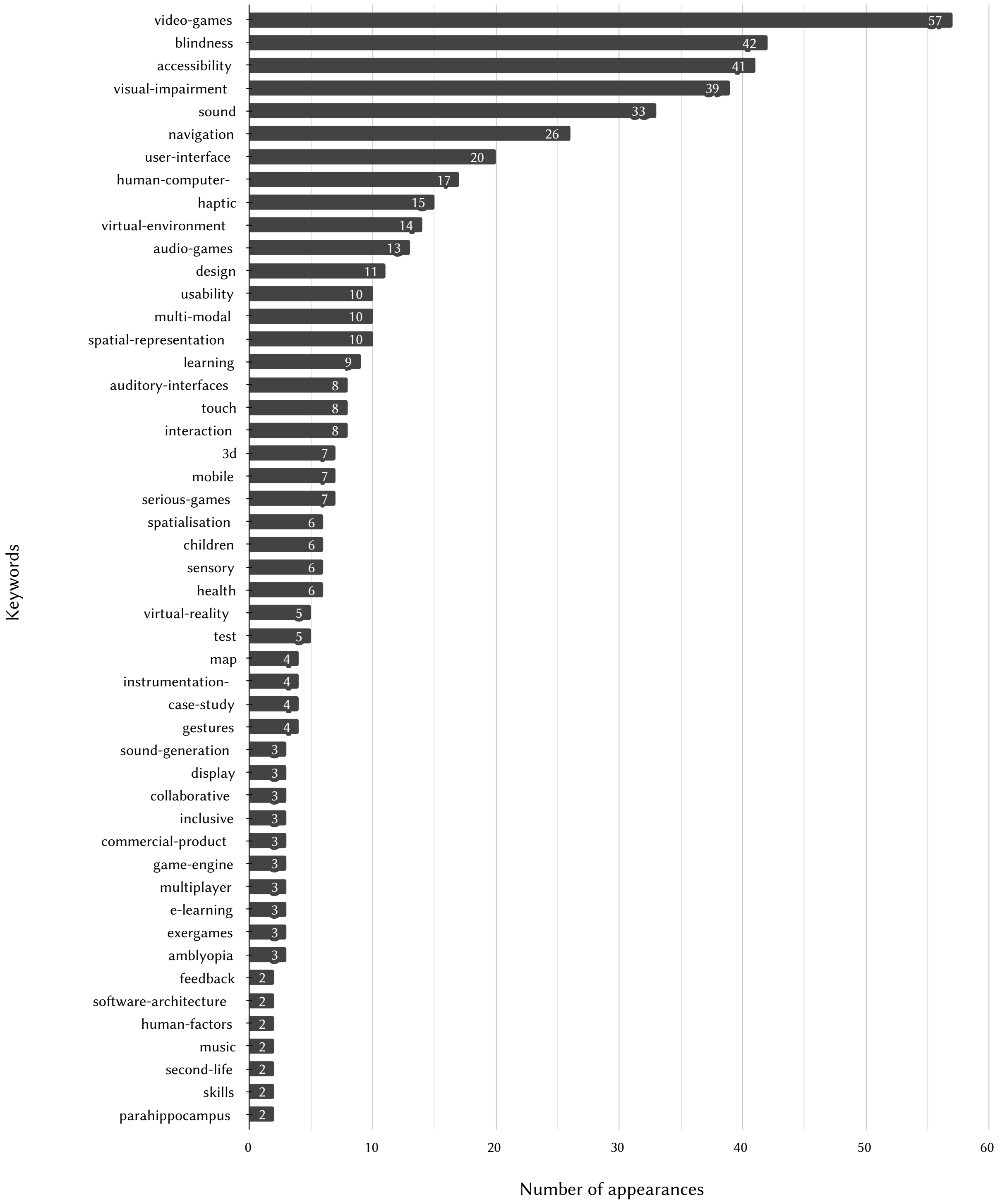


Fig. 4. Keyword distribution in reviewed literature.

This study also shows that video game development should start paying more attention to the production of accessible experiences. There is an increasing social demand, as well as a well established academic interest in this area. Still, it is also the responsibility of game creators to develop easy and intuitive ways to adapt to every manner of perceiving the world around us.

Lastly, many authors agree that there is still a long path ahead. While great companies perceive accessibility as social work (corporate social responsibility) more than a business itself, we will have two video games styles. The expensive, interesting, spectacular, and beautiful ones for the general public; and the old-fashioned, mostly dull and cheap, made to be accessible. Moreover, since accessibility must be taken into account from scratch, these two paths tend to fence off.

From our point of view, there is still much to be done before accessible video games go through academia and reach the real world, and in particular, successful, AAA projects. But a review like this one makes one thing clear: a high social demand exists, and interest in this field is growing, even among the most successful game studios. It is the role of academics to provide the world with a solid enough foundation which will allow for the development of significant games that will forever change how our societies understand computer entertainment.

Acknowledgments

This article was partially funded by a Complutense University grant (CT27/16-CT28/16) for postdoctoral research, in collaboration with Santander Bank. It has also received funding by the Ministry of Science, Innovation and Universities of Spain (Didascalias, RTI2018-096401-A-I00). Additionally, this research is carried out in collaboration with the ONCE-Tiflotechnology Chair of the Complutense University of Madrid.

References

[1] B. Leporini, F. Paternó, "Increasing usability when interacting through screen readers," *Universal Access in the Information Society*, vol. 3, no. 1, pp. 57–70, 2004, doi: 10.1007/s10209-003-0076-4.

[2] A. P. Freire, R. P. de Mattos Fortes, D. M. Barroso Paiva, M. A. Santos Turine, "Using screen readers to reinforce web accessibility education," *ACM SIGCSE Bulletin*, vol. 39, p. 82, jun 2007, doi: 10.1145/1269900.1268810.

[3] J. Lazar, A. Allen, J. Kleinman, C. Malarkey, "What frustrates screen reader users on the Web: A study of 100 blind users," *International Journal of Human-Computer Interaction*, vol. 22, no. 3, pp. 247–269, 2007, doi: 10.1080/10447310709336964.

[4] Y. Borodin, J. P. Bigham, G. Dausch, I. V. Ramakrishnan, "More than meets the eye: A survey of screen-reader browsing strategies," in *Proceedings of the International Cross Disciplinary Conference on Web Accessibility (W4A 2010)*, Raleigh, 2010, pp. 1–10.

[5] A. Bhalerao, S. Bhilare, A. Bondade, M. Shingade, "Smart Voice Assistant: A universal voice control solution for non-visual access to the Android operating system," *International Research Journal of Engineering and Technology(IRJET)*, vol. 4, no. 1, pp. 1713–1720, 2017.

[6] P. Bose, A. Malpthak, U. Bansal, A. Harsola, "Digital assistant for the blind," in *Proceedings of the 2nd International Conference for Convergence in Technology (I2CT 2017)*, vol. 2017, Mumbai, 2017, pp. 1250–1253.

[7] S. M. Felix, S. Kumar, A. Veeramuthu, "A Smart Personal AI Assistant for Visually Impaired People," in *Proceedings of the 2nd International Conference on Trends in Electronics and Informatics (ICOEI 2018)*, Tirunelveli, 2018, pp. 1245–1250.

[8] A. Abdolrahmani, R. Kuber, S. M. Branham, "Siri talks at you: An empirical investigation of voice-activated personal assistant (VAPA) usage by individuals who are blind," in *Proceedings of the 20th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS 2018)*, Galway, oct 2018, pp. 249–258, Association for Computing Machinery, Inc.

[9] S. M. Branham, A. R. M. Roy, "Reading between the guidelines: How commercial voice assistant guidelines hinder accessibility for blind users," in *Proceedings of the 21st International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS 2019)*, Pittsburgh, oct 2019, pp. 446–458, Association for Computing Machinery, Inc.

[10] N. Druckmann, A. Newman, K. Margenau, E. Schatz, R. Cambier, T. McIntosh, C. Gyrling, E. Pangilinan, J. Sweeney, C. Nakata, H. Gross, G. Santaolalla, "The Last of Us: Part II," 2020.

[11] D. Archambault, D. Olivier, "How to make games for visually impaired children," in *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology*, vol. 265, Valencia, 2005, pp. 450–453.

[12] D. Archambault, R. Ossmann, T. Gaudy, K. Miesenberger, "Computer Games and Visually Impaired People," *Upgrade*, vol. 8, no. 2, pp. 43–53, 2007.

[13] D. Archambault, T. Gaudy, K. Miesenberger, S. Natkin, R. Ossmann, "Towards generalised accessibility of computer games," in *Proceedings of the International Conference on Technologies for E-Learning and Digital Entertainment 2008*, vol. 5093, Nanjing, 2008, pp. 518–527, Springer.

[14] A. Buaud, H. Svensson, D. Archambault, D. Burger, "Multimedia games for visually impaired children," in *Proceedings of the International Conference on Computers for Handicapped Persons*, vol. 2398, 2002, pp. 173–180, Springer Verlag.

[15] T. Roden, I. Parberry, "Designing a narrative-based audio only 3D game engine," in *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology*, vol. 265, Valencia, 2005, pp. 274–277, ACM.

[16] M. T. Atkinson, S. Gucukoglu, C. H. Machin, A. E. Lawrence, "Making the mainstream accessible: Redefining the game," in *Proceedings of the Sandbox Symposium 2006: ACM SIGGRAPH Video Game Symposium (Sandbox 2006)*, Boston, 2006, pp. 21–28, Springer.

[17] S. Reardon, "Playing by ear," *Science*, vol. 333, no. 6051, pp. 1816–1818, 2011, doi: 10.1126/science.333.6051.1816.

[18] J. Carvalho, T. Guerreiro, L. Duarte, L. Carriço, "Audio-Based Puzzle Gaming for Blind People," in *Proceedings of the Mobile Accessibility Workshop at MobileHCI*, 2012.

[19] K. Bierre, J. Chetwynd, B. Ellis, D. M. Hinn, S. Ludi, T. Westin, "Game Not Over : Accessibility Issues in Video Games," in *Proceedings of the 3rd International Conference on Universal Access in Human-Computer Interaction*, 2005, pp. 22–27.

[20] E. Velleman, R. Van Toi, S. Huiberts, H. Verwey, "3D shooting games, multimodal games, sound games and more working examples of the future of games for the blind," *International Conference on Computers for Handicapped Persons*, vol. 3118, pp. 257–263, 2004, doi: 10.1007/978-3-540-27817-7_39.

[21] G. R. White, G. Fitzpatrick, G. McAllister, "Toward accessible 3D virtual environments for the blind and visually impaired," in *Proceedings of the 3rd International Conference on Digital Interactive Media in Entertainment and Arts (DIMEA 2008)*, 2008, pp. 134–141.

[22] J. Chakraborty, S. Chakraborty, J. Dehlinger, J. Hritz, "Designing video games for the blind: results of an empirical study," *Universal Access in the Information Society*, vol. 16, pp. 809–818, aug 2017, doi: 10.1007/s10209-016-0510-z.

[23] A. Sepchat, N. Monmarché, M. Slimane, D. Archambault, "Semi automatic generator of tactile video games for visually impaired children," in *Proceedings of the 2006 International Conference on Computers for Handicapped Persons*, vol. 4061 LNCS, Linz, 2006, pp. 372–379, Springer.

[24] J. F. Cheiran, L. Nedel, M. S. Pimenta, "Inclusive games: A multimodal experience for blind players," in *Proceedings of the Brazilian Symposium on Games and Digital Entertainment (SBGAMES)*, 2011, pp. 164–172.

[25] K. Drossos, N. Zormpas, A. Floros, G. Giannakopoulos, "Accessible games for blind children, empowered by binaural sound," in *Proceedings of the 8th ACM International Conference on Pervasive Technologies Related to Assistive Environments, (PETRA 2015)*, Corfu, 2015.

[26] T. Gaudy, S. Natkin, D. Archambault, "Pyvox 2: An audio game accessible to visually impaired people playable without visual nor verbal instructions," *Transactions on Edutainment*, vol. 5660, no. 2, pp. 176–186, 2009, doi: 10.1007/978-3-642-03270-7_12.

[27] E. P. Flores-Garzón, L. J. Intriago-Echeverría, A. Jaramillo-Alcázar, S. Criollo-C, S. Luján-Mora, "Catch the thief: An approach to an accessible video game with unity," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 10, no. 3, pp. 905–913, 2020, doi: 10.18517/ijaseit.10.3.10938.

[28] G. Giannakopoulos, N. A. Tatlas, V. Giannakopoulos, A. Floros, P. Katsoulis, "Accessible electronic games for blind children and young people," *British Journal of Educational Technology*, vol. 49, pp. 608–619, jul 2018, doi: 10.1111/bjet.12628.

[29] E. Glinert, L. Wyse, "AudiOdyssey: An accessible video game for both sighted and non-sighted gamers," in *Proceedings of the Conference on Future Play (Future Play '07)*, Toronto, 2007, pp. 251–252, ACM.

[30] D. Grammenos, A. Savidis, Y. Georgalis, C. Stephanidis, "Access invaders: Developing a universally accessible action game," in *Proceedings of the International Conference on Computers for Handicapped Persons*, vol. 4061 LNCS, Linz, 2006, pp. 388–395, Springer Verlag.

[31] J. Sánchez, T. Hassler, "AudioMUD: A multiuser virtual environment for blind people," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 15, no. 1, pp. 16–22, 2007, doi: 10.1109/TNSRE.2007.891404.

[32] D. Grammenos, A. Savidis, C. Stephanidis, "UA-Chess: A Universally Accessible Board Game Human-Computer Interaction View project LECTOR: Attention aware Intelligent Classroom View project," in *Proceedings of the 3rd International Conference on Universal Access in Human-Computer Interaction*, Las Vegas, 2005, pp. 1–10.

[33] L. R. Milne, C. L. Bennett, R. E. Ladner, "VBGhost: A Braille-based educational smartphone game for children," in *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS 2013)*, New York, 2013, pp. 75–76.

[34] L. R. Milne, C. L. Bennett, S. Azenkot, R. E. Ladner, "BraillePlay: Educational smartphone games for blind children," in *Proceedings of the 16th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS'14)*, Rochester, 2014, pp. 137–144, ACM.

[35] J. Torrente, Á. Del Blanco, P. Moreno-Ger, I. Martínez-Ortiz, B. Fernández-Manjón, "Implementing accessibility in educational videogames with <e-Adventure>," in *Proceedings of the 1st ACM International Workshop on Multimedia Technologies for Distance Learning, (MTDL 2009)*, Beijing, 2009, pp. 57–66.

[36] A. Sepchat, S. Descarpentries, N. Monmarché, M. Slimane, "MP3 players and audio games: An alternative to portable video games console for visually impaired players," in *Proceedings of the 2008 International Conference on Computers for Handicapped Persons*, vol. 5105 LNCS, Linz, 2008, pp. 553–560, Springer.

[37] J. McElligott, L. Van Leeuwen, "Designing sound tools and toys for blind and visually impaired children," in *Proceedings of the Conference on Interaction Design and Children (IDC 2004)*, jun 2004, pp. 65–72, Association for Computing Machinery, Inc.

[38] S. Nesteriuk, "Audiogames: Accessibility and Inclusion in Digital Entertainment," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10917 LNCS, 2018, pp. 338–352, Springer Verlag.

[39] T. Darin, "Towards a Methodology to Evaluate Multimodal Games for Cognition in People who are Blind," in *Proceedings of the International Conference on Human-Computer Interaction*, Bamberg, 2015, pp. 61–65, University of Bamberg Press.

[40] T. G. Darin, R. M. Andrade, L. B. Merabet, J. H. Sánchez, "Investigating the mode in multimodal video games: Usability issues for learners who are blind," in *Proceedings of the Conference on Human Factors in Computing Systems*, vol. Part F1276, Denver, 2017, pp. 2487–2495.

[41] D. Grammenos, A. Savidis, C. Stephanidis, "Unified design of universally accessible games," in *Proceedings of the International Conference on Universal Access in Human-Computer Interaction*, vol. 4556 LNCS, Beijing, 2007, pp. 607–616.

[42] M. J. Scott, G. Ghinea, "Promoting Game Accessibility: Experiencing an Induction on Inclusive Design Practice at the Global Games Jam," in *Proceedings of the Conference on the Foundations of Digital Games, Inaugural Workshop on the Global Games Jam (GGJ 2013)*, Chania, may 2013.

[43] B. Leporini, M. Hersh, "Games for the rehabilitation of disabled people," in *Proceedings of the 4th Workshop on ICTs for improving Patients Rehabilitation Research Techniques*, 2016, pp. 109–112.

[44] L. J. Lieberman, "Adapting Games, Sports and Recreation for Children and Adults who are Deaf-Blind," 1996. [Online]. Available: https://digitalcommons.brock-port.edu/pes_facpub/111.

[45] C. Mangiron, X. Zhang, "Game Accessibility for the Blind: Current Overview and the Potential Application of Audio Description as the Way Forward," in *Researching Audio Description*, 2016, pp. 75–95, doi: 10.1057/978-1-137-56917-2_5.

[46] D. Miller, A. Parecki, S. A. Douglas, "Finger dance: A sound game for blind people," in *Proceedings of the Ninth International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS'07)*, 2007, pp. 253–254.

[47] M. A. Oren, C. Harding, T. Bonebright, "Evaluation of spatial abilities within a 2D auditory platform game," in *Proceedings of the 10th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS'08)*, Halifax, 2008, pp. 235–236.

[48] J. R. Porter, "Understanding and addressing real-world accessibility issues in mainstream video games," *ACM SIGACCESS Accessibility and Computing*, pp. 42–45, jan 2014, doi: 10.1145/2591357.2591364.

[49] R. Raisamo, A. Hippula, S. Patomaki, E. Tuominen, V. Pasto, M. Hasu, "Testing usability of multimodal applications with visually impaired children," *IEEE Multimedia*, vol. 13, no. 3, pp. 70–76, 2006, doi: 10.1109/MMUL.2006.68.

[50] L. Garber, "Game Accessibility: Enabling Everyone to Play," *Computer*, vol. 46, no. 6, pp. 14–18, 2013, doi: 10.1109/mc.2013.206.

[51] J. Sánchez, T. Darin, R. Andrade, "Multimodal videogames for the cognition of people who are blind: Trends and issues," in *Proceedings of the International Conference on Universal Access in Human-Computer Interaction*, vol. 9177, 2015, pp. 535–546.

[52] T. Westin, K. Bierre, D. Gramenos, M. Hinn, "Advances in game accessibility from 2005 to 2010," in *Proceedings of the International Conference on Universal Access in Human-Computer Interaction*, vol. 6766, Berlin, 2011, pp. 400–409.

[53] B. Yuan, E. Folmer, F. C. Harris, "Game accessibility: A survey," *Universal Access in the Information Society*, vol. 10, pp. 81–100, mar 2011, doi: 10.1007/s10209-010-0189-5.

[54] M. Brown, S. L. R. Anderson, "Designing for Disability: Evaluating the State of Accessibility Design in Video Games," *Games and Culture*, vol. 0, no. 0, pp. 1–17, 2020, doi: 10.1177/1555412020971500.

[55] D. C. Moffat, D. Carr, "Using audio aids to augment games to be playable for blind people," in *Proceedings of the Audio Mostly Conference - A Conference on Interaction with Sound*, 2008, pp. 35–42.

[56] J. Sánchez, N. Baloian, T. Hassler, U. Hoppe, "AudioBattleship: Blind learners collaboration through sound," in *Proceedings of the Conference on Human Factors in Computing Systems*, 2003, pp. 798–799.

[57] B. A. Smith, S. K. Nayar, "The RAD: Making racing games equivalently accessible to people who are blind," in *Proceedings of the Conference on Human Factors in Computing Systems (CHI 2018)*, vol. April, Montréal, apr 2018, Association for Computing Machinery.

[58] R. J. Mccrindle, D. Symons, "Audio space invaders," in *Proceedings of the Third International Conference on Disability, Virtual Reality and Associated Technologies (ICDVRAT 2000)*, Alghero, 2000, pp. 59–65.

[59] T. Allman, R. K. Dhillon, M. A. Landau, S. H. Kurniawan, "Rock Vibe: Rock Band® computer games for people with no or limited vision," in *Proceedings of the 11th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS'09)*, Pittsburgh, 2009, pp. 51–58.

[60] B. Gasperetti, M. Milford, D. Blanchard, S. P. Yang, L. Lieberman, J. T. Foley, "Dance Dance Revolution and EyeToy Kinetic Modifications for Youths with Visual Impairments," *Journal of Physical Education, Recreation & Dance*, vol. 81, pp. 15–55, apr 2010, doi: 10.1080/07303084.2010.10598459.

[61] S. Kim, K. P. Lee, T. J. Nam, "Sonic-badminton: Audio-augmented Badminton game for blind people," in *Proceedings of the Conference on Human Factors in Computing Systems*, vol. 07-12-May-, may 2016, pp. 1922–1929, Association for Computing Machinery.

[62] T. Morelli, J. Foley, L. Columna, L. Lieberman, E. Folmer, "VI-Tennis: A vibrotactile/audio exergame for players who are visually impaired," in *Proceedings of the 5th International Conference on the Foundations of Digital Games (FDG 2010)*, Monterey, 2010, pp. 147–154, ACM.

[63] T. Morelli, J. Foley, E. Folmer, "VI-Bowling: A tactile spatial exergame for individuals with visual impairments," in *Proceedings of the 12th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS'10)*, Orlando, 2010, pp. 179–186.

[64] S. Vickers, H. Istance, M. J. Heron, "Accessible Gaming for People with Physical and Cognitive Disabilities: A Framework for Dynamic Adaptation," in *Proceedings of the Conference on Human Factors in Computing Systems (CHI 2013)*, vol. April, Paris, 2013, pp. 19–24, ACM.

[65] I. Khaliq, I. D. Torre, "A study on accessibility in games for the visually impaired," in *Proceedings of the 5th EAI International Conference on Smart Objects and Technologies for Social Good*, Valencia, 2019, pp. 142–148, ACM.

[66] T. Olsson, *Improving Accessibility for Shooter Games.* PhD dissertation, Chalmers University, 2020.

[67] B. Yuan, E. Folmer, "Blind Hero: Enabling Guitar Hero for the visually impaired," in *Proceedings of the 10th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS'08)*, Halifax, 2008, pp. 169–176, ACM.

[68] R. W. Li, C. Ngo, J. Nguyen, D. M. Levi, "Video-game play induces plasticity in the visual system of adults with amblyopia," *PLoS Biology*, vol. 9, aug 2011, doi: 10.1371/journal.pbio.1001135.

[69] I. Vedamurthy, M. Nahum, D. Bavelier, D. M. Levi, "Mechanisms of recovery of visual function in adult amblyopia through a tailored action video game," *Scientific Reports*, vol. 5, no. 8482, 2015, doi: 10.1038/srep08482.

[70] I. Vedamurthy, M. Nahum, S. J. Huang, F. Zheng, J. Bayliss,D. Bavelier, D. M. Levi, "A dichoptic custom-made action video game as a treatment for adult amblyopia," *Vision Research*, vol. 114, pp. 173–187, sep 2015, doi: 10.1016/j.visres.2015.04.008.

[71] S. Maidenbaum, A. Amedi, "Blind in a virtual world: Mobility-training virtual reality games for users who are blind," in *Proceedings of the IEEE Virtual Reality Conference (VR 2015)*, 2015, pp. 341–342.

[72] A. Rodríguez, I. Boada, S. Thió-Henestrosa, M. Sbert, "CPRforblind: A video game to introduce cardiopulmonary resuscitation protocol to blind people," *British Journal of Educational Technology*, vol. 49, pp. 636–645, jul 2018, doi: 10.1111/bjet.12627.

[73] J. Sánchez, M. Elías, "Science learning in blind children through audio-based games," in *Engineering the User Interface: From Research to Practice*, Springer London, 2009, pp. 87–102, doi: 10.1007/978-1-84800-136-7_7.

[74] L. Salvador-Ullauri, P. Acosta-Vargas, M. Gonzalez, S. Luján-Mora, "Combined method for evaluating accessibility in serious games," *Applied Sciences*, vol. 10, no. 18, p. 6324, 2020, doi: 10.3390/APP10186324.

[75] L. Salvador-Ullauri, P. Acosta-Vargas, S. Luján-Mora, "Web-based serious games and accessibility: A systematic literature review," *Applied Sciences*, vol. 10, no. 21, p. 25, 2020, doi: 10.3390/app10217859.

[76] R. Kupers, D. R. Chebat, K. H. Madsen, O. B. Paulson, M. Ptito, "Neural correlates of virtual route recognition in congenital blindness," in *Proceedings of the National Academy of Sciences of the United States of America*, vol. 107, 2010, pp. 12716–12721.

[77] L. Gagnon, F. C. Schneider, H. R. Siebner, O. B. Paulson, R. Kupers, M. Ptito, "Activation of the hippocampal complex during tactile maze solving in congenitally blind subjects," *Neuropsychologia*, vol. 50, no. 7, pp. 1663–1671, 2012, doi: 10.1016/j.neuropsychologia.2012.03.022.

[78] J. M. Loomis, R. L. Klatzky, R. G. Golledge, "Navigating without vision: Basic and applied research," *Optometry and Vision Science*, vol. 78, no. 5, pp. 282–289, 2001, doi: 10.1097/00006324-200105000-00011.

[79] J. P. Bigham, C. Jayant, A. Miller, B. White, T. Yeh, "VizWiz::LocateIt - Enabling blind people to locate objects in their environment," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Workshops) (CVPRW 2010)*, 2010, pp. 65–72.

[80] L. B. Merabet, J. Sánchez, "Audio-Based Navigation Using Virtual Environments: Combining Technology and Neuroscience," *AER Journal: Research and Practice in Visual Impairment and Blindness*, vol. 2, no. 3, pp. 128–137, 2009.

[81] L. B. Merabet, E. C. Connors, M. A. Halko, J. Sánchez, "Teaching the Blind to Find Their Way by Playing Video Games," *PLoS ONE*, vol. 7, sep 2012, doi: 10.1371/journal.pone.0044958.

[82] E. C. Connors, E. R. Chrastil, J. Sánchez, L. B. Merabet, "Action video game play and transfer of navigation and spatial cognition skills in adolescents who are blind," *Frontiers in Human Neuroscience*, vol. 8, p. 133, 2014, doi: 10.3389/fnhum.2014.00133.

[83] E. C. Connors, E. R. Chrastil, J. Sánchez, L. B. Merabet, "Virtual environments for the transfer of navigation skills in the blind: A comparison of directed instruction vs. video game based learning approaches," *Frontiers in Human Neuroscience*, vol. 8, may 2014, doi: 10.3389/fnhum.2014.00223.

[84] O. Lahav, "Using virtual environment to improve spatial perception by people who are blind," *Cyberpsychology and Behavior*, vol. 9, no. 2, pp. 174–177, 2006, doi: 10.1089/cpb.2006.9.174.

[85] J. Sánchez, M. Sáenz, M. Ripoll, "Usability of a multimodal videogame to improve navigation skills for blind children," in *Proceedings of the 11th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS'09)*, 2009, pp. 35–42.

[86] J. Sánchez, A. Tadres, A. Pascual-Leone, L. Merabet, "Blind children navigation through gaming and associated brain plasticity," in *Proceedings of the Virtual Rehabilitation International Conference (VR 2009)*, Haifa, 2009, pp. 29–36.

[87] J. Sánchez, M. Sáenz, "Three-dimensional virtual environments for blind children," *Cyberpsychology and Behavior*, vol. 9, no. 2, pp. 200–206, 2006, doi: 10.1089/cpb.2006.9.200.

[88] J. Sánchez, N. Baloian, "Issues in implementing awareness in collaborative software for blind people," in *Proceedings of the International Conference on Computers for Handicapped Persons*, vol. 4061 LNCS, Linz, 2006, pp. 1318–1325, Springer.

[89] M. Gröhn, T. Lokki, T. Takala, "Comparison of Auditory, Visual, and Audiovisual Navigation in a 3D Space," in *Proceedings of the 2003 International Conference on Auditory Display*, Boston, 2003, pp. 200–203.

[90] O. Lahav, D. Mioduser, "Haptic-feedback support for cognitive mapping of unknown spaces by people who are blind," *International Journal of Human-Computer Studies*, vol. 66, no. 1, pp. 23–35, 2008, doi: 10.1016/j.ijhcs.2007.08.001.

[91] G. Wersényi, "Virtual localization by blind persons," *AES: Journal of the Audio Engineering Society*, vol. 60, no. 7-8, pp. 568–579, 2012.

[92] S. W. Mereu, R. Kazman, "Audio enhanced 3D interfaces for visually impaired users," in *Proceedings of the ACM SIGCAPH: Computers and the Physically Handicapped*, 1996, pp. 10–15, ACM.

[93] T. Dingler, J. Lindsay, B. N. Walker, "Learnability of Sound Cues for Environmental Features: Auditory Icons, Earcons, Spearcons, and Speech," in *14th International Conference on Auditory Display*, 2008, pp. 1–6.

[94] J. Sánchez, M. Lumbreras, L. Cernuzzi, "Interactive virtual acoustic environments for blind children: computing, usability, and cognition," in *Extended Abstracts on Human Factors in Computing Systems (CHI 2001)*, Seattle, 2001, pp. 65–66.

[95] B. N. Walker, J. Lindsay, J. Godfrey, "The audio abacus: Representing numerical values with nonspeech sound for the visually impaired," *ACM SIGACCESS Accessibility and Computing*, p. 9, sep 2003, doi: 10.1145/1029014.1028634.

[96] A. Darvishi, V. Guggiana, E. Munteanu, H. Schauer, M. Motavalli, M. Rauterberg, "Synthesizing non-speech sound to support blind and visually impaired computer users," in *Proceedings of the International Conference on Computers for Handicapped Persons*, Linz, 1994, pp. 385–393, Springer.

[97] K. Spiel, S. Bertel, M. Heron, "Navigation and immersion of blind players in text-based games," *The Computer Games Journal*, vol. 3, pp. 130–152, oct 2014, doi: 10.1007/bf03392361.

[98] O. Balan, A. Moldoveanu, F. Moldoveanu, "Navigational audio games: An effective approach toward improving spatial contextual learning for blind people," *International Journal on Disability and Human Development*, vol. 14, no. 2, pp. 109–118, 2015, doi: 10.1515/ijdhd-2014-0018.

[99] K. Allain, B. Dado, M. V. Gelderen, O. Hokke, M. Oliveira, R. Bidarra, N. D. Gaubitch, R. C. Hendriks, B. Kybartas, "An audio game for training navigation skills of blind children," in *Proceedings of the IEEE 2nd VR Workshop on Sonic Interactions for Virtual Environments (SIVE 2015)*, Arles, 2015, IEEE.

[100] F. Wong, R. Nagarajan, S. Yaacob, "Application of stereovision in a navigation aid for blind people," in *Proceedings of the 2003 Joint Conference of the 4th International Conference on Information, Communications and Signal Processing and 4th Pacific-Rim Conference on Multimedia*, vol. 2, 2003, pp. 734–737.

[101] L. Valente, C. S. D. Souza, B. Feijó, "An exploratory study on non-visual mobile phone interfaces for games," in *Proceedings of the VIII Brazilian Symposium on Human Factors in Computing Systems*, Porto Alegre, 2008, pp. 31–39.

[102] E. Krajnc, J. Feiner, S. Schmidt, "User centered interaction design for mobile applications focused on visually impaired and blind people," in *Proceedings of the Symposium of the Austrian HCI and Usability Engineering Group (USAB 2010)*, vol. 6389 LNCS, 2010, pp. 195–202.

[103] E. Folmer, B. Yuan, D. Carr, M. Sapre, "TextSL: A command-based virtual world interface for the visually impaired," in *Proceedings of the 11th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS'09)*, Pittsburgh, 2009, pp. 59–66.

[104] P. Bach-y Rita, S. W. Kercel, "Sensory substitution and the human-machine interface," *Trends in Cognitive Sciences*, vol. 7, no. 12, pp. 541–546, 2003, doi: 10.1016/j.tics.2003.10.013.

[105] T. Morelli, E. Folmer, "Real-time sensory substitution to enable players who are blind to play video games using whole body gestures," *Entertainment Computing*, vol. 5, no. 1, pp. 83–90, 2014, doi: 10.1016/j.entcom.2013.08.003.

[106] R. Gutschmidt, M. Schiewe, F. Zinke, H. Jürgensen, "Haptic emulation of games: Haptic Sudoku for the blind," in *Proceedings of the 3rd International Conference on PErvasive Technologies Related to Assistive Environments*, Samos, 2010, pp. 1–8.

[107] R. Iglesias, S. Casado, T. Gutiérrez, J. I. Barbero, C. A. Avizzano, S. Marcheschi, M. Bergamasco, "Computer graphics access for blind people through a haptic and audio virtual environment," in *Proceedings of the 3rd IEEE International Workshop on Haptic, Audio and Visual Environments and their Applications (HAVE 2004)*, 2004, pp. 13–18.

[108] L. A. Johnson, C. M. Higgins, "A navigation aid for the blind using tactile-visual sensory substitution," in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology*, vol. 1, 2006, pp. 6289–6292.

[109] C. Sjöström, "Using haptics in computer interfaces for blind people," in *Proceedings of the Conference on Human Factors in Computing Systems*, 2001, pp. 245–246.

[110] M. De Pascale, S. Mulatto, D. Prattichizzo, "Bringing Haptics to second life for visually impaired people," in *Proceedings of the International Conference on Human Haptic Sensing and Touch Enabled Computer Applications (Eurohaptics 2008)*, vol. 5024 LNCS, Madrid, 2008, pp. 896–905.

[111] D. R. Chebat, F. C. Schneider, R. Kupers, M. Ptito, "Navigation with a sensory substitution device in congenitally blind individuals," *Neuroreport*, vol. 22, no. 7, pp. 342–347, 2011, doi: 10.1097/WNR.0b013e3283462def.

[112] S. Chandrashekar, D. Fels, T. Stockman, R. Benedyk, "Using think aloud protocol with blind users: A case for inclusive usability evaluation methods," in *Proceedings of the Eighth International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS 2006)*, vol. 2006, 2006, pp. 251–252.

Manuel López Ibáñez

Manuel López Ibáñez is a researcher at the Department of Software Engineering and Artificial Intelligence of the Complutense University of Madrid. He holds degrees in Media Studies and Journalism from the Carlos III University (2012), a master's degree in Video Game Design (2015) from the Complutense University of Madrid, and a Ph. D in Computer Science from the same university (2019). He is also a pianist and composer, and has published three studio albums under the pseudonym of "Asomnu", as well as the soundtrack for the video game Song of Horror, by Protocol Games. Manuel has been involved in research ranging from virtual reality audio to accessible computer entertainment. He also has a keen interest in the study of sound perception, as well as in the relationship between audio and emotions. Additionally, he is the developer of LitSens: a multi-engine adaptive music generator specifically designed for first-person and virtual reality interactive experiences.

Alejandro Romero-Hernández

Alejandro Romero has a degree in Computer Engineering from the Complutense University of Madrid (UCM) and a master's degree in Digital Language Arts: Advanced Studies in Electronic Textualities from the same university (2016). Since 2015, he has been dedicated to the development of artistic video games for its application in the educational world. He is currently developing his PhD in Computer Engineering at the same university. His main research interest is to develop different artistical tools using new technologies as video games or different types of extended realities in order to increase the awareness of the general public to these different arts. He is currently collaborating with artistic companies such as the National Ballet of Spain, the National dance company, the royal theater of Spain, etc.

Borja Manero

Borja Manero is Associate Professor and Deputy Director of the Software Engineering and Artificial Intelligence Department, at Complutense University of Madrid. He holds a PhD in Computers Sciences and a degree in Physics, both from the Complutense University of Madrid (www.ucm.es). Besides, he studied Dramatic Arts, at Réplika Theater School of Madrid. Borja Manero is a researcher focused on the potential of the artificial intelligence and video-games to improve artistic education and communication skills. In this line, he is the director of the public speaking initiative of UCM. His involvement in different research projects funded by diverse entities (such as the Spanish Ministry for Science and Education or the European Union Framework Programme) have resulted in various articles in high impact research journals and contributions to international conferences and workshops. He also collaborated with different institutions in different countries (including a two-years scholarship at CERN, and a one-year visiting scholarship at Harvard University) resulting in a broad international experience. Finally, he co-chairs the Arts and Technology in Leadership study group at RCC/Harvard.

María Guijarro

María Guijarro, associate professor in the department of Computer Architecture and Automation, Faculty of Computer Science of the Complutense University of Madrid. She received a degree in Computer Engineering and a Master's degree in Computer Research. In 2009, she received a PhD in Computer Engineering, specialising in Artificial Intelligence, with Extraordinary PhD Prize mention. Since then, her career has focused on designing Artificial Intelligence techniques in the area of Computer Vision, e-learning, accessible technologies and machine learning. She has more than 50 scientific publications and large-scale project management.

Automatic Classification of Oral Pathologies Using Orthopantomogram Radiography Images Based on Convolutional Neural Network

Anuradha Laishram*, Khelchandra Thongam

Department of Computer Science and Engineering, National Institute of Technology Manipur (India)

Received 20 July 2020 | Accepted 31 July 2021 | Published 29 October 2021

unir
LA UNIVERSIDAD
EN INTERNET

Abstract

An attempt has been made to device a robust method to classify different oral pathologies using Orthopantomogram (OPG) images based on Convolutional Neural Network (CNN). This system will provide a novel approach for the classification of types of teeth (viz., incisors and molar teeth) and also some underlying oral anomalies such as fixed partial denture (cap) and impacted teeth. To this end, various image preprocessing techniques are performed. The input OPG images are resized, pixels are scaled and erroneous data are excluded. The proposed algorithm is implemented using CNN with Dropout and the fully connected layer has been trained using hybrid GA-BP learning. Using the Dropout regularization technique, over fitting has been avoided and thereby making the network to correctly classify the objects. The CNN has been implemented with different convolutional layers and the highest accuracy of 97.92% has been obtained with two convolutional layers.

Keywords

Classification, CNN, Dropout, Image Pre-processing, Orthopantomogram Radiography Images.

DOI: 10.9781/ijimai.2021.10.009

I. Introduction

In human anatomy, the mouth is one of the principal organs and being the first portion of gastrointestinal tract, it plays a significant role in communication, breathing and digestion. This cavity organ involves different organs, for example, teeth, tongue, hard sense of taste and delicate sense of taste. Great oral wellbeing can give numerous extraordinary advantages. It improves our visual appearance, the inspiration of our mentality just as improving the personal satisfaction. An unhealthy mouth, exceptionally gum malady and tooth rot are the principal two reasons for tooth loss and furthermore offer ascent to numerous clinical issues and causes inconvenience, draw out agony, distortion. So, it makes sense to give oral health the best possible care. There are various kinds of dental x-rays to check the status of our oral wellbeing. Dental x-rays are a helpful demonstrative instrument of any dental consideration treatment plan which utilized low degree of radiation to catch pictures of the inside of the mouth region. The utilization of Dental x-ray images had denoted an unprecedented milestone in clinical discovering in light of its brief openness and for the most part, lower radiation estimation. One such radiograph image utilized in our work is Orthopantomogram radiograph.

An Orthopantomogram (OPG) by and large known as panoramic radiography catches wide perspective of the mouth in a solitary x-ray image including the upper and lower jaw. It distinguishes the situation of completely developed just as affected teeth and furthermore recognize other bone anomalies An Orthopantomogram radiograph image is shown in Fig. 1.

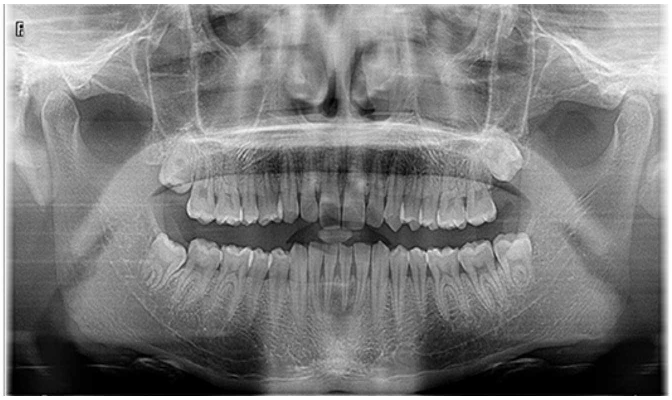


Fig. 1. An Orthopantomogram X-ray.

Over the recent years Convolutional Neural Network has made a significant contribution in the field of medical science [1]-[5]. Many researchers started working on the classification of dental images and signals recently [6]-[12]. Some researchers have explained the use of Convolutional network in the field of dentistry.

Oprea et al. [13] proposed how image processing techniques can help to check the x-ray and examine the extent to which the caries lesion is present and then classify the type of caries present in the dental radiograph. Two different operations such as thresholding images, image differentiation are used in order to detect the dental carries and hence the efficiency of the algorithm is compromised. The database used for the experimentation is not of a good quality due to the lack of better instrumentation (x-ray machines) at that time.

Prajapati et al. [14] made an endeavor towards precise arrangement of three dental ailments, specifically Dental Caries, Periapical Infection

* Corresponding author.

E-mail address: annu0286@gmail.com

and Periodontitis. Marked dataset comprising of 251 Radiovisiography (RVG) x-ray images is utilized. In this paper, CNN is used for determination of small labelled dental dataset. Likewise, Transfer Learning is utilized to improve the precision. The overall accuracy of 88.46% is accomplished. In Yu. [15], a method in which SIFT, HOG2Ö2, HOG3Ö3, and Color features are used as image descriptors, the Bag-of-Words (BoW) is applied and also a spatial pyramid pipeline. K-nearest neighbors algorithm (k-NN) and an error correcting output coding support vector machine (ECOC-SVM) classifiers are deployed. 4-layer CNN model and 16-layer CNN model methods were proposed. Sharpness and precision of the x-ray images were not determined due to which the system was not very reliable. A maximum accuracy of 90.36% was achieved but the dataset size was very small.

Srivastava et al. [16] developed a Computer Aided Diagnosis (CAD) system that enhances the performance of dentists in detecting wide range of dental caries. Annotations were made using loose polygons around the carries and classification is done using a Fully Convolutional Neural Network. The dataset was collected from 100+ different clinics and hence providing different image quality and also required some type of normalization. Since it was difficult to annotate the carries, it was done using a loose polygon around the carries. Also the amount of False Positive test results was also high. Veena et al. [17] presented two distinct image processing algorithms for detection of dental anomalies. The first approach for the detection of dental caries uses hybridized negative transformation. The second approach uses statistical texture analysis for the dental images containing cysts along with dental caries. Gray Level Co-occurrence Matrix (GLCM) is used for feature extraction from panoramic images. The system was experimented on several datasets and high percentage of false positive were output in the first method.

Zakirov et al. [18] proposed a system that achieves 96.3% accuracy in tooth localization and an average of 0.94 AUROC for 6 common tooth conditions using deep convolutional neural networks and algorithmic heuristics. Cone-Beam CT imaging is a time-consuming process that requires a physician to work with complicated software. Since it a new imaging technique in dentistry, the availability of the machines is questionable. Also, the very small amount of dataset cannot be considered to present the validity of the proposed system.

In our study a Computer Aided Diagnosis system is being proposed in order to assist dentists in detection and diagnose the dental pathology faster and more accurately and also provide better judgment in the treatment of any oral diseases. The study will provide a novel approach in classifying the type of teeth such as incisors and molar teeth and also some underlying oral anomalies such as fixed partial denture and impacted teeth present in an OPG using deep learning technique. The Convolutional Neural Network with Dropout is used to classify the images and is implemented using Keras, a python neural network API. The fully connected layer of the CNN has been trained using hybrid GA-BP learning and provides a classification accuracy of 97.92%. The remaining paper is organized as follows: Section II describes the proposed methodology. Section III gives the details of Experimental Results. Section IV gives the comparison details of our proposed method with other standard method. Finally, the conclusion is given in section V.

II. Proposed Methodolgy

A flow diagram of the proposed methodology is given in Fig. 2. The proposed system, however, is an end-to-end solution, which classifies some of the dental pathology directly from the original radiographs with a bit of image pre-processing of the dental panoramic radiographs. This has been possible by using Convolutional Neural Networks with deep learning. A convolutional neural network (CNN or ConvNet) is one of the most popular algorithms for deep learning, a type of machine learning in which a model learns to perform classification tasks directly from images. The methodology consists of the following modules:

1. Image Acquisition
2. Image Pre-processing
3. Building of a CNN
4. Image Augmentation
5. Training, Validation and Testing

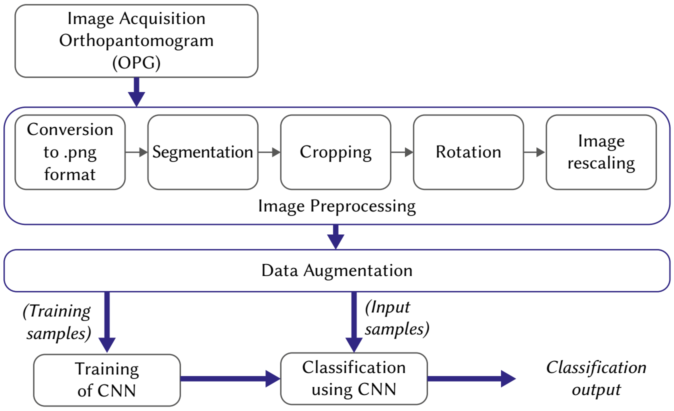


Fig. 2. Flow diagram of the proposed algorithm.

A. Image Acquisition

Image acquisition is the initial step of any vision framework. The images acquired are not a processed image. After the acquisition of images, further processing techniques has been applied to process the image. The process of image acquisition is done using Care stream (KODAK) Dental's CS 8100; an OPG machine that has machine has gained exceptional value in the market for its efficient performance and sturdiness.

B. Image Pre-processing

Image pre-processing is a technique to select and enhance the region of interest of acquired digitized images by applying various mathematical operations. The purpose of pre-processing was discussed in [19], [20]. In our study, we apply different image pre-processing techniques namely edge detection, cropping, rotation and image scaling. Before applying any preprocessing techniques, the raw data which is in .pano format is converted to .png format in order to be used for further processing.

Edge Detection: Edge detection is one of the vital steps of image preprocessing. Although cropping operation segments the teeth from the original image, it does not crop along the boundary of teeth. Edge detection technique are used to define the edge of the teeth to find the region of interest. In our study to perform the task of segmentation the Active Contour method [21] (also known as snake method) was used.

Cropping: Cropping is one of the most well-known image activity which is performed on the obtained images to expel the undesirable composed names and to complement the region of interest bringing about a rectangular shape comprising of just the region of interest which is required for the classification and furthermore lessen the potential outcomes of mistake.

Image Rotation: To get better visualization of the acquired images, the images are rotated about its center to a specified degree (θ=45, 90). This is performed to maintain uniformity in images representation.

Image scaling: In image pre-processing, image scaling alludes to the resizing of a picture. Image resizing is essential when we have to increment or diminishing the all-out number of pixels, though remapping can occur when you are correcting for lens distortion or rotating an image.

C. Building of CNN

Convolutional Neural Network consists of Convolution as the first layer to extract features from the image. The feature maps are passed on to activation function known as Rectified Linear Unit(ReLU) and then the input is passed on to pooling layers and finally given to the fully connected network, which gives the output [22]-[28]. Fig. 3 shows our CNN model which contains two convolutional layers with two pooled feature mapped layers. Our CNN architecture has been implemented using Dropout. Dropout is a regularization technique for neural network models to prevent the networks from over fitting by ignoring randomly selected neurons during training. Fig. 8 shows the proposed CNN architecture using two convolutional layers with dropouts.

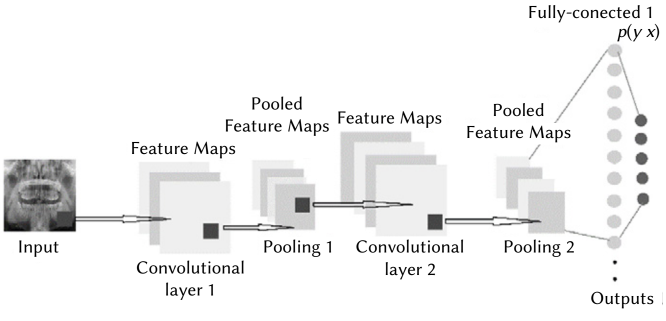


Fig. 3. Model of the Proposed CNN.

1. Convolution Layers

Convolutional Neural Network consists of Convolution as the first layer to extract features from the image also preserves the relationship between pixels by learning images features using small square of input data known as feature maps. The input image is convolved with a feature detector also called kernel or filter. To increase its values by the original pixel values is the task of this filter. The equation for this mathematical operation is given as:

$$s[t] = (x * w)[t] = \sum_{a=-\infty}^{a=\infty} x[a]w[a+t] \quad (1)$$

where $s[t]$ is the feature map, x is the input, w is the filter.

Fig. 4 shows convolutional layers with feature mapping function and activation function.

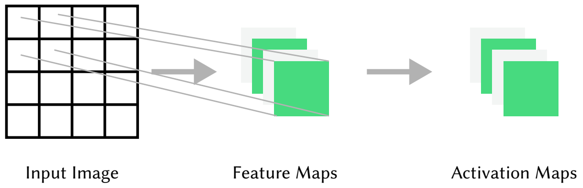


Fig. 4. Convolution Layers.

2. Non-linear Units

Since this proposed method is to classify dental pathology radiograph images and image classification is a nonlinear problem, non-linear unit increases non-linearity into the input images. As non-linearity is achieved through an activation function known as the ReLU (Rectified Linear Unit) function. Without this property a network would not be sufficiently intense and will not be able to model the response variable. Mathematically, ReLU converts all negative values to zero which decreases the computational complexity. Fig. 5 shows the graph of Rectified Linear Unit. The equation for ReLU activation function is as follows:

$$Equation: f(x) = \begin{cases} 0, x < 0 \\ x, x \geq 0 \end{cases} \quad (2)$$

$$Derivative: f'(x) = \begin{cases} 0, x < 0 \\ 1, x \geq 0 \end{cases} \quad (3)$$

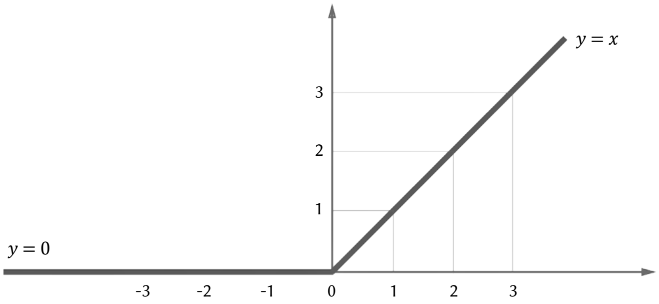


Fig. 5. Rectified Linear Unit (ReLU).

3. Pooling Layer

The next layer in a convolutional network comprises three steps which include pooling, down sampling and subsampling. This layer simplifies the output by performing nonlinear down sampling, reduces the spatial size of the representation to reduce the number of parameters and computation in the network. Out of many pooling techniques, Max pooling is used as it preserves the most matching feature in the activation map and also it helps to gain spatial invariance.

4. Fully Connected Layer

Fig. 6 shows a fully connected layer. Before the pooled layers are completely fed to the fully connected layer, the pooled feature map needs to be converted to a 1-D vector. This can be achieved by a process called the flattening process. The resultant 1-D vector contains the spatial structure information or pixel pattern and passed on to the fully connected network. The final layer of the CNN architecture which main purpose is to combine the feature into more attributes uses a classification layer such as softmax to provide the classification output. A fully connected layer works with high level features which strongly correlates to a particular class and particular weights that the products between the weights and the preceding layer has the accurate probabilities for the different classes.

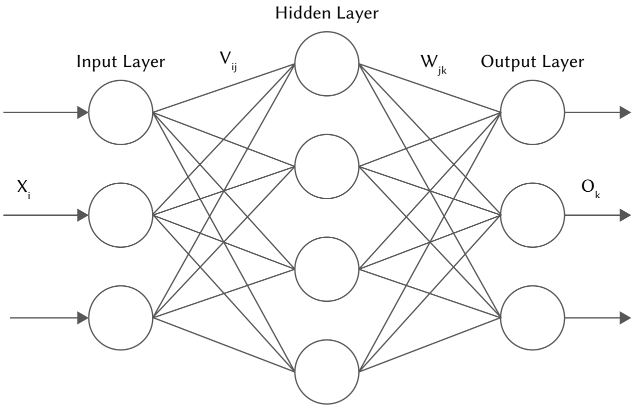


Fig. 6. Fully Connected Layer.

The proposed fully connected layer model contains three layers: input, hidden and output which is used to classify the images into different classes. The layers are connected by synaptic weights. The learning of the network is realized by a hybrid system of Genetic Algorithm (GA) and Back Propagation (BP). The GA is an evolutionary algorithm and is based on the principle of natural selection and bio-inspired operations.

GA is a global search algorithm good at determining the rough position of the global optimal solutions and BP is a local search algorithm good at fine tuning. It is possible to have both advantages by combining GA and BP. Use GA first to find the rough position of the global optimal solution and then use BP to refine.

Steps of the GA-BP hybrid learning:

1. Use GA learning first to find a good starting point
2. Use BP learning to find a local Solution
3. If the solution is worse than the current best result, return to Step 1; otherwise, continue
4. If the current best good enough, stop; otherwise, return to Step 1

Important parameters that are used during the learning process are as follows:

x_i: The i^{th} input

y_i: The output of the j^{th} hidden neuron

o_k: The output of the k^{th} output neuron

d_k: The desired output

v_{ij}: The weight from the i^{th} input to the j^{th} hidden neuron

w_{ij}: The weight from the j^{th} hidden neuron to the k^{th} output neuron

The steps of the GA learning are as follows:

i) Initialization of the weights

The weights of the connections of the neurons from one layer to another of the fully connected network will be represented by GA solution which is a binary string.

The number of total weights, TW is given by

$$TW = I * HN + HN * ON \tag{4}$$

where I is the size of each input sample, HN is the number of hidden neurons, ON is the number of output neurons.

In our work, I = 3136, HN = 64 and ON = 4, so TW comes out to be 200960.

The gene length, GL is given by the equation

$$GL = \left(NB * (I * HN + HN * ON)\right) \tag{5}$$

Where NB = Number of bits per weight

In our work, NB = 6, so GL comes out to be 1205760.

ii) Phenotype reconstruction using genotype

Consider,

$$y_i = \sum_{k=1}^{NB} b_{ik}\, 2^{-k} \tag{6}$$

Where b_{ik} is the k^{th} bit for the i^{th} weight. Then,

$$w_i = y_i \times A + B \tag{7}$$

w_i is the i^{th} weight present in the string or solution, A is the scaling factor and B is the shifting factor.

In our application, we set A=20 and B=10 so that the weight will take value from [-10, 10].

iii) Output of the hidden layer and the output layer

We calculate the outputs of the hidden neurons using the relations:

$$s1 = \sum_{i,j} v_{ji} \times x_{pi} \tag{8}$$

$$y_j = sigmoid(s1) \tag{9}$$

Where y_i is the output of the j^{th} hidden neurons.

We calculate the output of the output neurons with the equations:

$$s2 = \sum_{j,k} w_{kj} \times y_j \tag{10}$$

$$o_k = sigmoid(s2) \tag{11}$$

Where o_k is the output of the k^{th} output neurons. The function sigmoid is a unipolar sigmoid function.

These two operations of finding the output are performed for all the input patterns. Then the Error is updated with the equation:

$$E = \frac{1}{2}\sum_{k=0}^{k} (d_k - o_k)^2 \tag{12}$$

d_k is the desired output. This process is performed until all the training samples have been used.

iv) Calculate the fitness of string or solution.

The fitness definition to calculate the fitness of the string/solution is given by the equation

$$fitness = (1 - E)/N \tag{13}$$

Where, N is the number of patterns or training examples. The above processes are repeated from step ii) for all the strings or solutions of the population.

v) Selection

We find out the string with the highest fitness value. The weights representing this string with highest fitness value will be used for further operation.

vi) Reproduction

The population is modified using operators namely crossover and mutation. The above processes from step ii) are repeated for many generations.

The steps of the BP learning are as follows:

The size of each input sample is I. There are J hidden neurons and K output neurons.

1. Get a training input, calculate the output of hidden neurons and output neurons
2. Calculate the learning signals

   The learning/error signal δ_k produced by the k^{th} output neuron is given as:

   $$\delta_k = (d_k - o_k) * (1 - o_k) * o_k \tag{14}$$

   Using δ_k, we can calculate δ_j as follows:

   $$\delta_j' = \sum_{k=0}^{k} \delta_k * w_{jk} \tag{15}$$

   $$\delta_j = \left(1 - y_j\right) * y_j * \delta_j' \tag{16}$$

3. Update the weights

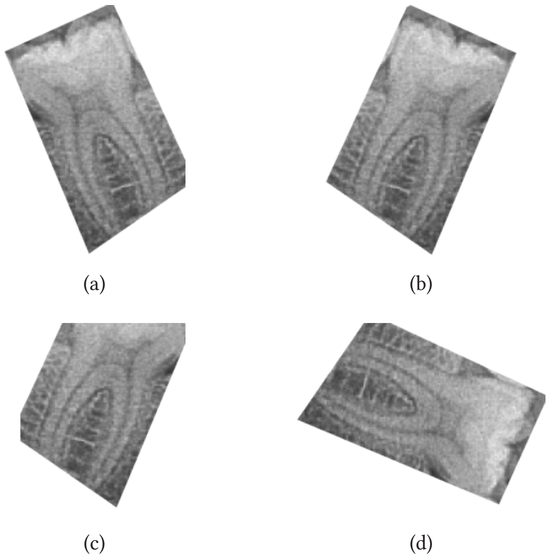
   The weights of the output layer are updated using the equation:

   $$w_{jk} = w_{jk} + \eta * \delta_k * y_j \tag{17}$$

   Where η is the learning rate.

   The weights of the hidden layer is updated using the equation

   $$v_{ij}(n+1) = v_{ij}(n) + \eta * \delta_j(n) * x_i(n) \tag{18}$$



(a) (b) (c) (d)

Fig. 7. Image Augmentation: (a) Original image (b) Horizontal flip (c) Height shift (d) Rotation by 90°.

4. Update the Error

   Error is updated using the equation:

$$Error = \frac{1}{2}\sum_{k=0}^{k}(d_k - o_k)^2 \tag{19}$$

5. Check if all the training patterns have been used, if not return to Step 1.
6. If Error is smaller than desired error, then terminate otherwise repeat the whole process

D. Image Augmentation

Data augmentation is a way of creating new 'data' with different orientations [29], [30]. The benefits of this are twofold, the first being the ability to generate 'more data' from limited data and secondly it prevents over fitting making it less likely that the neural network recognizes unwanted characteristics in the data-set. Image data augmentation is applied to the training dataset, and not to the validation or test dataset as shown in Fig. 7. This is different from data preparation such as image resizing and pixel scaling; they must be performed consistently across all datasets that interact with the model.

E. Training, Validation and Testing

800 images belonging to 4 classes are used for training, 116 images belonging to 4 classes for validation. The training has been done with 16 numbers of batches, validation with 2 numbers of batches. Python Fit-generator method is employed for training, validation and testing of the dataset. The generator runs parallel to the model which in turn allows training the model on GPU and in parallel to perform real-time data augmentation on images on CPU.

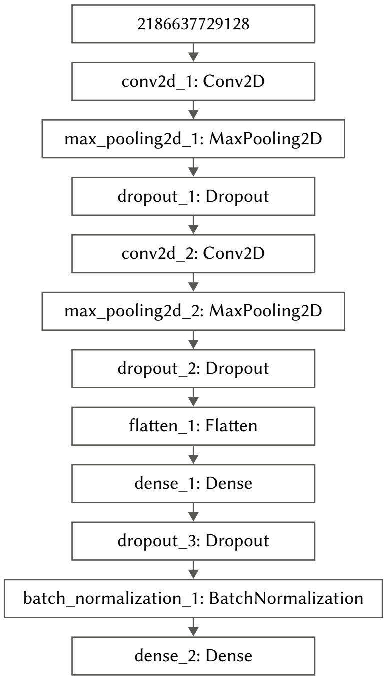


Fig. 8. Proposed architecture using two Convolutional Layer.

III. Experimental Results and Discussions

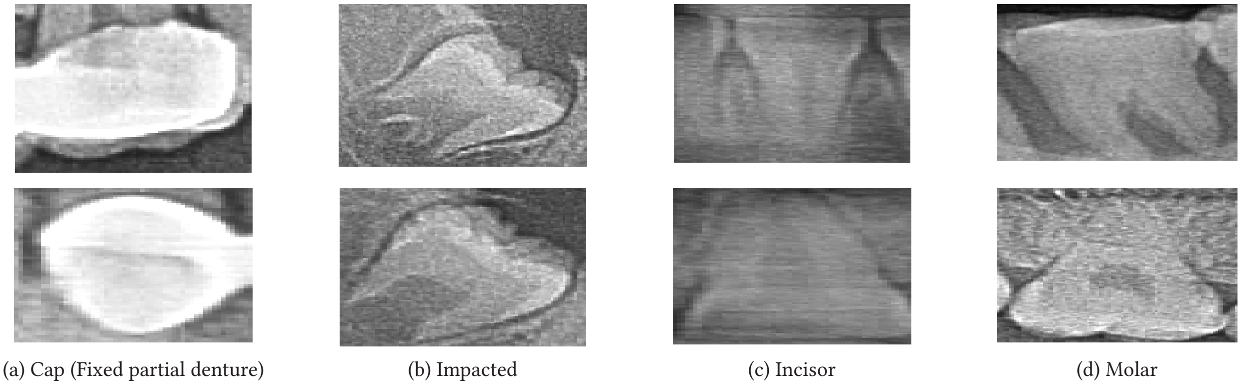
CNN is implemented for training, validation and testing using different convolutional layers. Using 2 convolutional layers, the best accuracy is obtained and the results are provided in the section.

A. Image Pre-processing

Different dental anomalies and different types of teeth are shown in Fig. 9 which is a result of image pre-processing. a) and b) are types of dental anomalies namely Cap(fixed partial denture) and Impacted teeth whereas c) and d) are types of teeth namely Incisor and Molar.

B. Dataset

As shown in Table I, 800 images belonging to 4 classes are used for training, 116 images belonging to 4 classes for validation. The training has been done with 16 numbers of batches, validation with 2 numbers of batches. Testing has been done with 120 random samples.



(a) Cap (Fixed partial denture) (b) Impacted (c) Incisor (d) Molar

Fig. 9. Input samples of Class A, B, C, D respectively.

TABLE I. Split of Samples in the Dataset

Dataset	Number of Images
Training Set	200 images for class A
	200 images for class B
	200 images for class C
	200 images for class D
Validation Set	29 images for class A
	29 images for class B
	29 images for class C
	29 images for class D
Test Set	120 random images for testing

C. GA Fitness and Accuracy and Loss for Training and Validation

GA is use at determining the rough position of the global optimal solutions and BP is use for fine tuning. We use GA learning first to find a good starting point.

The Number of bits per weight is 6. The GeneLength comes out to be 1205760. The population size is 60. The mutation rate is set as 0.001. The fitness of the solution/string obtained after some GA generations is given in Table II.

TABLE II. Highest Fitness of the Solution for Different GA Generations

Sl. No.	Generation	Training Loss	Fitness Value
1.	30	1.633	0.397
2.	60	0.792	0.484
3.	90	0.524	0.563
4.	120	0.501	0.621
5.	150	0.495	0.684

After GA learning generates global optimal weights, we use BP learning to refine the weights.

i) As shown in Fig. 10, using 2 convolutional layers and epochs upto 400, the training loss and accuracy comes out to be 0.19 and 97.57 respectively. The validation loss and accuracy comes out to be 0.18 and 97.94 respectively.

ii) As shown in Fig. 11, using 2 convolutional layers and epochs upto 1000, the training loss and accuracy comes out to be 0.23 and 96.32 respectively. The validation loss and accuracy comes out to be 0.69 and 83.85 respectively. We can see that validation loss rises more and validation accuracy drops down because of over fitting.

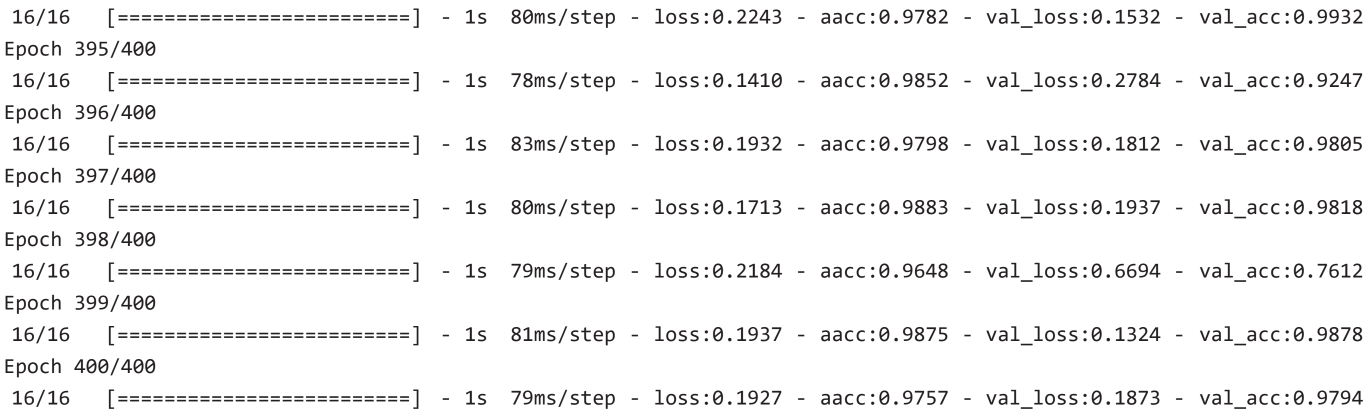
D. Model Graph

i) Fig. 12(a) shows Model Accuracy graph of Training and Validation between Epoch and Accuracy with epochs upto 400. We can observe that almost as the epoch increases, the accuracy increases with some drop down in between. The training accuracy and validation accuracy comes out to be 97.57 and 97.94 respectively.

ii) Fig. 12(b) shows Model Loss graph of Training and Validation between Epoch and Loss with epochs upto 400. Here, we can observe that almost as the epoch increases, the loss decreases with some rises in between. The training loss and validation loss comes out to be 0.19 and 0.18 respectively.

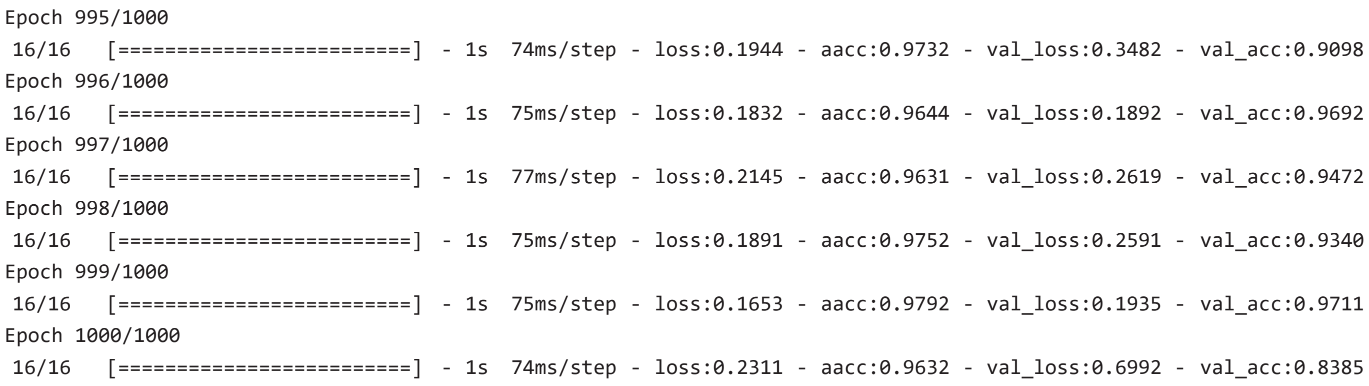
iii) Fig. 13(a) shows Model Accuracy graph of Training and Validation between Epoch and Accuracy with epochs up to 1000. The training accuracy and validation accuracy comes out to be 96.32 and 83.85 respectively. We can see that validation accuracy drops down because of over fitting.

iv) Fig. 13(b) shows Model Loss graph of Training and Validation between Epoch and Loss with epochs up to 1000. The training loss and validation loss comes out to be 0.23 and 0.69 respectively. We can see that validation loss rises because of over fitting.



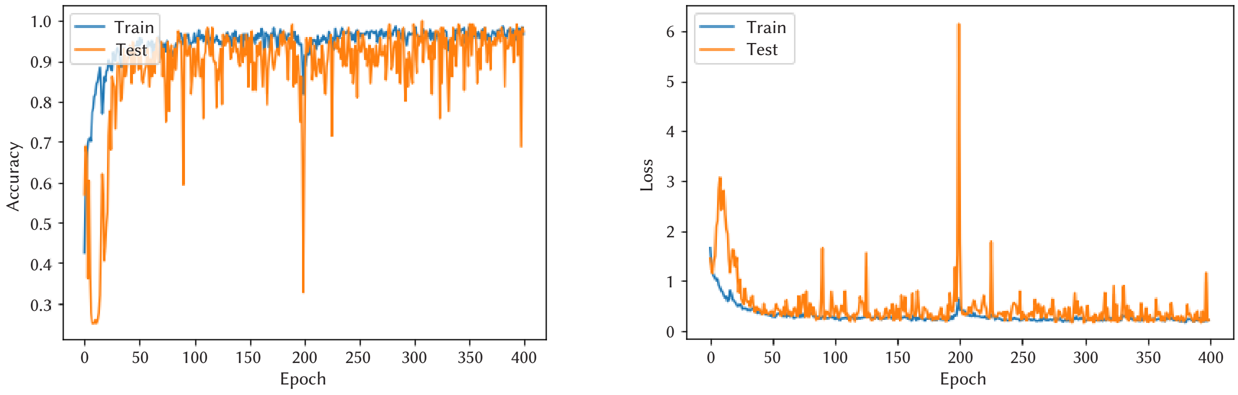
```
 16/16   [=========================] - 1s  80ms/step - loss:0.2243 - aacc:0.9782 - val_loss:0.1532 - val_acc:0.9932
Epoch 395/400
 16/16   [=========================] - 1s  78ms/step - loss:0.1410 - aacc:0.9852 - val_loss:0.2784 - val_acc:0.9247
Epoch 396/400
 16/16   [=========================] - 1s  83ms/step - loss:0.1932 - aacc:0.9798 - val_loss:0.1812 - val_acc:0.9805
Epoch 397/400
 16/16   [=========================] - 1s  80ms/step - loss:0.1713 - aacc:0.9883 - val_loss:0.1937 - val_acc:0.9818
Epoch 398/400
 16/16   [=========================] - 1s  79ms/step - loss:0.2184 - aacc:0.9648 - val_loss:0.6694 - val_acc:0.7612
Epoch 399/400
 16/16   [=========================] - 1s  81ms/step - loss:0.1937 - aacc:0.9875 - val_loss:0.1324 - val_acc:0.9878
Epoch 400/400
 16/16   [=========================] - 1s  79ms/step - loss:0.1927 - aacc:0.9757 - val_loss:0.1873 - val_acc:0.9794
```

Fig. 10. Accuracy and loss upto 400 epochs.



```
Epoch 995/1000
 16/16   [=========================] - 1s  74ms/step - loss:0.1944 - aacc:0.9732 - val_loss:0.3482 - val_acc:0.9098
Epoch 996/1000
 16/16   [=========================] - 1s  75ms/step - loss:0.1832 - aacc:0.9644 - val_loss:0.1892 - val_acc:0.9692
Epoch 997/1000
 16/16   [=========================] - 1s  77ms/step - loss:0.2145 - aacc:0.9631 - val_loss:0.2619 - val_acc:0.9472
Epoch 998/1000
 16/16   [=========================] - 1s  75ms/step - loss:0.1891 - aacc:0.9752 - val_loss:0.2591 - val_acc:0.9340
Epoch 999/1000
 16/16   [=========================] - 1s  75ms/step - loss:0.1653 - aacc:0.9792 - val_loss:0.1935 - val_acc:0.9711
Epoch 1000/1000
 16/16   [=========================] - 1s  74ms/step - loss:0.2311 - aacc:0.9632 - val_loss:0.6992 - val_acc:0.8385
```

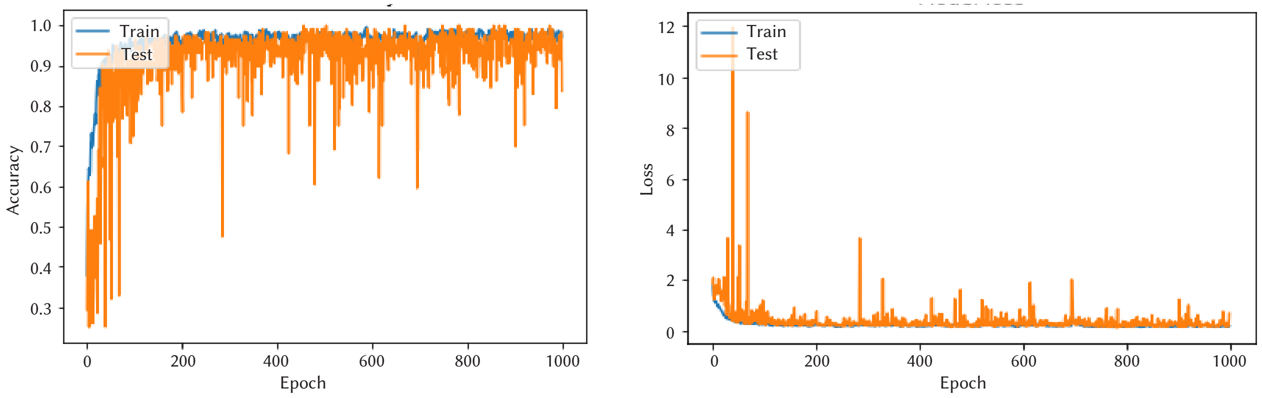
Fig. 11. Accuracy and loss upto 1000 epochs.



(a) Model accuracy

(b) Model loss

Fig. 12. Model Graph up to epochs 400.



(a) Model accuracy

(b) Model loss

Fig. 13. Model Graph up to epochs 1000.

E. Performance Analysis

For the performance analysis of the classification, Recall, Precision, F1 Score and Accuracy (Acc) are used.

$$Precision=\frac{TP}{TP+FP} \tag{20}$$

$$Recall=\frac{TP}{TP+FN} \tag{21}$$

$$F1Score=\frac{2*Recall*precision}{Recall+precision} \tag{22}$$

$$Accuracy=\frac{TP+TN}{TP+TN+FP+FN}\times 100 \tag{23}$$

where, TP, TN, FP and FN are the True Positive, True Negative, False Positive and False Negative respectively.

[[29 0 0 0]
[0 28 1 0]
[0 0 29 0]
[0 1 0 28]]

Fig. 14. Confusion matrix (up to epochs 400).

Fig. 14 shows the Confusion Matrix of the classification done with our 29x4=116 samples and upto 400 epochs. The method has been implemented with 2 convolutional layers. From the matrix, the following information is obtained:

i) of the actual 29 Cap samples, our algorithm has predicted that 29 are Cap.

ii) of the actual 29 Impacted samples, our algorithm has predicted that 28 are Impacted and 1 is Incisor.

iii) of the actual 29 Incisor samples, our algorithm has predicted that 29 are Incisor.

iv) of the actual 29 Molar samples, our algorithm has predicted that 28 are Molar and 1 are Impacted.

The Precision, Recall, F1 Score and Accuracy of the four classes are provided in Table III. The Average Precision, Average Recall and Average F1 Score are found to be 98%, 98% and 98% respectively. The Average Accuracy for the classification is found to be **97.92%**.

TABLE III. Performance Analysis for Epochs 400

Category	Precision	Recall	F1	Acc
Cap	1.00	1.00	1.00	100.00
Impacted	0.97	0.97	0.97	96.45
Incisor	0.97	1.00	0.98	98.00
Molar	1.00	0.97	0.98	97.25
Average	0.98	0.98	0.98	**97.92**

Fig. 15 shows the Confusion Matrix of the classification done with our 29x4=116 samples and upto 1000 epochs. The method has been implemented with 2 convolutional layers.

The Precision, Recall, F1 Score and Accuracy of the four classes are provided in Table IV. The Average Precision, Average Recall and Average F1 Score are found to be 90%, 84% and 83% respectively. The Average Accuracy for the classification is found to be **83.64%**. The accuracy drops down as the epochs increases because of overfitting.

```
[ [ 27    0     1     1  ]
  [ 0     27    1     1  ]
  [ 0     0     29    0  ]
  [ 0     1     14    14 ] ]
```

Fig. 15. Confusion matrix (up to epochs 1000).

TABLE IV. Performance Analysis for Epochs 1000

Category	Precision	Recall	F1	Acc
Cap	1.00	0.93	0.96	95.93
Impacted	0.96	0.93	0.95	95.78
Incisor	0.70	1.00	0.82	81.52
Molar	0.88	0.48	0.61	61.27
Average	0.90	0.84	0.83	**83.64**

The algorithm has been implemented with 1 convolutional layer, 2 convolutional layers, 3 convolutional layers and 4 convolutional layers with CNN. As shown in Table V, the highest accuracy is obtained when the CNN is implemented with 2 Convolutional layers.

TABLE V. Loss and Accuracy of the Model With Different Convolutional Layer

No. of Convolutional Layer	Loss	Accuracy	Validation loss	Validation accuracy
1	0.6012	80.12	1.2054	73.78
2	0.1927	97.57	0.1873	97.94
3	0.2931	91.73	0.5774	79.59
4	0.2547	91.84	0.6711	77.87

IV. Comparison

Our method is compared with four standard algorithms and the comparison is shown in Table VI. The details of the four other methods are shown in first four rows and our proposed algorithm in last row.

The method proposed by [31] using CNN gives the classification accuracy of 85.3%.

The method proposed by [32] using Deep Convolutional Neural Networks (DCNNs) gives the Classification accuracy of 0.81(0.02). Mean (SD) sensitivity and specificity was 0.81 and 0.81 respectively. [33] Proposed a method using CNN which gives a classification accuracy rate of 83.0%.

[34] Proposed a method using Deep Convolutional Neural network (DCNN) which gives a classification accuracy of 95.6%.

Our proposed method uses CNN with Dropout and gives a classification accuracy of **97.92**%. Using the Dropout regularization technique, over fitting has been avoided and thereby the training and validation loss has been dropped down to significant level.

V. Conclusion

Over the past few years, the convolutional neural networks (CNNs) model has gain a huge potential in the field of medical imaging technology. In this paper, we have proposed a computer aided system which helps in classifying the type of teeth and also some underlying oral anomalies present in an OPG using CNN with deep learning technique to assist the dentists to have a better judgment in the treatment of any dental abnormalities. The fully connected layer of the CNN which does the classification has been trained using hybrid GA-BP learning. Dropout technique has been used to avoid over fitting thereby improving the performance of the network classification. The proposed algorithm is implemented with different convolutional layers and the model with two convolutional layers gives the highest accuracy of **97.92**%.

Acknowledgement

We are immensely thankful to NIT Manipur for providing a platform to do the research work and Cosmo Dental Clinic Manipur for providing the dataset.

References

[1] A. Qayyum, S.M. Anwar, M. Majid, M. Awais, and M.R. Alnowami, "Medical Image Analysis using Convolutional Neural Networks: A Review," *Journal of Medical Systems*, vol. 42, pp. 1-13, 2018, doi: 10.1007/s10916-018-1088-1.

[2] J. Prada, Y. Gala, and A. L. Sierra, "COVID-19 Mortality Risk Prediction Using X-Ray Images," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 6, no. 6, pp.7-14, 2021, doi:10.9781/ijimai.2021.04.001.

[3] M. I. Khattak, M.'ath. Al-Hasan, A. Jan, N. Saleem, E. Verdú, and N. Khurshid, "Automated Detection of COVID-19 using Chest X-Ray Images and CT Scans through Multilayer- Spatial Convolutional Neural Networks," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 6, no. 6, pp. 15-24, 2021, doi: 10.9781/ijimai.2021.04.002.

[4] S. Kaliyugarasan, A. Lundervold, and A. S. Lundervold, "Pulmonary Nodule Classification in Lung Cancer from 3D Thoracic CT Scans Using fastai and MONAI," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 6, no. 7, pp. 83-89, 2021, doi:10.9781/ijimai.2021.05.002.

[5] B. S. Harish, M. S. Maheshan, and N. Nagadarshan, "A Convolution Neural Network Engine for Sclera Recognition," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 6, no. 1, pp. 78-83, 2020, doi: 10.9781/ijimai.2019.03.006.

[6] J. J. Hwang, Y. H. Jung, B. H. Cho, and M. S. Heo, "An overview of deep learning in the field of dentistry," *Imaging science in dentistry*, vol. 49, no. 1, pp. 1–7, 2019, doi:10.5624/isd.2019.49.1.1.

[7] M. Mupparapu, C. W. Wu, and Y. C. Chen, "Artificial intelligence, machine learning, neural networks, and deep learning: Futuristic concepts for new dental diagnosis," *Quintessence international*, vol. 49, no. 9, pp. 687–688, 2018, doi:10.3290/j.qi.a41107.

TABLE VI. Comparison of Proposed Method With Other Standard Methods

Researchers	Year	Methods	Types of Radiography	Teeth Images class	Classification Accuracy
Kuo[31]	2017	CNN	Panoramic radiograph	Dental Anomalies	85.3%
Krois[32]	2019	Deep Convolutional Neural Network(DCNN)	Panoramic radiograph	Dental Anomalies	Classification accuracy was 0.81(0.02). Mean(SD) sensitivity and specificity was 0.81 and 0.81
Poedjiastoeti [33]	2018	CNN	Panoramic radiograph	Dental Anomalies	Sensitivity, specificity, accuracy were 81.8%, 83.3%, 83.0%.
Singh[34]	2020	CNN	Panoramic radiograph	Dental types	95.6%
Our proposed Method	2020	CNN with Dropout	Orthopantomogram Images (panoramic radiograph)	Dental types and Anomalies	**97.92**%

[8] S.B. Khanagar et al., “Developments, application, and performance of artificial intelligence in dentistry-A systematic review,” *Journal of Dental Sciences*, vol. 16, pp. 508-522, 2021, doi: 10.1016/j.jds.2020.06.019.

[9] J. Yang, Y. Xie, L. Liu, B. Xia, Z. Cao, and C. Guo, “Automated Dental Image Analysis by Deep Learning on Small Dataset,” *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, Tokyo, 2018, vol.1, pp. 492–497, doi:10.1109/COMPSAC.2018.00076.

[10] N. Karimian, H. S. Salehi, M. Mahdian, H. Alnajjar, and A. Tadinada, “Deep learning classifier with optical coherence tomography images for early dental caries detection,” *Proc. SPIE 10473, Lasers in Dentistry XXIV*, 1047304, pp. 10-17, 2018, doi:10.1117/12.2291088.

[11] K.L. Devito, F.D. Barbosa, and W.N. Filho, “An artificial multilayer perceptron neural network for diagnosis of proximal dental caries,” *Oral surgery, oral medicine, oral pathology, oral radiology, and endodontics*, vol. 106, pp. 879–884, 2008, doi:10.1016/j.tripleo.2008.03.002

[12] R.B. Ali, R. Ejbali, and M. Zaied, “Detection and classification of dental caries in X-ray images using deep neural networks,” *In: International Conference on Software Engineering Advances (ICSEA).* 2016, pp. 223-227.

[13] S. Oprea, C. Marinescu, I. Lita, M. Jurianu, D. A. Visan, and I. B. Cioc, “Image processing techniques used for dental x-ray image analysis,” In *2008 31st International Spring Seminar on Electronics Technology*, 2008, pp.125-129, doi:10.1109/ISSE.2008.5276424.

[14] S. A. Prajapati, R. Nagaraj, and S. Mitra, “Classification of dental diseases using CNN and transfer learning,” In *5th International Symposium on Computational and Business Intelligence (ISCBI)*, Dubai, 2017, pp. 70-74, doi:10.1109/ISCBI.2017.8053547.

[15] Y. Yu, “Machine Learning for Dental Image Analysis,” *ArXiv*, vol. abs/1611.09958, 2016, https://arxiv.org/abs/1611.09958.

[16] M. M. Srivastava, P. Kumar, L. Pradhan, and S. Varadarajan, “Detection of tooth caries in bitewing radiographs using deep learning,” *ArXiv*, vol. abs/1711.07312, 2017, https://arxiv.org/abs/1711.07312 (accessed 16 May 2019).

[17] D. K. Veena, A. Jatti, R. Joshi, and K. S. Deepu, “Characterization of dental pathologies using digital panoramic X-ray images based on texture analysis,” In *39th Annual International Conference of IEEE Engineering in Medicine and Biology Society (EMBC)*, 2017, pp. 592-595, doi:10.1109/EMBC.2017.8036894.

[18] A. Zakirov, M. Ezhov, M. Gusarev, V. Alexandrovsky, and E. Shumilov, “Dental pathology detection in 3D cone-beam CT,” *ArXiv*, vol. abs/1810.10309, 2018.

[19] R.C. Gonzales, and R. E. Woods, “Digital Image Processing,” *Prentice Hall, Inc, New Jersey*, 2008, ISBN: 9780131687288 013168728X 9780135052679 013505267X.

[20] Y.M. Abdallah, and T. Alqahtani, “Research in Medical Imaging Using Image Processing Techniques,” *Medical Imaging - Principles and Applications,* IntechOpen, 2019, doi:10.5772/intechopen.84360.

[21] A. S. Unde, V. A. Premprakash, and P. Sankaran, “A novel edge detection approach on active contour for tumor segmentation,” In *2012 Students Conference on Engineering and Systems,* Allahabad, Uttar Pradesh, 2012, pp. 1-6, doi:10.1109/SCES.2012.6199100.

[22] J. Wu, “Introduction to Convolutional Neural Networks,” *National Key Lab for Novel Software Technology*, Nanjing University, China, 2017.

[23] T. Wiatowski, and H. Bolcskei, “A mathematical theory of deep convolutional neural networks for feature extraction,” in *IEEE Transactions on Information Theory*, vol. 64, no. 3, pp. 1845-1866, March 2018, doi:10.1109/TIT.2017.2776228.

[24] A. Suruliandi, A. Kasthuri, and S. P. Raja, “Deep Feature Representation and Similarity Matrix based Noise Label Refinement Method for Efficient Face Annotation,” *International Journal of Interactive Multimedia and Artificial Intelligence*, In Press (In Press), 1-12, http://doi.org/10.9781/ijimai.2021.05.001.

[25] Y. Li, J. Deng, Q. Wu, and Y. Wang, “ Eye-Tracking Signals Based Affective Classification Employing Deep Gradient Convolutional Neural Networks,” *International Journal of Interactive Multimedia And Artificial Intelligence*, In Press (In Press), pp. 1-10, doi: 10.9781/ijimai.2021.06.002.

[26] K. K. Verma, B. M. Singh, H. L. Mandoria, and P. Chauhan, “Two-Stage Human Activity Recognition Using 2D-ConvNet,” *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 6, no. 2, pp. 125-135, 2020, doi: 10.9781/ijimai.2020.04.002.

[27] X. Jin, Y. Xiao, S. Li, and S. Wang, “Deep Learning-based Side Channel Attack on HMAC SM3,” *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 6, no. 4, pp. 113-120, 2020, doi: 10.9781/ijimai.2020.11.007.

[28] A. Ghazvini, S. N. H. S. Abdullah, and M. Ayob “A Recent Trend in Individual Counting Approach Using Deep Network,” *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 5, no. 5, pp. 7-14, 2019, doi:10.9781/ijimai.2019.04.003.

[29] M.D. Bloiceet, C. Stocker, and A. Holzinger, “Augmentor: An Image Augmentation Library for Machine Learning,” *The Journal of Open Source Software*, vol. 2, no. 19, 2017, doi:10.21105/joss.00432.

[30] J. Wang and L. Perez, “The effectiveness of data augmentation in image classification using deep learning,” *Convolutional Neural Networks Vis. Recognit*, vol. 11, pp.1–8, 2017.

[31] Y.F. Kuo, S.Y. Lin, C.H. Wu, S.L. Chen, T. Lin, N.H. Lin, C.H. Mai, and J.F. Villaverde, “A Convolutional Neural Network Approach for Dental Panoramic Radiographs Classification,” *Journal of Medical Imaging and Health Informatics*, vol. 7, pp. 1693-1704, 2017, doi:10.1166/JMIHI.2017.2257.

[32] J. Krois, T. Ekert, L. Meinhold, T. Golla, B. kharbot , A. Wittemeier, C. Doofer, and F. Schwendicke, “Deep Learning for the Radiographic Detection of Periodontal Bone Loss,” *Scientific Reports*, vol. 9, 8495, 2019, doi: 10.1038/s41598-019-44839-3.

[33] W. Poedjiastoeti, and S. Suebnukarn, “Application of Convolutional Neural Network in the Diagnosis of Jaw Tumors,” *Healthcare Informatics Research,* vol. 24, pp. 236-241, 2018, doi: 10.4258/hir.2018.24.3.236.

[34] P. Singh, and P. Sehgal, “Numbering and Classification of Panoramic Dental Images Using 6-Layer Convolutional Neural Network,” *Pattern Recognition and Image Analysis*, vol.30, pp. 125–133, 2020, doi: 10.1134/S1054661820010149.

Anuradha Laishram

Anuradha Laishram received her B.Tech from Visvesvaraya Technological University, India in 2008 and M.E in Computer science and Engineering from pursuing her Ph.D degree from National Institute of Technology Manipur (NIT Manipur), India. She is presently working as Lecturer in NIT Manipur, India. Her main research interest includes Machine Learning, Deep Learning, Hybrid Intelligent System, and Medical Image Processing. Email: annu0286@gmail.com

Khelchandra Thonga

Khelchandra Thongam received his M.S degree in Computer science and Engineering from The University of Aizu, Japan in 2007 and his Ph.D also from The University of Aizu, Japan. He is currently working as Assistant Professor in National Institute of Technology Manipur (NIT Manipur), India. His main research interest includes Machine Learning, Soft Computing, Hybrid Intelligent System, Medical Image Processing, Speech Processing, and Mobile Robot Navigation. Email: thongam@gmail.com

CDPS-IoT: Cardiovascular Disease Prediction System Based on IoT Using Machine Learning

Jameel Ahamed[1], Abdul Manan Koli[1], Khaleel Ahmad[1]*, Mohd. Alam Jamal[1], B. B. Gupta[2]*

[1] Department of Computer Science & Information Technology, Maulana Azad National Urdu University, Gachibowli, Hyderabad (India)

[2] Department of Computer Engineering, National Institute of Technology Kurukshetra, India & Asia University (Taiwan)

Received 23 January 2021 | Accepted 18 June 2021 | Published 7 September 2021

Abstract

Internet of Things, Machine learning, and Cloud computing are the emerging domains of information communication and technology. These techniques can help to save the life of millions in the medical assisted environment and can be utilized in health-care system where health expertise is less available. Fast food consumption increased from the past few decades, which makes up cholesterol, diabetes, and many more problems that affect the heart and other organs of the body. Changing lifestyle is another parameter that results in health issues including cardio-vascular diseases. Affirming to the World Health Organization, the cardiovascular diseases, or heart diseases lead to more death than any other disease globally. The objective of this research is to analyze the available data pertaining to cardiovascular diseases for prediction of heart diseases at an earlier stage to prevent it from occurring. The dataset of heart disease patients was taken from Jammu and Kashmir, India and stored over the cloud. Stored data is then pre-processed and further analyzed using machine learning techniques for the prediction of heart diseases. The analysis of the dataset using numerous machines learning techniques like Random Forest, Decision Tree, Naive based, K-nearest neighbors, and Support Vector Machine revealed the performance metrics (F1 Score, Precision and Recall) for all the techniques which shows that Naive Bayes is better without parameter tuning while Random Forest algorithm proved as the best technique with hyperparameter tuning. In this paper, the proposed model is developed in such a systematic way that the clinical data can be obtained through the use of IoT with the help of available medical sensors to predict cardiovascular diseases on a real-time basis.

Keywords

Cardiovascular Diseases, Cloud Computing, Internet Of Things, Machine Learning.

DOI: 10.9781/ijimai.2021.09.002

I. Introduction

Since the last few decades, the World witnessed an increase in the death rate due to cardiovascular diseases (CVD). In the United States of America, one person is dying every minute due to the arrest of CVD. Numerous researchers had tried the classification techniques of machine learning to diagnose CVDs to help medical practitioners in improving the health care system world-widely. According to the World Health Organization (WHO), an estimation of 30% of global deaths is caused by CVDs over three-quarters of which catch a spot in low and middle-income countries. CVDs also killed 25% of the population in the age group of 25-69 years in India[1]. Internet of Things (IoT) is usually referred to as physical things connected to the internet world with limited storage and processing ability. It is still struggling with performance, interoperability, security, and privacy issues challenges with a huge scope for improvement in the near future [2], [3], [4]. IoT is also proven as a benchmark for healthcare systems wherein smart objects are used to continuously monitor the patient for particular diseases [2]. Smart objects comprise biomedical sensors which gather health-related information and pass it to the physician through cloud/edge for further diagnosis. Hence, IoT helps in bridging the gap between patient and Physician located at any geographical location [5], [6]. Cloud computing has the nearly endless capacity of processing power and storage capacity, which is a more advanced technology to a specific extent to resolve the technical issues in the IoT while data mining is an intelligent technology that is used to extract new information by examining huge voluminous datasets [7]. It can be utilized to make certain decisions, estimates, and predictions using different machine learning algorithms. In the present era, most of the data in the medical sector are collected via the computerized system but not utilized world-wide for analysis. It is stacked up in a database like old handwritten records and put to no use. This data can be harnessed to predict diseases such as Cancer, CVDs, Diabetes, Dengue, etc., [2], [8]. Thus, an innovative Information technology (IT) paradigm is proposed, in which IoT and machine learning with cloud computing are the three interrelated technologies integrated together to overcome both current and future world obstacles related to the health-care system and termed as IoT-ML-Cloud paradigm. Medical assisted technologies and healthcare services are nearly related and remedial to the public welfare for better health-care facilities. The collaboration of cloud computing and IoT for the applications of

* Corresponding author.

E-mail addresses: gupta.brij@gmail.com (B. B. Gupta), khaleelahmad@manuu.edu.in (K. Ahmad).

modern medical assisted technologies play an important role in the prediction of chronic diseases. The development of public cloud (cloud computing) in hospitals, which has benefits like high security, improved efficiency, virtualization, reliability, and scalability can promote resource sharing, cost savings, medical monitoring, management, and administration system with high efficiency and accuracy.

II. Related Work

Many research studies have been done for the prediction of diseases using artificial intelligence, machine learning algorithms, IoT, etc. In paper [9], the authors proposed a healthcare monitoring system using a random forest algorithm assisted by the IoT. In this paper, different diseases like Heart diseases, Diabetes, and Breast Cancer were predicted and achieved a maximum accuracy of 97.26% on the Dermatology dataset using the Random Forest algorithm. In paper [10], the authors proposed a hybrid recommender system for cardiovascular diseases using IoT in a cloud environment and also introduced the multiclass classification problem which can predict the eight types of cardiovascular diseases [11] [12]. Further, the model improved accuracy by 98% using the feature selection techniques. In paper [11], the authors worked on lung cancer and built an IoT-based prediction system using Fuzzy Cluster-Based Segmentation and Classification. The proposed system was developed in a MATLAB environment that focused to classify the lung X-ray images. The proposed model got 85% accuracy. In paper [13], the authors developed an online web-based platform for Clinical Decision Support System (CDSS) which is based on Optimal Deep Neural Networks (DNN). They developed a Cloud-based CDSS framework for the prediction of Chronic Kidney Diseases (CKD) with their level of severity. They employed IoT sensors to get the clinical data from disease-affected patients and predict the condition of normal & abnormal and achieved the highest accuracy of 98% using DNN. In paper [14], the researchers built an android-based monitoring system that can monitor the heart rate of cardiac patients. Decision Tree algorithm was used to develop the model which can trigger an alarm if a patient has an abnormal heart rate. In the paper [15], the authors proposed an IoT-based system that is used to early detect cardiac disease using machine learning algorithms. In this work, they took heart data from the MAX30100 Sensor connected to Arduino and sent the observed data to Oxywatch for further analysis. In paper [16], the researchers analyzed IoT safety, security, and privacy feature including the requirements of a security issue, use patterns, and attack taxonomies from the health care prospect. They also invented a wearable health care environment to decide how people get facilities by economic and societies in terms of sustainability for handling noisy, missing values using Decision tree-based C4.5 classifier. In the paper [17], the authors used some statistical methods like Nearest Neighbor Imputation, Hot Deck imputation, Cold Deck Imputation, Substitution, and Mean Substitution for prediction. In [18], the authors worked with 6 predictive state imputations, the frequency-based imputation used by the C4.5 algorithms, and reduces modeling for a classification tree. In the paper [19], the authors suggested the approach of only numerical values to assign the disappeared values that can able to handle categorical attributes, and in last, they compare with the different factors i.e. cost, space, time, etc., which further helps in analyzing the data [19] [20]. In the paper [21], the researchers performed the comparison between decision trees, k-nearest neighbors' algorithm (K-NN), and Naïve Bayes algorithm using Rapid miner, WEKA, Tanagra, Knime, and Orange tools on Dataset of Liver Patients [22]. In the paper [23], the authors proposed a system to deal with missing values in test datasets and training datasets for predicting diabetic disease in patients. Paper [24] showed a comparison of different machine learning algorithms with the help of RapidMiner, Tanagra, WEKA tool, Knime and Orange. In paper [25], the diabetic dataset was analyzed with Classification and Regression Trees (CART), Support Vector Machine (SVM), K-NN, J48 algorithms for finding classifications. The accuracy rate of K-NN was 53.39% for correct classification and incorrect classification was 46.605%.

III. Findings From Related Work

We did not find any reliable model which can accurately predict cardiovascular disease in developing nations. Most of the researchers developed the cardiovascular disease models based on single and two algorithm combinations which are not strong enough to make accurate predictions in developing nations. To overcome these research gaps, we performed the following:

I. We used three different feature selections and also the five classifiers for cardiovascular disease prediction. Till date, we have not found any research work which utilized such classifiers for feature selections and cardiovascular disease predictions. Mostly, research scholars used only one or two algorithms in their research work.

II. We did not find any research work pertaining to cardiovascular disease prediction based upon clinical data for the area of Jammu and Kashmir (India) using machine learning techniques, therefore, this research work may be considered as the first cardiovascular disease prediction model for such developing areas.

III. We applied machine learning techniques for cardiovascular disease because it is considered to be one of the best and most innovative techniques for prediction work and is also used for other disciplines like cancer disease prediction, stock market predictions and weather forecasting, etc.

IV. Machine Learning Algorithms

In this research, five types of machine learning algorithms are used for the development of the disease prediction model. The algorithms for disease prediction model are k-Nearest Neighbor (KNN), Decision Tree classifier (DTC), Support Vector Machine (SVM), Random Forest (RF), and Naïve Bayes (NB). The main aim of these algorithms is classification. As our dataset has an output class label, hence these supervised algorithms have been chosen because they work with class label problems.

A. K-Nearest Neighbors

K-Nearest Neighbors is the most simple classifier that stores all the available cases and classifies new instances based on nearest majority neighbors (for example distance function) [26]. K-NN has been adopted in statistical estimation and pattern recognition was already considered as a non-parametric tool. K-NN assumes that identical classes exist in shut proximity i.e entities that are similar, exist together. In the name of the K-NN algorithm, the alphabet 'K' means the number of nearest neighbors for determining the class of an instance, as shown in Fig. 1 [9].

In this algorithm, two parameters are obligatory that is neighborhood cardinality (K), and measure to evaluate the similarity. Mathematically, we can calculate the Euclidean distance between two points. The Euclidean distance equation is given below.

$$\rho(x, x') = |x - x'| = \sqrt{\sum_{k=0}^{n} (xi - xi')^2} \tag{1}$$

K represents the number of the data point in the KNN algorithm that is close to the new data point [11]. For instance, if K = 1, 2, 3 then it will choose one nearest neighbor, two nearest neighbors three nearest neighbor data points respectively.

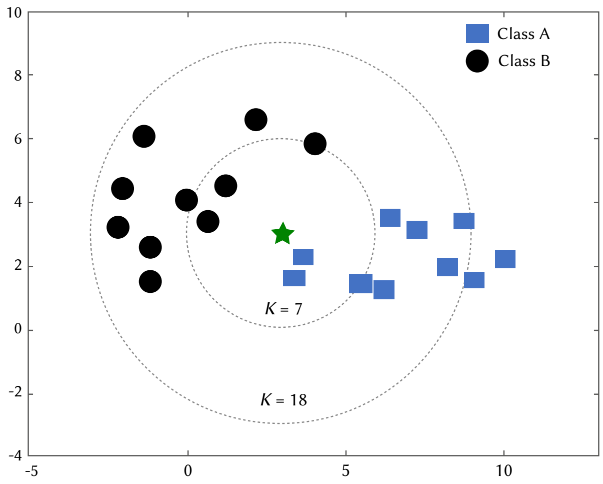


Fig. 1. K-Nearest Neighbors.

Now, it will classify the data point based on the majority of voting [27]. This idea is portrayed in Fig. 1. After classification, the new data point is represented by the green star point using the nearest neighbor technique, as depicted in Fig. 1. Here two classes A and B which is represented by a sky color rectangle and black color circle respectively. For K = 7, the star is close to the rectangle, hence the KNN algorithm classified it as a class A.

Before applying KNN, data should be cleaned and no outlier or noise value should be presented in the dataset. And also, the variable should be normalized otherwise the higher value variables can lead the model bias.

B. Random Forest

Random forest is a classification algorithm and it constructs multiple decision trees during the training phase[13]. The random forest algorithm takes a final decision to choose the trees based on the majority of voting [14]. It reduces the risk of overfitting of the model based on the utilization of multiple trees [10]. It works efficiently on a large database and produces highly accurate results [27].

Steps of Random Forest algorithm:

- Pick random R data objects from the training dataset.
- Create a DT (Decision Tree) for the K data point.
- Picks the n-tree subset from the newly generated trees and execute step 1 and step2
- Take the decision or result based on the majority of voters.'

Fig. 2 portrays the working of the Random Forest algorithm. We made three different Decision trees from the dataset, but the final classification algorithm takes decisions based upon the majority of voters.

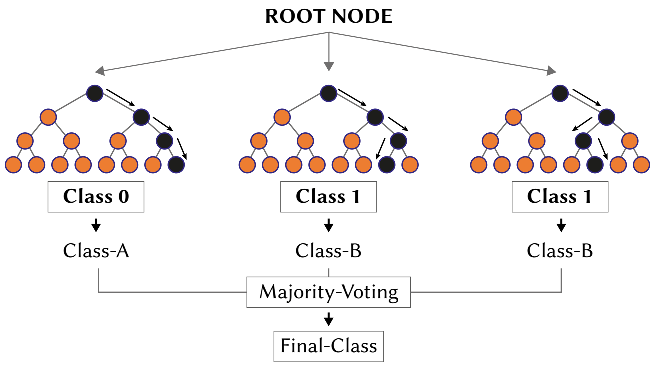


Fig. 2. Random Forest Algorithm.

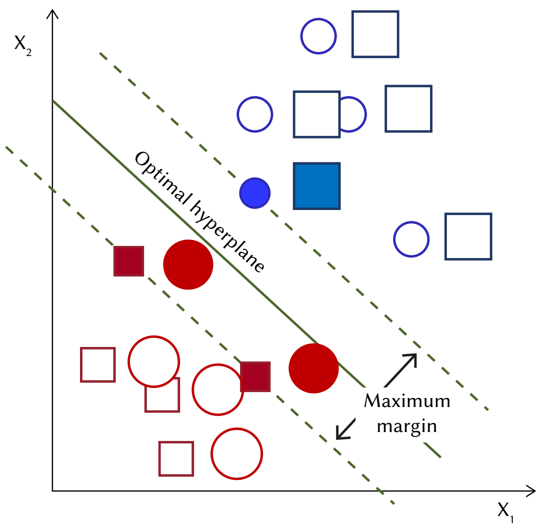


Fig. 3. Support Vector Machine.

C. Support Vector Machine (SVM)

SVM is a classifier or a classification algorithm that uses the hyperplane to classify the data [15]. The primary objective of SVM has to find the best and optimal hyper-plane in n-dimension space because it requires such type of hyper-plane which has a maximum margin [16]. Hence, the thumb rule is choosing the hyper-plane, which separates the two classes better.

In Fig. 3, the circle and rectangle are two classes and separated from each other by hyper-plane. But the queries raised here that how many hyper-planes are possible and which one has to choose out of possible hyper-planes. It will be chosen based upon the maximum margin between the classes. For the non-linear data points, SVM uses the kernel trick to draw the hyper-plane between them. Hence, it can say that SVM also works for non-linear data points [27].

D. Decision Tree

A decision tree can be used for both regression and classification problems. It is a non-parametric supervised learning method. It is a tree-structured classifier where the characteristic of a dataset is defined by internal nodes and branches of the tree represent the decision rules and the outcome is defined by each leaf node [28].

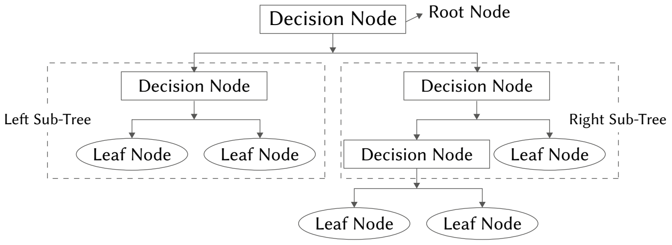


Fig. 4. Decision Tree.

There are two nodes in the Decision tree, which are the leaf node and the decision node. Decision nodes have several branches and are used to make some decisions while leaf nodes are the outcome of such decisions and do not have any further branches. A test or decision is carried out based on features of the dataset.

A decision tree asks a question, and it further partitions the tree into sub-trees based on the appropriate response (Yes / No), as portrayed in Fig. 4. A decision tree includes categorical data (YES/NO) as well as numeric data [29].

V. Primitive Statistics

The aim of this research work has to predict heart diseases which is a very delicate factor and risky as well. If done properly, it can be used by the medical administration for the proper diagnosis of patients who are suffering from CVDs. The dataset is obtained from Jammu and Kashmir (India), which is a real-world dataset. Thereafter, a combination of five ML-based classification algorithms used viz-a-viz Naïve Bayes, K-Nearest Neighbors, Decision tree, Random Forest, and Support Vector Machine.

The heart disease prediction model was developed using the anaconda tools which can handle large datasets with Jupyter notebook in python language. The developed model is then validated by medical domain experts.

To initialize the development of the model following suitable steps were performed:

A. Data Collection

This is the first step of model development in this step's dataset related to cardiovascular disease is collected from Jammu and Kashmir (India). Information and description of the dataset are given in Table I.

TABLE I. Attributes Information and Description of Dataset

S.No.	Attribute	Description
1	Age	Age in Years
2	Sex	Gender
3	trestbps	Resting blood pressure (in mm Hg
4	chol	serum cholesterol in mg/dl
5	fbs	fasting blood sugar > 120 mg/dl
6	restecg	resting electrocardiographic results
7	thalac	maximum heart rate achieved
8	Alcoholic	Alcohol Consumption
9	oldpeak	ST depression induced by exercise relative to rest
10	Ca	number of major vessels (0-3) colored by fluoroscopy
11	Slope_0	the slope of the peak exercise ST segment
12	Stress_0	Stress level
13	Thal_0	Thalassemia

B. Data Preprocessing:

In this step, the collected data is preprocessed using various methods like mean, mode, and median.

C. Feature/Attributes Selection Techniques:

After the dataset is cleaned and is ready for the mined purpose. We applied various attributes/features selection algorithms like Embedded Methods, Wrapper Methods, and Filter Methods. Then we take the mean of all the utilized methods and based upon the obtained mean we observed the weightage of each attribute.

D. Classification Algorithms:

Supervised learning algorithms like SVM, k-NN, Decision Tree, etc. were employed for the development of the proposed model.

VI. Experimentation

The proposed heart disease prediction model based on IoT-Cloud-ML consists of four-tier architecture starting from data collection, data storage, data pre-processing & analysis, and development of Graphical User Interface (GUI). Tier 1 focused on collecting health care data using IoT or wearable devices. Tier 2 and Tier 3 focused on cloud computing where data storage service and machine learning models are deployed. Tier 4 focuses on the front-end graphical user interface from where a patient or a doctor can view and interpret the results.

Tier 1 – Data Collection: Data collection tier is employed for collecting the patient's data using different IoT medical sensors and wearable devices. These IoT-based healthcare devices attached to the patient's body continuously collect clinical data. These devices then transmit the data to the cloud using various communication technologies like Wi-Fi, ZigBee, GPRS/LTE. Algorithm 1 shows IoT device initialization, data collection in a continuous manner, and transmission of data to the cloud database. In this system, we used 'THINGSPEAK Cloud' to collect the medical sensor data with the help of the NODEMCU server. The data which we used in our work was taken from Jammu and Kashmir as a repository system. Further data collected from IoT-based devices will be employed in this model for real-time prediction [30].

Algorithm 1: IoT device initialization and sending data to the cloud

```
Step 1: attach the IoT medical sensors in patients' body and Start Device Initialization
Step 2: configure the nodemcu with ssd and password and connect with wifi
Step3 : attach nodemcu with sensor to work as a mediator
If(nodemcu==connectedwithwifi)
{
    If(sensor==attachedwithnodemcu)
    {
        Read data and send it to the cloud
    }
    Else
    {
        Error: Sensor is not attached
    }

} else {
    Error: Nodemcu is not connected
}
```

Tier 2- Data Storage: IoT devices employed in Healthcare, Industrial IoT have the objective of sensing patient clinical data and transmit it to data storage service in a continuous manner [31], [32], [33]. But it is a very complicated task to sense and send such data by conventional data processing tools and techniques. IoT devices and physical personal computers do not have sufficient storage to store and process the voluminous clinical data of cardiovascular patients etc., [34]. Hence, the proposed method used cloud computing for this task. In the proposed model, HEROKU cloud PostgreSQL is used for storing and pre-processing the data. HEROKU PostgreSQL cloud is a data storage service widely used for data storage as described in Algorithm 2.

Algorithm 2: Storage of Patient Data onto Heroku PostgreSQL database

```
Step 1: initialize PostgreSQL in Heroku cloud
Step 2: setup PostgreSQL with user name and password
Step 3: if the database and table not created Then
create a database and table.
Step 4: configure IOT sensor data by using a thing speak write API key to insert
Step5: if (thingspeakDiseaseData==True or PatientPersonalData==True)
{
    Insert data into the table
}
Else {
    Error: no data to insert}
```

Tier 3 – Data Analysis on a cloud: Data analytics tier uses the machine learning models and techniques to predict the output class label. The proposed methodology implemented various types of machine learning algorithms and chose one of them which gives the best and accurate results. The dataset is divided into training and testing datasets. The trained data was treated with different classification algorithms like Decision Tree, SVM, KNN, Random Forest, and Naive based to further test the model for prediction of cardiovascular diseases as described in Algorithm 3.

Algorithm 3: Predication of the diseases using the proposed model

```
Step 1: select the patient of which we need to predict the disease
Step 2: check the patient data and its attribute
If(Any Attribute==null)
{
insert the required data into the database
}
Else {
Predict the disease and show the result}
```

Tier 4 - GUI on a cloud: This tier provides the frontend interface which can be accessed by both patient and the doctor (expert). The patient can view their clinical results using patient_id. This GUI was developed using the python flask library. The flask application is deployed on the cloud so that it can easily communicate with other services on the cloud.

Fig. 5 shows the overall proposed system composed of a four-tier system and able to communicate with each other by integrating Cloud computing, Machine Learning, and the Internet of things in a precise manner.

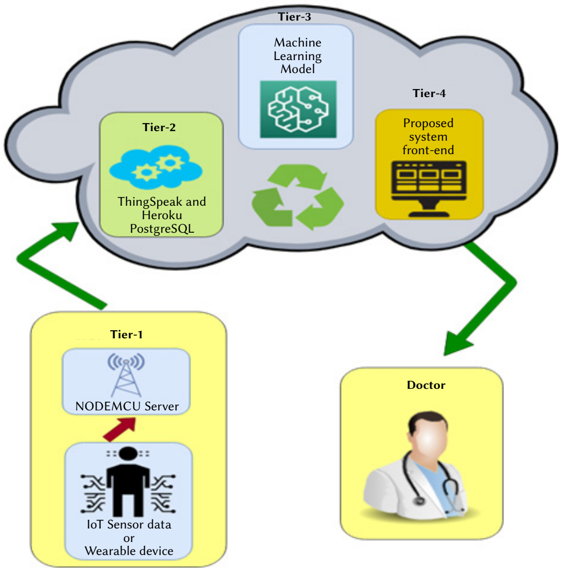

Tier-3
Machine Learning Model
Tier-4
Proposed system front-end
Tier-2
ThingSpeak and Heroku PostgreSQL
Tier-1
NODEMCU Server
IoT Sensor data or Wearable device
Doctor

Fig. 5. Proposed Methodology Diagram.

VII. Results and Discussion

In this section, the authors applied different feature selection techniques like filter method, embedded method, and wrapper methods, and finally, their mean weightage is obtained for all attributes for their use in the development of the model, which are depicted in Table II and Fig. 6.

The results of the developed model are described in this section. From Fig. 6, it is clear that the persons having a high old peak has more chance of getting heart disease as compared to the Alcoholic person which has less chance of having heart disease based on weightage. In this way, the weightage scale can be analyzed.

TABLE II. Attributes Along With Their Mean Weightage

Feature Selection techniques	Filter Method	Embedded Method	Wrapper Method	Mean
age	0.98	0.18	0.92	0.69
sex	0.81	0.82	0.50	0.71
trestbp	0.99	0.20	0.75	0.65
chol	1.00	0.18	1.00	0.73
fbs	1.00	0.13	0.00	0.38
restecg	0.97	0.00	0.25	0.41
thalach	0.93	0.11	0.83	0.62
Alcoholic	0.08	0.73	0.33	0.38
oldpeak	0.73	0.90	0.67	0.77
Ca	0.61	0.97	0.58	0.72
Slope_0	0.98	0.14	0.17	0.43
Stress_0	0.00	1.00	0.42	0.47
Thal_0	1.00	0.18	0.08	0.42

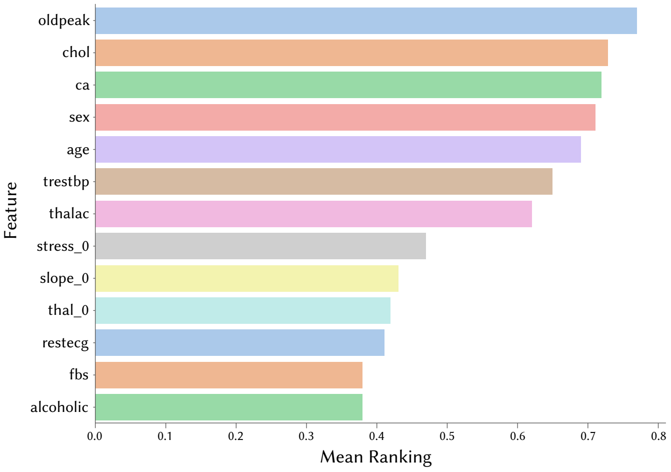


Fig. 6. Attributes along with their Weightage.

After obtaining the mean weightage of all the attributes, various Machine learning classifiers like Decision tree (DTC), Random Forest search (RFS), K-nearest neighbor (KNN), Support vector machine (SVM), and Naïve Bayes (NB) are utilized in various situation and tested the performance of each classifier.

The confusion matrix along with the AUROC curve of a Decision tree is shown in Fig. 7.

Fig. 7 depicted the confusion matrix and AUROC curve for the decision tree, and its use to measure the performance of the classifier in terms of Accuracy, Precision, Recall, and F1-Score. Firstly, the Decision tree algorithm was applied to the test data set and found 83.43% accuracy with 84.38% precision, 81.63% recall, and 82.98% F1-Score. It means that the decision tree model has the capability of differentiating the person having heart disease or not is 83.43%.

The confusion matrix along with the AUROC curve of the Random forest Model is shown in Fig. 8.

Fig. 8 depicts the confusion matrix along with the AUROC curve for the Random Forest model. After applying Random Forest Algorithm on the test dataset, it was found that the (RF) model has 87.72% accuracy with 86.16% precision, 89.59% recall, and 87.84% F1-Score. It means that the Random Forest model has the capability of differentiating the person having or not having heart disease is 86.16%.

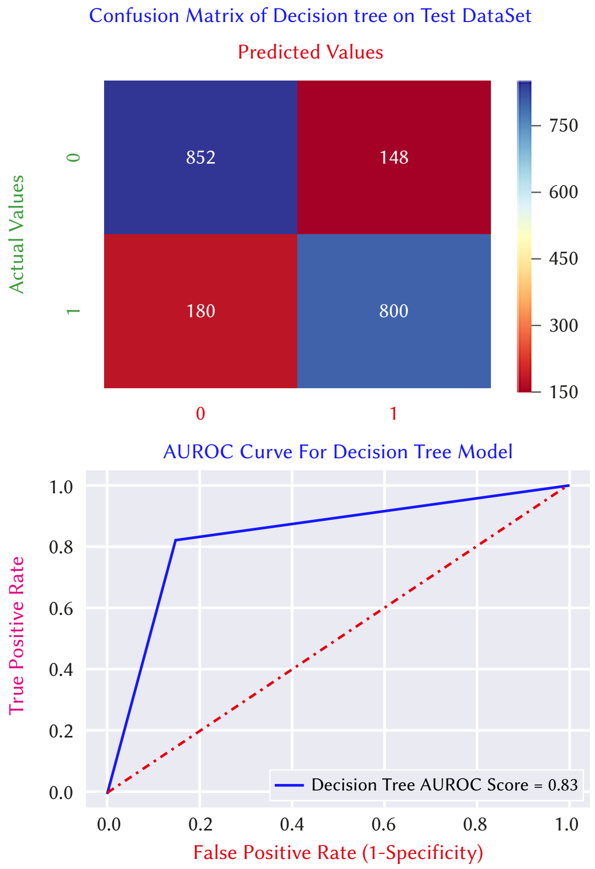


Fig. 7. Confusion matrix and AUROC curve of Decision Tree Model.

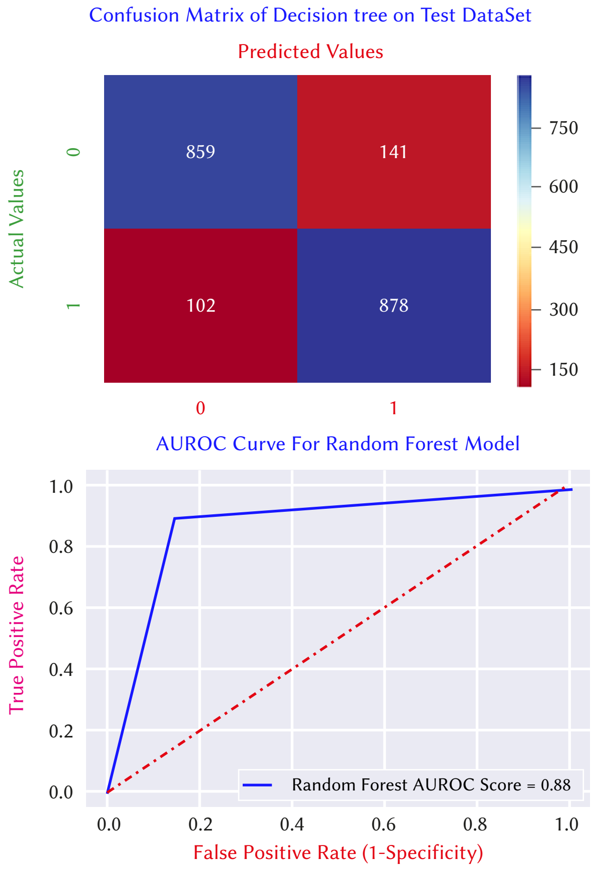


Fig. 8. Confusion matrix and AUROC curve of Random Forest Model.

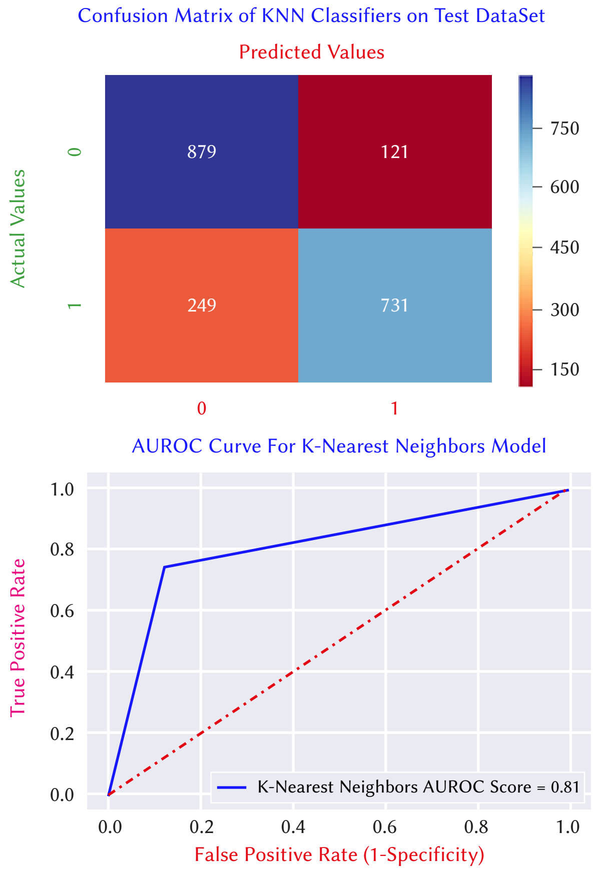


Fig. 9. Confusion matrix and AUROC curve of K-Nearest Neighbor Model.

The confusion matrix along with the AUROC curve of the K-Nearest Neighbors algorithm is shown in Fig. 9.

Then, the K-Nearest Neighbors algorithm was applied and found 81.31% of accuracy on the test dataset with 85.79% of precision, 74.59% of recall, and 79.80% of F1-Score. It means that the K-Nearest Neighbor model has the capability of differentiating the person having or not having heart disease is 81.31%. The confusion matrix along with the AUROC curve of the K-Nearest Neighbor model is shown in Fig. 9.

The confusion matrix along with the AUROC curve of the Support Vector Machine model is shown in Fig. 10.

Fig. 10 shows the confusion matrix along with the AUROC curve of the Support Vector Machine model. The obtained results of (SVM) are much better than KNN, in which the accuracy is 85.40% with precision 84.24%, Recall 86.73% and F1-Score is 85.47% respectively. It means that the Support Vector Machine model has the capability of differentiating the person having or not having heart disease is 85.40%.

The confusion matrix along with the AUROC curve of the Naïve Bayes model is shown in Fig. 11.

Finally, a Naïve Bayes classifier was applied and found 82.62% of accuracy without hyperparameter tuning. It means that the Naïve Bayes model has the capability of differentiating the person having or not having heart disease is 82.62%. The obtained precision of Naïve Bayes is 80.57%, with Recall and F1-Score85.51% and 82.97% respectively. The confusion matrix along with AUROC is shown in Fig. 11. Overall results of all models are shown in Table III.

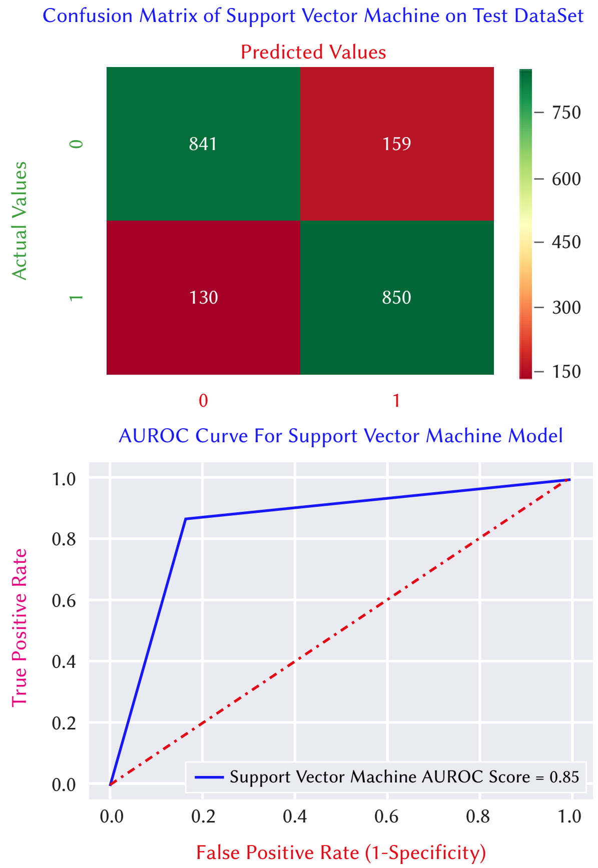


Fig. 10. Confusion matrix and AUROC curve of Support Vector Machine Model.

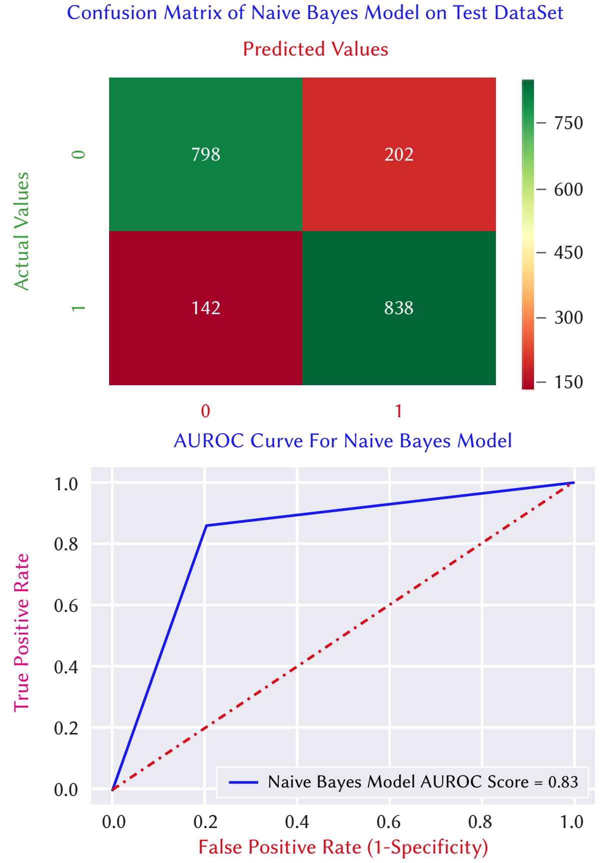


Fig. 11. Confusion matrix and AUROC curve of Naïve Bayes Model.

For the refinement of results, hyperparameter tuning was applied which uses the Grid Search method of Scikit-Learn. The conventional approach of performing hyperparameter optimization has been Grid search, or a parameter sweep, which is simply a comprehensive search of candidate parameter values over all appropriate values in the specified search space [19]. After all possible parameter combinations which are calculated for a model; the finest combination will be retained. Grid search gives the training to an algorithm for all possible combinations using two hyperparameters sets (learning rate and several layers) and measures the performance using a cross-validation technique [21]. These validation techniques ensured that the trained model gets most of the patterns from the dataset. Grid Search uses a manual method to apply the parameter to the algorithm [23]. Naïve Bayes does not support Grid Search because it doesn't have any parameter. Hence, we applied Parameter Tuning on the other algorithms.

TABLE III. Performance Comparison of Different Proposed Heart Disease Prediction Models

Performance Metrics	Comparison of Heart Disease Prediction Models				
	Decision Tree	Random Forest	KNN	SVM	Naïve Bayes
AUROC	0.834163	0.877459	0.812459	0.854173	0.826551
F-1 Score	0.829876	0.878439	0.799035	0.854701	0.829703
Accuracy	0.834343	0.877273	0.813131	0.854040	0.826263
Recall	0.816327	0.895918	0.745918	0.867347	0.855102
Precision	0.843882	0.861629	0.857981	0.842418	0.805769
Error Rate	0.165657	0.122727	0.186869	0.145960	0.173737

From the above results, we came to the conclusion that without parameter tuning of Naïve Bayes performed better which gave 82.63% accuracy, and with hyperparameter tuning, the Random Forest search gave 87.72% accuracy which is higher than all other proposed classifiers. So, for this research work, the Random Forest model is the best for implementation and model building.

The experimental result is shown in Table III which shows that the Random Forest model outperforms other proposed models. So, for this research work the Hyperparameter Random Forest model is selected for making final outputs because of its higher accuracy and lesser error rates. The performance of the proposed model is tested and compared with existing models that portray very promising results.

VIII. Conclusion and Future Work

The main aim of this research was to develop an efficient cardiovascular disease prediction model for Jammu and Kashmir (India). Machine learning algorithms with IoT were utilized towards this aim. The prediction model is developed based on parameters. The parameters are selected after consulting with the domain experts. This research work came out to the conclusion that the Naïve Bayes algorithm performed better without hyper tuning of parameters while the Random Forest model is proven as an efficient technique with hyper parameter tuning. Because the Hyperparameter Random Forest model has higher accuracy and low error rates in comparison to all other selected models thus we select the Hyperparameterized Random Forest model for making the final output. After applying the machine learning classifiers to the heart disease dataset, it appears not astonishment that the complex classifiers like SVM and Random Forest brought better results with the loftiest accuracies of 0.8772% and 0.8540% as compared to the K-Nearest neighbor 0.8131%, Naïve Bayes 0.8262%, and Decision tree 0.8343% respectively. It is well worth emphasizing that in most instances hyper-parameter tuning is a must prerequisite to get sturdy results out of these classifiers. For future improvements, our preference will be that further research should certainly be conducted to simulate this proposed model by adding new parameters with hybrid methods like ensemble techniques and more real-world datasets. These techniques can also be applied for predicting other fields like weather forecasting, election predictions, sales predictions, and other Bioinformatics predictions, etc.

Hence, we conclude this work by stating that the combination of IoT, Machine learning, and Cloud computing can be proven as a future reality for the prediction of diseases in general and cardiovascular diseases.

IX. Current and Future Developments

In this research work, a novel technique is developed for predicting cardiovascular disease for Jammu and Kashmir (India) based upon parameters. Machine learning algorithms with python language were utilized towards the objective of this research. While the results of this model are promising, if we add more parameters the accuracy of forecasting may increase. In this model, we had predicted cardiovascular disease for Jammu and Kashmir. In future, the further research will be forecasting the election outcomes for the whole of India's well. Currently, this model can predict cardiovascular disease only for Jammu and Kashmir but using this methodology researchers will extend this research work to other developing areas also by incorporating certain parameters pertaining to those areas. Further researchers can develop the new algorithm or combine all the algorithms using an ensemble technique in order to achieve better accuracy with low error rates.

References

[1] V. Fuster, B. B. Kelly, eds., "Promoting cardiovascular health in the developing world: a critical challenge to achieve global health," *National Academies Press*, 2010, pp 465.

[2] P. Kaur, R. Kumar, M. Kumar, "A healthcare monitoring system using random forest and internet of things (IoT)", 2019, pp. 19905-1916.

[3] I. Cvitić, D. Peraković, M. Periša, M. Botica, "Novel approach for detection of IoT generated DDoS traffic," Wireless Networks, 2019, pp. 1-14, doi: 10.1007/s11276-019-02043-1.

[4] A. Tewari, B.B. Gupta, "Security, privacy and trust of different layers in Internet-of-Things (IoTs) framework," Future generation computer systems", 2002, pp 909-920.

[5] I. Cvitic, D. Perakovic, M. Perisa,M. Botica, "Definition of the IoT device classes based on network traffic flow features", in 4th EAI International Conference on Management of Manufacturing Systems. Springer, Cham, 2020, pp 1-17.

[6] C. L. Stergiou, K.E. Psannis, et al. "IoT-based Big Data secure management in the Fog over a 6G Wireless Network", IEEE Internet of Things Journal, 2020, doi: 10.1109/JIOT.2020.3033131

[7] P.K. Senyo, E. Addae, R. Boateng, "Cloud computing research: "A review of research themes, frameworks, methods and future research directions", International Journal of Information Management, vol. 38, no. 1, pp 128-139, 2018.

[8] Y. Deepthi, Kalyan K.P., Vyas M., Radhika K., Babu D.K., Krishna Rao N.V. "Disease Prediction Based on Symptoms Using Machine Learning". In: Sikander A., Acharjee D., Chanda C., Mondal P., Verma P. (eds) Energy Systems, Drives and Automations. Lecture Notes in Electrical Engineering, vol. 664, Springer, 2020.

[9] F. Jabeen, M. Maqsood, M. A. Ghazanfar, F. Aadil, S. Khan, K. Kim, "An IoT based efficient hybrid recommender system for cardiovascular disease," Peer-to-Peer Networking and Applications, vol. 12, no. 5, pp 1263-1276, 2019.

[10] D. Palani, K. Venkatalakshmi, "An IoT Based Predictive Modelling for Predicting Lung Cancer Using Fuzzy Cluster Based Segmentation and Classification," Journal of medical systems, vol. 43, no. 2, pp. 1-12.

[11] A. Benjemmaa, H. Lti, M. Ben Ayed, "Design of Remote Heart Monitoring System for Cardiac Patients", In International Conference on Advanced Information Networking and Applications. Springer, Cham, 2019. pp. 963-976.

[12] Hashi, E. K., Zaman, M. S. U., Hasan, M. R. "An expert clinical decision support system to predict disease using classification techniques". International Conference on Electrical, Computer and Communication Engineering (ECCE), 2017. pp. 396-400.

[13] P. P. Chavda, "Early Detection of Cardiac Disease Using Machine Learning," In 2nd International Conference on Advances in Science & Technology (ICAST) 2019.http://dx.doi.org/10.2139/ssrn.3370813.

[14] K. Saravananathan, T. Velmurugan, "Analyzing Diabetic Data using Classification Algorithms in Data Mining," Indian Journal of Science and Technology, vol. 9, no. 43, pp 1-6, 2016.

[15] K.U. Sreekanth, K.P. Nitha., "A Study on Health Care in Internet of Things. International", Journal on Recent and Innovation Trends in Computing and Communication, vol. 4, no. 2, 2016.

[16] D. Dziak, B. Jachimczyk, W. Kulesza, "IoT-Based Information System for Healthcare Application: Design Methodology Approach," Applied Sciences, vol. 7, no. 6, pp. 596, 2017.

[17] L. Yehia, A. Khedr, and A. Darwish, "Hybrid Security Techniques for Internet of Things Healthcare Applications," Advances in Internet of Things, vol. 5, no. 3, pp. 21–25, 2015.

[18] G. Huang, K. Huang, T. Lee and J. Weng, "An interpretable rule-based diagnostic classification of diabetic nephropathy among type 2 diabetes patients", BMC Bioinformatics, vol. 16, no. 1, pp. 1-10, 2015.

[19] Iyer, A., Jeyalatha, S., Sumbaly, R. "Diagnosis of diabetes using classification mining techniques". arXiv preprint arXiv:1502.03774. 2015. DOI:10.5121/ijdkp.2015.5101.

[20] Md. Shahriar Hassan, Atiqur Rahman, Ahmed Wasif Reza, "Health status from your body to the cloud: The behavioral relationship between IoT and classification techniques in abnormal situations," Sensors for Health Monitoring, Elsevier 2019 Jan 1, pp. 135-156.

[21] S. M. Riazul Islam, M. HumaunKabir, M. Hossain, Daehan Kwak, Kyung-Sup Kwak, "The Internet of Things for Health Care: A Comprehensive Survey," IEEE Access, vol. 3, pp. 678–708, 2015.

[22] Hashi, E. K., Zaman, M. S. U., Hasan, M. R. "An expert clinical decision support system to predict disease using classification techniques". In 2017 International Conference on Electrical, Computer and Communication Engineering (ECCE), 2017, pp. 396-400.

[23] P. Patidar, A. Tiwari, "Handling Missing Value in Decision Tree Algorithm," International Journal of Computer Applications, vol. 70, no. 13, pp. 31-36, 2013.

[24] A. Naik, L. Samant, "Correlation Review of Classification Algorithm Using Data Mining Tool: WEKA, Rapidminer, Tanagra, Orange and Knime," Procedia Computer Science, vol. 85, pp. 662-668, 2016.

[25] Saravananathan, K., Velmurugan, T. "Analyzing Diabetic Data using Classification Algorithms in Data Mining," Indian Journal of Science and Technology, vol. 9, no. 43, pp. 1-6, 2016.

[26] A.M. Koli, M. Ahmed, "Machine Learning Based Parametric Estimation Approach for Poll Prediction", Recent Advances in Computer Science and Communications, vol. 14, no. 4, pp. 1287 - 1299, doi: 10.2174/266625 5813666191204112601.

[27] Thanh Noi, P., Kappas, M." Comparison of random forest, k-nearest neighbor, and support vector machine classifiers for land cover classification using Sentinel-2 imagery. Sensors", vol. 18, no. 1, pp. 18, 2018.

[28] Podgorelec V, Kokol P, Stiglic B, Rozman I. "Decision trees: an overview and their use in medicine", Journal of medical systems, vol. 26, pp 445-463, 2002.

[29] Elmachtoub, A., Liang, J. C. N., McNellis, R. "Decision trees for decision-making under the predict-then-optimize framework". In International Conference on Machine Learning 2020, pp. 2858-2867.

[30] Kumar, P. M., & Gandhi, U. D. "A novel three-tier Internet of Things architecture with machine learning algorithm for early detection of heart diseases," Computers & Electrical Engineering, vol. 65, pp. 222-235, 2018.

[31] Chen PT, Lin CL, Wu WN. "Big data management in healthcare: Adoption challenges and implications", International Journal of Information Management, vol. 53, 102078, 2020.

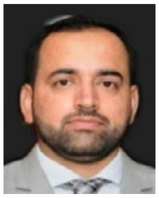
[32] Islam M. S., Islam M. T., Almutairi A. F., Beng G. K., Misran N., Amin N., "Monitoring of the human body signal through the Internet of Things (IoT) based LoRa wireless network system," Applied Sciences, vol. 9, pp. 1884, 2019.

[33] Dejene, D., Tiwari, B., Tiwari, V. "TD2SecIoT: Temporal, Data-Driven and Dynamic Network Layer Based Security Architecture for Industrial IoT," International Journal of Interactive Multimedia and Artificial Intelligence, vol. 6, no. 4, pp. 146-156, 2020, doi: 10.9781/ijimai.2020.10.002.

[34] Harish, B. S., Roopa, C. K. "Automated ECG Analysis for Localizing Thrombus in Culprit Artery Using Rule Based Information Fuzzy Network," International Journal of Interactive Multimedia and Artificial Intelligence, vol. 6, no. 1, pp. 16-25, 2020, doi: 10.9781/ijimai.2019.02.001.

Jameel Ahamed

Jameel Ahamed is a student of Doctorate of Philosophy in Department of Computer Science and Engineering, National Institute of Technology, Srinagar, J&K. He is also working as an Assistant Professor in the Department of CS and IT, Maulana Azad National Urdu University, Hyderabad-Telangana with over six years of experience in academic and research. His current research interests include the internet of things, wireless sensor networks, and computer networks.



Abdul Manan Koli

Abdul Manan Koli completed his BSC from the University of Jammu and MCA from the University of Jammu. He is currently pursuing PhD from the department of CS&IT, MANUU, Hyderabad. He has worked on numerous projects in the field of machine learning and Data Mining. He has published several conference and journal papers in national and international journals. His areas of research include Political forecasting, Machine Learning, and Big Data Analytics.

Khaleel Ahmad

Khaleel Ahmad is currently an Assistant Professor in the Department of Computer Science and Information Technology, Maulana Azad National Urdu University, Hyderabad, India. He has visited the National Defence University of Malaysia, Malaysia as a visiting faculty. He holds a Ph.D. in Computer Science & Engineering and M.Tech in Information Security. His research areas are Blockchain Technology, Cyber Security, Cryptography, and Opportunistic Network. He has 30 published papers in refereed national/international journals and conferences (viz. Elsevier, ACM, IEEE, Springer, InderScience, CMC Journal-Tech Science Press, Bentham Science), 10 book chapters (Springer, CRC Press, IGI Global, Wiley). He has edited two books for Taylor & Francis titled "Opportunistic Networks: Mobility Models, Protocols, Security & Privacy" and "Emerging Security Algorithms and Techniques" and one book for Springer titled "Functional Encryption". He has completed a successful project of Rs. 1 Lakh and Five Thousand (INR). He has supervised 4 M.Tech and 4 Ph.D. in progress. He has delivered guest lectures at the Central University of Haryana, Telangana University, and JNTU Hyderabad and also chaired the session at an international conference in Malaysia and India. He has also served as Guest Editor of the International Journal of Innovative Computing and Applications (InderScience), Journal of Recent Patents on Engineering, Bentham Science and Scalable Computing: Practice and Experience (SCPE).

Mohammad Alam Jamal

Mr. Mohammad Alam Jamal completed his B. Tech and M. Tech. from Maulana Azad National Urdu University. His areas of interest include Machine Learning, the Internet of Things, and Deep Learning.

Brij B. Gupta

Prof Brij B. Gupta received the PhD degree from Indian Institute of Technology (IIT) Roorkee, India. In more than 16 years of his professional experience, he published over 400 papers in journals/conferences including 30 books and 08 Patents with over 14000 citations. He has received numerous national and international awards including Canadian Commonwealth Scholarship (2009), Faculty Research Fellowship Award (2017), from the Govt. of Canada, MeitY, GoI, IEEE GCCE outstanding and WIE paper awards and Best Faculty Award (2018 & 2019), NIT Kurukshetra, respectively. Prof Gupta is also serving as Distinguished Research Scientist with LoginRadius Inc., USA which is one of leading cybersecurity companies in the world, especially in the field of customer identity and access management (CIAM). He is also selected in the 2021 and 2020 Stanford University's ranking of the world's top 2% scientists. He is also a visiting/adjunct professor with several universities worldwide. He is also an IEEE Senior Member (2017) and also selected as 2021 Distinguished Lecturer in IEEE CTSoc. Dr Gupta is also serving as Member-in-Large, Board of Governors, IEEE Consumer Technology Society (2022-204). Prof. Gupta is also leading IJSWIS, IJSSCI and IJCAC, IGI Global, as Editor-in-Chief. Moreover, he is also serving as lead-editor of a Book Series with CRC, World Scientific and IET press. He also served as TPC members and organized/special session chairs in ICCE-2021, GCCE 2014-2021 and TPC Chair in 2018 INFOCOM: CCSNA Workshop and Publicity Co-chair in 2020 ICCCN. Dr Gupta is also serving/served as Associate/Guest Editor of IEEE TII, IEEE TITS, IoT, IEEE Big Data, ASOC, FGCS, etc. At present, Prof. Gupta is working as Director, International Center for AI and Cyber Security Research and Innovations, and Full Professor with the Department of Computer Science and Information Engineering (CSIE), Asia University, Taiwan. His research interests include information security, Cyber physical systems, cloud computing, blockchain technologies, intrusion detection, AI, social media and networking.

Optimistic Motion Planning Using Recursive Sub-Sampling: A New Approach to Sampling-Based Motion Planning

Lhilo Kenye[1,2], Rahul Kala[1] *

[1] Centre of Intelligent Robotics, Indian Institute of Information Technology Allahabad, Jhalwa, Allahabad (India)
[2] NavAjna Technologies Private Limited, HITEC City, Hyderabad (India)

Received 19 July 2020 | Accepted 20 April 2021 | Published 1 April 2022

LA UNIVERSIDAD
EN INTERNET

Abstract

Sampling-based motion planning in the field of robot motion planning has provided an effective approach to finding path for even high dimensional configuration space and with the motivation from the concepts of sampling based-motion planners, this paper presents a new sampling-based planning strategy called Optimistic Motion Planning using Recursive Sub-Sampling (OMPRSS), for finding a path from a source to a destination sanguinely without having to construct a roadmap or a tree. The random sample points are generated recursively and connected by straight lines. Generating sample points is limited to a range and edge connectivity is prioritized based on their distances from the line connecting through the parent samples with the intention to shorten the path. The planner is analysed and compared with some sampling strategies of probabilistic roadmap method (PRM) and the experimental results show agile planning with early convergence.

Keywords

Probabilistic Roadmap, Sampling-Based Motion Planning, Robot Motion Planning, Robotics.

DOI: 10.9781/ijimai.2022.04.001

I. Introduction

Motion planning in robotics is one of the major tasks for any robot and motion planning in itself defines the process of planning how a robot should move from one position (source) to the other (goal). The problem of motion planning is extensively large, which is not just about moving a robot or a part of it from a source to a goal randomly using any path, rather it implies to problems of considering and solving the overall scenario for efficiency and producing effective results of where and how to move even in higher dimensional spaces.

There are generally two types of motion planning: deliberative planning and reactive planning. In deliberative motion planning, the workspace (map) is defined where the robot is given the information about the environment which may include the obstacles, objects to work with, free working area which then can be converted to its configuration space for planning. Whereas in reactive planning, the robot is not aware of what the global environment is like. Here as the robot moves along the environment, it performs actions without pre-planning and is generally used in unpredictable environments. The robot senses and acts in reactive whereas, in deliberative, it senses, plans then acts. In this paper, we introduce a new motion planning algorithm which falls under a type of deliberative motion planning known as sampling-based motion planning. The application of motion planning varies over a wide area which includes Industrial robot arms, planetary exploration, robotic surgery, animation, transportation systems, etc A part from robotics, the sampling-based motion planners are also used in computational biology and animations [1].

A. Sampling-Based Motion Planners

In sampling-based motion planning the workspace is converted into its configuration space, C. The pose of a robot is described by the configuration of the robot and the set of all configurations is the configuration space. In the configuration space where the source and the goal locations are given, random samples are generated and these samples are generally checked for two conditions: collision-prone (C_{obs}) or collision-free (C_{free}) and the collision-free is considered for connecting the edges. Taking the random C_{free} samples a roadmap or a tree can be introduced for the robot to move through C. Only after a collision free roadmap or path is generated the robot should move around the environment [2]. Sampling based motion planning produces fast solutions in complex maps but may not always give a solution which is why it usually is termed probabilistic completeness [3]. Sampling-based motion planning algorithms or rather planners are classified into roadmap-based planners like the probabilistic roadmap method (PRM) [2] and tree-based algorithms planners like the rapidly exploring random tree (RRT) [4], [5] and in this paper, the new planner introduced is experimentally compared with some variants of PRM. The concepts of sampling-based motion planners are also extended to reactive planning [6], [7].

B. Probabilistic Roadmap Method

The probabilistic roadmap method (PRM) is a roadmap-based planner which is one of the most commonly used sampling-based

* Corresponding author.

E-mail address: lkenye02@gmail.com (L. Kenye), rkala001@gmail.com (R. Kala).

motion planners. In a generic PRM planner, for solving in static environments, a set of random sample points are generated, say q_{rand} and from this set the C_{free} samples are considered, while the C_{obs} sample points are rejected. Using $q_{rand} \in C_{free}$ a roadmap is constructed by connecting the samples by C_{free} edges [3], [8]. After a roadmap is constructed, the condition for connectivity is checked, that is, if the source and goal are able to connect to at least one node or vertex of the roadmap without collision. As there can be multiple possible paths from source to goal through the roadmap, this kind of planners are also termed as multi-query planners. Once condition for connectivity is satisfied, graph traversal algorithms like breath-first search [9], Dijkstra, A^*, D^*, anytime A^*, anytime dynamic A^* (AD^*) [3, 10], etcetera, can be used to find the path from source to goal.

C. Quiddity of Optimistic Motion Planning Using Recursive Sub-Sampling

With the motivation from the concepts of sampling-based motion planners, we introduce a new sampling based motion planner, named optimistic motion planning using recursive sub-sampling (OMPRSS), where the planner tries to find the path optimistically without having to construct a roadmap or a tree and the least connectivity condition needs to be checked.

Whenever the straight-line path from source to destination, say L, is collision prone, OMPRSS generates a set of sample points, sort them based on their distances from the L, then tries to find a path through the sorted points. In other words, sorting heuristically prioritizes the points with easiest and shortest points solved first. If the planner fails to find a path through the first set of sample points, a new set of the same number of sample points is generated and the planner attempts to find a path with the help of the previous set of sample points. Every sample point divides the problem into a smaller problem, and the problem is solved recursively. The removal for the need of building a roadmap or a tree helps the planner to find a path faster. From the application point of view, OMPRSS could be used in systems where the time matters and the path length may not be of much concern, for example industrial robots.

The rest of the paper is arranged as follows: literature review is presented in section II, section III discusses the methodology behind the proposed approach, the experiments and results are presented in section IV, section V confers the results and the conclusions in section VI.

II. Literature Review

The geometric sampling-based motion planners are categorized into multi-query based planners which include probabilistic roadmap method (PRM), Lazy PRM, PRM*, Lazy PRM*, SParse roadmap spanner algorithm (SPARS), SPARS2, etcetera, and single-query based planners which includes rapidly-exploring random tree (RRT), RRTConnect, RRT*, Lower bound tree RRT (LBTRRT), Sparse stable RRT, Lazy RRT, bidirectional RRT (BI-RRT) and so on [11]. All the planners have a different approach to sampling and planning strategies and are capable of finding solutions in high dimensional environments. The single-query based planners intend to plan out a collision free path by incrementally building a tree towards the destination either controlled or randomly, whereas multi-query based planners generally build a roadmap and tries to find a path from source to goal through the roadmap.

Kavraki and Latombe [2] presented the elementary working aspect of PRM where the planning approach is deliberative. Here, the planning system first creates a configuration space followed by random sampling and roadmap generation belonging to a collision free field and the robot is intended to move through the constructed roadmap. Different types of PRM have been developed over time which improved the effectiveness and efficiency of PRM. Certain variants can work better in certain type of scenario. Generally, the fundamental concept of PRM stands whereas the sampling strategy is changed in the variants. Bohlin and Kavraki [12] presented a variant of PRM called Lazy PRM which reduces the overall number of collision-checking by first postulating that the roadmap is collision free followed by collision checking while searching for the shortest possible path. The PRM* falls under the category of asymptotically optimal planners, presented by Karaman and Frazzoli [13] which proves the probabilistic completeness and asymptotic optimality of the algorithm. Unlike PRM, PRM* tries to connect to an established set of neighbouring samples. Dobson and Bekris [14] promulgated the SParse roadmap spanner algorithm (SPARS) and SPARS2 which generates sparse roadmaps instead of having to consider the entire number of edges involved in constructing the roadmap.

Some of the other variants of PRM include the Bridge Test, Gaussian sampling PRM and Obstacle-based PRM where the sampling strategies are refined such that the PRM improves its performance in scenarios such as narrow corridors. In this paper, these variants including the uniform sampling PRM [2] are used to compare with the proposed planner. Hsu et al. [15] presented the Bridge Test PRM which hikes up the samples in narrow corridor areas. It basically takes a sample point, say q, and generates another sample point, say q', in the neighbourhood of q if q is collision prone and q' is also checked if it is collision prone. If they are both collision prone, the mid-point, say m of the line $L(q, q')$ is checked whether it is collision free, C_{free}. If m belongs to C_{free}, m is added to the set of vertices for roadmap construction. Boor et al. [16] presented the Gaussian sampling PRM where the samples are generated at a higher density near the obstacles. The Gaussian sampler reduces the number of samples thus improving the efficiency of the planner. Amato et al. [17] promulgated the Obstacle-Based PRM in 3-dimensional scenario where the samples are generated on or near the obstacle surfaces. If the samples are collision prone, they are moved at unit steps towards collision free space until they are in obstacle free space thus, yielding vertices near obstacles, reducing the need of taking in the samples in larger free areas. Several works have been done to compare the various variants of PRM [18, 19].

Lavalle [4] introduced the rapidly-exploring random tree (RRT). RRT has a pulling effect where the tree is iteratively built by expanding towards the random sample point instead of directly connecting to the sample point. Kuffner and LaValle [20] presented a work similar to RRT called the RRT-Connect which iteratively constructs two trees from both the source and the destination and expanded till the trees meets. Similar to the PRM*, Karaman and Frazzoli [13] presented the RRT* which is asymptotically optimal. These planners can be combined to bring out better performances. Kala [21] presented an approach of using both the RRT and PRM. Here the RRT is used for the initial search from several points simultaneously. The points are then used to create a graph through which the path planning is carried out using PRM. The results show significant improvements in the performance. Jason et al. [22] presented a different approach to sampling-based motion planning which, instead of relying entirely on a random sampler, the path planning is enforced by a deterministic approach. Solovey and Kleinbort [23] presented a percolation approach over PRM and its variants leveraging the finesse of the variants. Ichter et al. [24] presented an approach which learns to identify essential regions or samples (for example, a doorway) which is enforced by graph theory and neural networks. Vonásek et al. [25] presented a sampling-based motion planning method which first finds approximate solutions using scaled-down robot and uses the approximated solutions as a guide to find the actual path between the source and the goal. Kim et al. [26] presented lazy collision checking approach which adaptively checks collision region by generating an approximating free configuration space allowing the system to obtain clues on which region should be checked first.

The sampling-based planners have been extensively used in several works. Švestka and Overmars [27] proposed a strategy of collaborative and coordinated motion planning for multiple robots using sampling-based planners. Clark [28] also presented a work on multi-robot motion planning using PRM. Yao and Gupta [29] also presented the use of sampling-based motion planners for end-effector path planning. Sampling based motion planners are also embedded in Open Motion Planning Library (OMPL) [11] which is being extensively used. The sampling-based motion planners has led to an amplification of new problem-solving domains. Kala [30] presented a work on using multiple robots for performing mission driven tasks with the help of sampling-based motion s planners and its concepts.

III. Optimistic Motion Planning Using Recursive Sub-Sampling

Optimistic motion planning using recursive sub-sampling (OMPRSS) is a new approach to path finding based on the concept of sampling-based motion planning. The approach is considered to be optimistic as the algorithm always assumes that there is a collision free path between the source and the goal or the sub-source or sub-goal, where the source and the goal are always connected by straight lines. Only when there is a collision prone path, the algorithm generates new set of controlled random points in an attempt to find the path through the sub-source and sub-goal. OMPRSS is exclusive of trees and roadmaps. Given a map with the source and the goal, the planner first connects the source s and the goal g with a straight line, say $L(s, g)$. Then this line is checked for collision. If the line is collision free, say C_{free}, it is taken as the path as shown in Fig. 1, which is the shortest possible path from s to g. An assumption here is $C_{free} = R^N$.

In the following discussions, the terms C_{free} and C_{obs} indicates collision free and collision prone respectively, while $\mathcal{C}_{free}$ and $\mathcal{C}_{obs}$ represent the collision free configuration space and collision prone configuration space respectively. Also, a line L with at least a point in $\mathcal{C}_{obs}$ is considered a C_{obs} line.

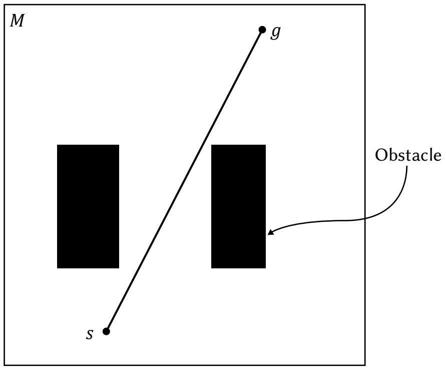


Fig. 1. A straight-line collision free path from source to goal. M denotes the map.

If the line $L(s, g)$ is collision prone, a set of uniform random sample points $q \in Q \subset \mathcal{C}_{free}$ are generated, which are collision free with the hope that the path from s to g will be collision free. A certain range, say R, is set for generating the random points. This range factor is maintained to limit the area of generating the random points with the intention to get a shorter path length. The range factor is defined as the diagonal distance of the space aligned to the coordinate axis system, within which the sample points q are generated. As an example, consider the heuristic that the robot never travels such that the projection of motion on the straight-line path is negative. Now the sampling region can be limited to the hyperspace bounded by s and g, space aligned along the coordinate axis system with the range factor as $d(s, g)$ where d is the distance function. The range factor could start from the length of $d(s, g)$ to the upper and lower limit of the map, i.e., the entire configuration space. Fig. 2 shows a condition where $d(s, g)$ is collision prone, C_{obs}, and the range of generating uniform random sample points is within the n-orthotype or hyper rectangular area with the length of $d(s, g)$ as its diagonal. Considering the coordinate points of s as $(s_1, s_2, s_3, \ldots, s_n)$ and g as $(g_1, g_2, g_3, \ldots, g_n)$, the area would be: for one corner of the hyper rectangle the point will be $c_1 = [min(s_1, g_1), min(s_2, g_2), \ldots, min(s_n, g_n)]^T$ and the other corner as $c_2 = [max(s_1, g_1), max(s_2, g_2), \ldots, max(s_n, g_n)]^T$.

Consider the case given in Fig. 2. Since $L(s, g)$ is a C_{obs} line, we generate a set of random points, say four points Q_r: $\{q_1, q_2\ q_3, q_4\}$ within a rectangular area – area resulting from the range factor taken as length of $d(s, g)$. The generated random sample points are sorted according to the distance of each point to the line $L(s, g)$. The farther a point is from the line, the lesser the priority of that point in the population of random sample points. The notion is to sample a point q such that s to g via q is a path. Closer be a point to the line from s to g, more likely it is to produce a shorter path. Moreover, collision checking effort is proportional to the length of the path. Closer be a point to the line, lesser is the computational cost. Let us consider that, after sorting according to the distance, the points are arranged as Q_p: $\{q_2, q_4\ q_3, q_1\}$. Here, Q_p is a priority queue of the points where for further task the planner first chooses the point with the highest priority (least distance from q_i to $d(s, g)$), in this case point q_2. Before going further, let us first look into the method of prioritizing the sample points.

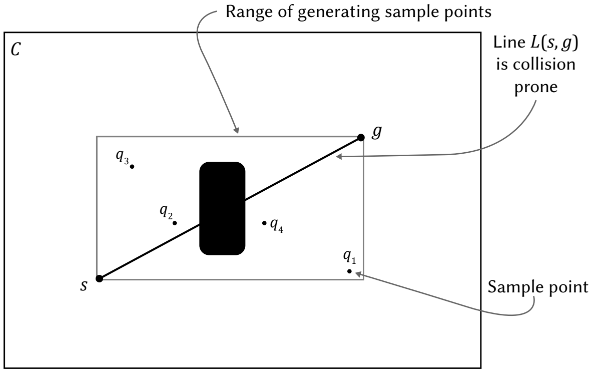

Range of generating sample points
C
Line $L(s, g)$ is collision prone
g
q_3
q_2
q_4
q_1
s
Sample point

Fig. 2. Range set to length of $L(s, g)$ resulting in a rectangular area for generating sample points. C denotes the configuration space.

A. Sample Prioritization

As mentioned, the random points belonging to $\mathcal{C}_{free}$ can be prioritized based on the distance of a point q_i to the line $L(s, g)$ All the sample points generated randomly will be at a particular distance from the line $L(s, g)$. Finding the distances of each point from $L(s, g)$ can help prioritize them. One way of determining the point distance to a line is measuring the normal distance, which basically is the shortest distance of a point to a line.

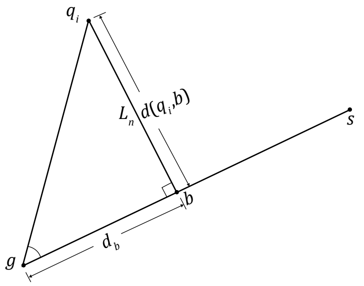


Fig. 3. An illustration of the normal distance of point q_i to line $L(s, g)$, $d(q_i, b)$ which would be used for prioritizing the sample point.

In Fig. 3, points $s = (x_s, y_s)$ and $g = (x_g, y_g)$ are the location of source and goal respectively. The point $q_i = (x_i, y_i)$ is the location of a random sample point generated and the point $b = (x_b, y_b)$ is the intercept of the normal line, say L_n, to $L(s,g)$. Here, we are interested in determining the distance of point q_i to the line $L(s,g)$, say $d(q_i, b)$ as depicted in Fig. 3. Let the distance between point g and b be d_b. We can deduce that

$$(q_i - g).(s - g) = |q_i - g|\,|s - g|\,cos\theta \tag{1}$$

The line L_n is normal to $L(s,g)$ forming a right triangle Δgbq_i. We have:

$$d_b = |q_i - g|\,cos\theta \tag{2}$$

$$\Rightarrow cos\theta = \frac{d_b}{|q_i - g|} \tag{3}$$

Replacing the expression of $cos\theta$ in Eq. (1), we have:

$$(q_i - g).(s - g) = |q_i - g|\,|s - g|\,\frac{d_b}{|q_i - g|} \tag{4}$$

$$d_b = \frac{(q_i - g).(s - g)}{|s - g|} \tag{5}$$

$$d_b = (q_i - g).\hat{u}(s - g) \tag{6}$$

Where, $\hat{u}$ is a unit vector. We can determine $d(q_i, b)$, depicted in Fig. 3 as:

$$d(q_i, b) = \sqrt{|q_i - g|^2 - d_b^2} \tag{7}$$

The sample points can be prioritized based on the distances of each point from the line $L(s,g)$ which can be obtained using Eq. (7).

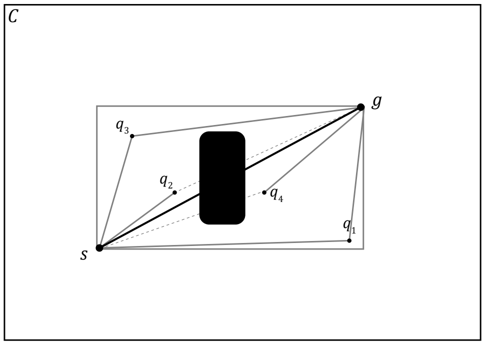


Fig. 4. Path from s to g through the sample points (q_1, q_2, q_3, q_4) generated within a specified range or area. Dotted lines denote collision-prone paths.

B. Working Principle of OMPRSS

In this section, we will discuss the working of the planner when the straight-line path from source to goal is collision prone. If $L(s,g)$ is collision prone we generate a set of uniform random points $Q \subset C_{free}$. These points are prioritized using the approach of point prioritization discussed in the previous section. Considering the case given in Fig. 2, we generated four C_{free} random points and prioritized them as Q_p: $\{q_2, q_4\ q_3, q_1\}$. The purpose of prioritization is to go about with the point with the highest priority for the next step.

After the points are prioritized, a path from s to g is checked through each sample points, shown in Fig. 4. As point q_2 has the higher priority in this case, we go about with q_2 by first checking whether the straight-line path from s to q_2 either belongs to C_{free} or C_{obs} and followed by checking the straight line path from q_2 to g. If a C_{free} path is found from s to g through $L(s, q_2)$ and $L(q_2, g)$, the planner returns with a path connecting the source to goal through the sample point q_2. But as in the case of Fig. 4, since $L(q_2, g)$ is collision prone (C_{obs}), we consider the next point in the priority queue q_4, which is the next closest point to $L(s, g)$. The path from s to g is still C_{obs} as the line $L(s, q_4)$ is C_{obs}. Next, we consider q_3. Here, both $L(s, q_3)$ and $L(q_3, G)$ are C_{free}, so in this case the planner returns a successful path. The point q_1 also produces a C_{free} path, but as its priority is lesser compared to q_3, the path through q_1 is ignored. Through the example of Fig. 4 we can conclude that point prioritization helps in finding a shorter path as any point closer to the line would produce a shorter path length than that of a point farther to the line of concern.

Let us consider a different case shown in Fig. 5. In this case, we can observe that all sample points produce C_{obs} path. This planner works recursively, where in this type of cases we repeat the same procedure as we did for the case of Fig. 4 with the sample point as the new source or goal depending on which of the line ($L(s, q_i)$ or $L(q_i, g)$) is C_{obs}. In Fig. 5, we can observe two cases:

- Firstly, one of $L(s, q_i)$ or $L(q_i, g)$ is C_{obs} as in cases of points q_1, q_2 and q_4. Here, in case of points q_1 and q_2 the sample points become the new goals and the source remains the same in the next recursion. Whereas in case the of q_4, the goal remains the same as it is and the sample point becomes the new source.
- The second case is when both $L(s, q_i)$ and $L(q_i, g)$ are C_{obs}. In this case, we change the priority of that point by multiplying with a penalty. The value of the penalty can be chosen based on the distances produced by all sample points such that after multiplying with the penalty, the distance of that point increases so that the priority of that point gets lesser than the point with the largest distance but only has one C_{obs} line initially. If all the paths produced the random points are C_{obs}, a penalty is multiplied to all points and in this condition, all points get back to its original priority list. This is done to reduce computational complexities as for a point with both lines to be C_{obs}, the number of recursions would increase. If we first work on points with only one C_{obs} line and find a path, it would reduce the depth of recursion as compared to points having both lines to be C_{obs}. When, for all sample points, if both $L(s, q_i)$ and $L(q_i, g)$ are C_{obs}, we can avoid the random points and generate new points or start the planner all over, but if we keep avoiding points when all points produce both paths as C_{obs}, the time cost could gradually increase. In Fig. 5, point q_3 has both $L(s, q_3)$ and $L(q_3, g)$ as C_{obs}. The priority queue changes from Q_p: $\{q_1, q_2, q_3, q_4\}$ to Q_p: $\{q_1, q_2, q_4, q_3\}$.

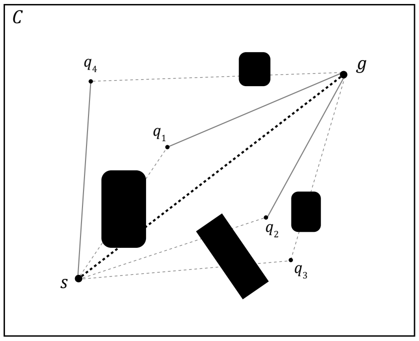


Fig. 5. All sample points producing collision-prone (C_{obs}) paths.

Summarizing, the collision aware priority is given by:

$$\pi' = d\perp(q_{i,}L)(1 + \alpha C_1 + \beta C_2) \tag{8}$$

Where, α and β are very large constants and C_1 and C_2 can be defined as

$$C_1 = \begin{cases} 0, & if\ L(s, q_i)\ is\ C_{free} \\ large\ number, & otherwise \end{cases} \tag{9}$$

$$C_2 = \begin{cases} 0, & if\ L(q_i, g)\ is\ C_{free} \\ large\ number, & otherwise \end{cases} \tag{10}$$

When a line from source to sample point or sample point to the goal (or both) is C_{obs}, we get into a recursion where the same set or number of C_{free} random points are generated. Let us consider the point q_1 in Fig. 5. The path from q_1 to g is C_{free} but the path from s to q_1 is C_{obs}. As a result, we generate another four random points, say Q_{11}: $\{q_{11}, q_{12}, q_{13}, q_{14}\} \in C_{free}$ within the range, say between s and q_1. The area would be a square or rectangle with $L(s, q_1)$ as the diagonal. The same procedure is followed with the intention to find a path from s to q_1 by checking for C_{free} paths through one of the new sample points with q_1 as the new goal. Once a path is found, the original source and goal can get connected via the sample points as shown in Fig. 6. In this case, the path would be $s \rightarrow q_{13} \rightarrow q_1 \rightarrow g$. If a path is not found, from s to g through the sample points Q_{11}: $\{q_{11}, q_{12}, q_{13}, q_{14}\}$, we continue with the same procedure using Q_{11} as the new source or goal, resulting into another depth of recursion. The limit of recursion can be set to a certain depth. Change in the depth limit can also change the response of the planner. The procedure of generating the random points is repeated until a path is found or the depth limit is reached. We can observe that by maintaining a range the total path length can be reduced but, setting the range factor has some issues which we will discuss in the next section.

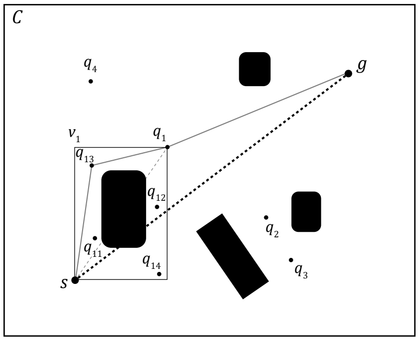


Fig. 6. Illustration of new sample points (q_{11}, q_{12}, q_{13}, q_{14}) generated in recursion of depth 1. V_1 denotes the new area of generating the sample points setting the range factor to be $L(S, q_1)$.

C. Repercussion of Range and Proposed Solution

We have observed that setting a range factor can greatly reduce the path length, but there are some issues which pops up with it. In this section, we will discuss some of the possible situations where the range factor becomes an issue and the solutions to it. Let us first consider the issues with regard to the range factor set to the length $L(s, g)$, say d_L. To understand these issues, let us take a 2D scenario as an example. There are two possible cases where the planner can get into a no-solution state:

- When $L(s, g)$ becomes perpendicular to x-axis: Here, the slope of $L(s, g)$ becomes infinite.

$$m = \frac{\Delta y}{\Delta x} = \frac{y_g - y_s}{x_g - x_s} \tag{11}$$

Here:

$$x_g = x_s \Rightarrow m = \frac{y_g - y_s}{0} = undefined \tag{12}$$

- When $L(s, g)$ becomes parallel to x-axis: The slope of the normal to $L(s, g)$ will be undefined as change in y will be zero.

$$m = \frac{0}{x_g - x_s} = 0 \tag{13}$$

Here, slope of normal to the line $L(s, g)$ will be

$$s = -\frac{1}{m} = undefined \tag{14}$$

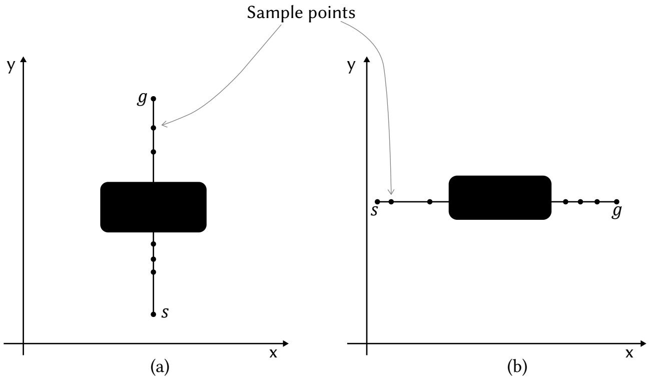


Fig. 7. Representation of the two cases where the range for defining the area of generating random sample points would fail to produce points to find a path. (a) The line $L(s, g)$ perpendicular to the x-axis, (b) The line $L(s, g)$ parallel to x-axis. In both cases, the points are generated along the line $L(s, g)$ for choosing range factor as d_L.

Finding the distance of a point to the line will not be an issue in both cases as the distance in the case of the first problem will be the difference in the x values of the sample point and the intercept of the line and for the second, the difference in y values. The problem is that the volume of generating random sample points will be the line itself, i.e., the points will lie on the line as shown in Fig. 7. In these cases, no path can be found.

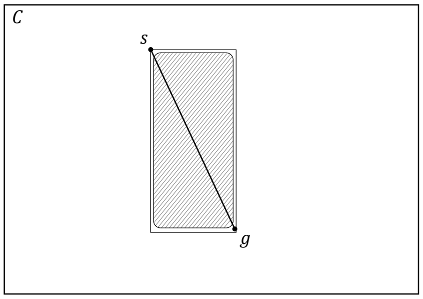


Fig. 8. An illustration where OMPRSS has high possibility of failing due to less C_{free} region or area for generating sample points. The inner most rectangular shape filled with lines indicates an obstacle, say O. s and g are the source and goal respectively and the rectangular box, say r with the diagonal line connecting s and g is the area for generating sample points. The space between the O and r is the C_{free} region for generating the sample points.

Another problem that can occur is the situation given in Fig. 8. In this case, the source and the goal are both too close to the obstacle which results in small C_{free} area to generate the C_{free} points. The planner in this case has very thin chances to find a path from source to goal because the points can get too close to each other with the increase in the depth of recursion and may even reach to an extend where the points are generated in the same point in the configuration space resulting in generating zero distances. A similar case would be as illustrated in Fig. 10 where, if the range factor is taken as d_L, there is no way to find a path. All points generated will not be able to find out a path as long as the range factor is d_L in this case.

The solution to the above problems would be increasing the range of generating the points by some factor which will eventually increase the volume of sample point generation. Various range factors can be assigned to the planner such that the prospect of finding a path is increased. The degree to which the range factor increases is flexible and there are no discrete methods to increasing it. In this paper, we propose a particular way of increasing the volume. Instead of using d_L, which is the length of $L(s, g)$, for setting up the volume of generating the sample points, the line $L(s, g)$ is extended by an extension factor, say η, such that the volume increases. Here, both the ends of $L(s, g)$ are extended by η. In other words, both s and g are either added or subtracted by η depending on where s and g are located. If the location of source is $s = (s_1, s_2, \ldots, s_n)^T$ and goal is $g = (g_1, g_2, \ldots, g_n)^T$, the expression for volume extension can be written as:

$$s' = \begin{cases} s_i + \eta, & if\ (s_i - g_i) \geq 0, \forall i \in n \\ s_i - \eta, & if\ (s_i - g_i) < 0, \forall i \in n \end{cases} \tag{15}$$

$$g' = \begin{cases} g_i - \eta, & if\ (s_i - g_i) \geq 0, \forall i \in n \\ g_i + \eta, & if\ (s_i - g_i) < 0, \forall i \in n \end{cases} \tag{16}$$

Here, s' and g' are the new corner points of the volume and $L(s', g')$ becomes the diagonal line of the extended volume. In both Eq. (15) and (16), the greater or equal condition ($\geq$) is used in order to deal with cases when $L(S, G)$ is perpendicular or parallel to one of the axes. The case where $L(S, G)$ is perpendicular or parallel to any axes occurs when for $i = 1, 2, \ldots, n$, $(S_i, - G_i) = 0$.

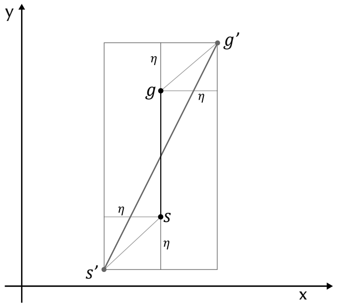


Fig. 9. Illustration of volume extension for generating the sample points by η for $L(s, g)$ perpendicular to the x-axis. The extension operation redefines the volume such that $L(s', g')$ is the diagonal of the incremented volume.

One of the possible extension factors could be the half of the length of $L(s, g)$, that is $\eta = 0.5 \times d_L$. Let us once again consider a 2D scenario (Fig. 9) with source as (x_s, y_s) and goal as (x_g, y_g) where, $x_g > x_s$ and $y_g > y_s$. Let the length of $L(s, g)$ be d_L. Based on the extension condition, the end points after extension of d_L would be

$$(x'_s, y'_s)^T = (x_s - \eta, y_s - \eta)^T \tag{17}$$

$$(x'_g, y'_g)^T = (x_g + \eta, y_g + \eta)^T \tag{18}$$

This will result in a larger volume in the configuration space for generating the random points as shown in Fig. 10. The factor can even be increased if required such as using the whole d_L as the factor, that is $\eta = d_L$ or even up to the extent of using the whole configuration space as the range. The chances of getting a path may increase in η, but as it increases, the chances of getting points farther away from the line are more, as a result we might end up with a larger path length.

If the range factor is changed, the process of point prioritization needs to be changed to some extend as problems could arise in condition where the point is beyond the actual line as shown in Fig. 11. Here, a normal line does not pass through the line $L(s, g)$ as a result the point of intercept B would go beyond $L(s, g)$. This issue is taken into consideration because though a point may lie the closest to the line based on the normal line it may still be quite far away from the source and the goal, which when given with higher priority may result in a longer path than those points which have larger normal distance. This problem can be solved by comparison of the distance between $d(s, g)$, $d(s, b)$ and $d(b, g)$. First the following condition as given in Eq. (19) needs to be checked:

$$d\perp(q_i L) = \begin{cases} d(q_i, b), if\ d(s, g) = d(s, b) + d(b, g) \\ \min\big(d(q_i, s), d(q_i, g)\big), \quad otherwise \end{cases} \tag{19}$$

We can conclude from these issues that the larger the range factor, the chances of getting a shorter path reduces. The range factor can be configured based on the type of map the robot is dealing with.

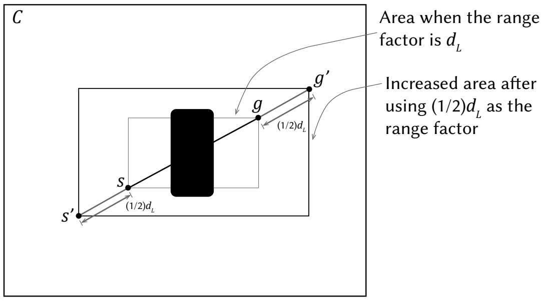

C
Area when the range factor is d_L
Increased area after using $(1/2)d_L$ as the range factor
g'
g
s
s'
$(1/2)d_L$
$(1/2)d_L$

Fig. 10. A representation of using a factor of half of d_L to increase the volume for generating random points. Here, d_L is the length of $L(s, g)$.

D. Pseudo Code

The pseudo code for OMPRSS is given in Algorithm 1. The function $L(s, g)$ mentioned in Algorithm 1 is given in Algorithm 2. The function *connect* $(s, g, depth)$ returns two values. First is the connectivity *con* and second is the path τ. If the source and the goal is connected via a path, a true value is returned and all the points connecting the source and the goal (including the source and the goal) are returned in τ. The function *distanceToLine*(q_i, s, g) returns the distance of a sample point q_i to the line $L(s, g)$. Here, s and g will keep changing with every depth as the sample point will become new source or goal with every recursion depth. The distances are used for sorting which actually is the step of prioritizing. The parameter *limit* in Algorithm 1 is the limit or threshold for the recursion depth.

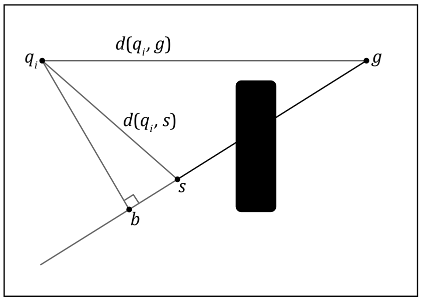


Fig. 11. An illustration of the intercept b laying beyond the line $L(s, g)$ due to volume extension.

IV. Experiments and Results

The experiment was conducted on two dimensional maps, bitmap images, considering the obstacles to be 0s and the collision free area to be 1s. The robot is taken to be a point robot inferring that the workspace need not be converted to its configuration space. The results on the working of OMPRSS is first presented followed by the response of the planner to different parameter alterations and comparison with other sampling-based motion planners.

Algorithm 1: Optimistic motion planner using recursive sub-sampling

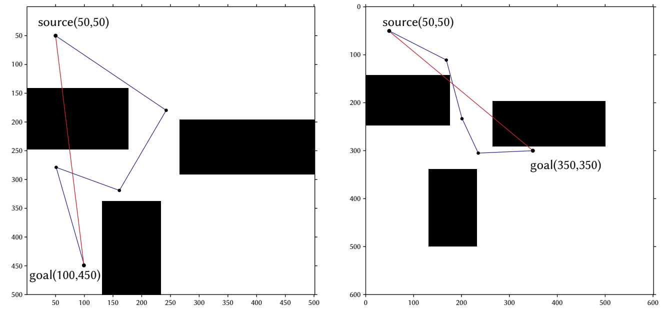
```
$[con, \tau] = connect(s, g, depth)$
Input: source ($s$), goal ($g$) and current recursion depth ($depth$)
Output: $con = true$ if connected, $false$ if no path found, $\tau$ = Path from $s$ to $g$
if $L(s, g) = true$ then
    $con = true, \tau = [s, g]$
    return $[con, \tau]$
end if
if $depth > limit$ then
    $con = false$
    return $[con, NIL]$
end if
$\tau = \phi$
if $L(s, g) = false$, $Q$ = uniform random points, $n = 1, 2, \ldots$ then
    for $i = q_i \in Q$ do
        $\pi(q_i) = distanceToLine(q_i, s, g)$
        if $L(s, q_i) = false$ or $L(q_i, g) = false$ then
            $\pi'(q_i) = \pi(q_i)(1 + \alpha C_1 + \beta C_2)$ // Eq. (8)
        end if
    end for
    Sort $Q$ based on $\pi'(q_i)$
    for $i = 1$ to $n$ do
        if $L(s, q_i) = false$ and $L(q_i, g) = true$ then
            $[con, \tau] = connect(s, \pi'(q_i), depth + 1)$
        end if
        if $L(s, q_i) = true$ and $L(q_i, g) = false$ then
            $[con, \tau] = connect(\pi'(q_i), g, depth + 1)$
        end if
        if $L(s, q_i) = false$ and $L(q_i, g) = false$ then
            $[con_1, \tau_1] = connect(s, \pi'(q_i), depth + 1)$
            $[con_2, \tau_2] = connect(\pi'(q_i), g, depth + 1)$
            $con = (con_1 \wedge con_2)$
            $\tau = [\tau_1, \tau_2]$
        end if
        if $con = true$ and $NotEmpty(\tau)$ then
            return $[con, \tau]$
        end if
    end for
    $con = false$
    return $[con, \tau]$
end if
```

Algorithm 2: Line-Collision check

```
$L(S, G)$
Input: Points source ($s$) and goal ($g$)
Output: true if straight line from $s$ to $g$ is $C_{free}$, else $false$
$s = source, g = goal$
for $i = 0$ to $1$ in small steps do
    if $i \times s + (1-i) \times g \in C_{obs}$ then
        end if
    end if
end for
return $true$
```

A. Results on the Working of OMPRSS

The initial experiment was performed on the basic working method of OMPRSS. The maps used were 500 ×500 bitmap images. The results are shown in Fig. 12(a) with the source at (50, 50) and the goal at (100, 450) and Fig. 12(b) with the source at (50, 50) and the goal at (350, 300), both had four sample points generated per recursion and the recursion depth limit was kept at 4. The range factor used was set to half of d_L. The time required to find the path in the case of Fig. 12(a) was 0.05681274 seconds and the cost (path length) was 690.2206, while 0.8073688 seconds and 453.9041 in case of Fig. 12(b). In the figures, the entire generated sample points are not indicated besides the sample points resulting to the path formation. As the sample points are generated randomly, different path lengths can be generated each time the planner is executed. Experimental results are shown in Fig. 13.



(a) (b)

Fig. 12. OMPRSS path planning results in a bitmap image of size 500 ×500. (a) and (b) illustrates the paths found by OMPRSS in different maps where, in both cases, the planner generates sample points in C_{free} region to connect the source and goal as the straight lines connecting them are collision prone.

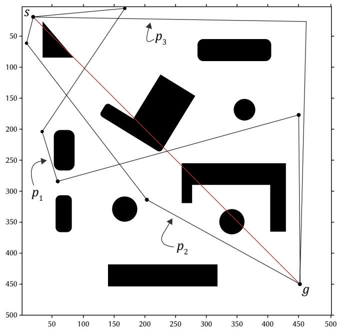


Fig. 13. Response of OMPRSS at different executions over the same source and goal locations - source at (10,10) and goal at (450,450), using the same range. The range used was half of d_L The path p_1 with length 1154, path p_2 with length 866 and p_3 with path length 645.

B. Response to Changes in Parameters

There are a few parameters which can be changed in OMPRSS. They include:

- Range (which determines the area of generating random sample points)
- Sample points
- Depth of recursion

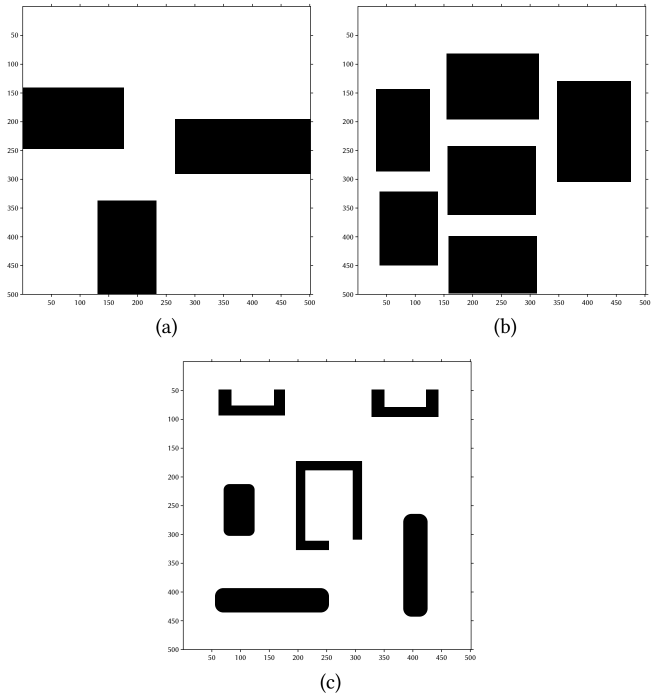
1. Altering the Range Factor

For generating the random sample points three variations of range (variants discussed in section III.B) were used for which all the variants responded differently in different scenarios. The variants of range are:

- Half the line of $L(s, g)$, i.e. $0.5 \times d_L$. Let us consider this range as R_1

- The length of $L(s, g)$, d_L. Let this range be R_2
- The entire map. Let this range be R_3

The maps used for this experiment are given in Fig. 14. Table I shows the observations of different range factors responding to different scenarios. Map size used: 500 ×500. Number of sample points used is 4. Depth (or limit) of recursion is 4. Cost is the path length and the time taken are in seconds.



(a)

(b)

(c)

Fig. 14. 500 ×500 bit maps used for comparison of the response of OMPRSS over change in range factors. (a) as M_1, (b) as M_2 and (c) as M_3 in Table I.

2. Altering the Number of Sample Points

The other parameter that can be changed is the number of sample points generated at each recursion. The experiment was performed on map M_3 (Fig. 14(c)). The depth of recursion was 4 and the range factor was half of d_L. Fig. 15 and Fig. 16 shows the results of the experiment. Here the number of sample points is incrementally increased at a step of 1 per iteration from 4 to 53, whereas the map, depth of recursion, range factor and the location of both source and goal are not changed.

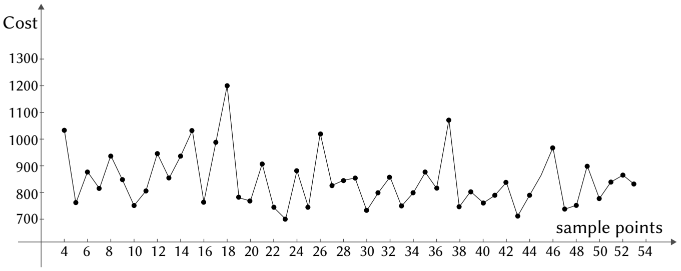


Fig. 15. Plot of cost versus sample points: showing the response of OMPRSS over change in sample points on M_3 (Fig. 14(c)) with source at (10, 10) and goal at (470, 470).

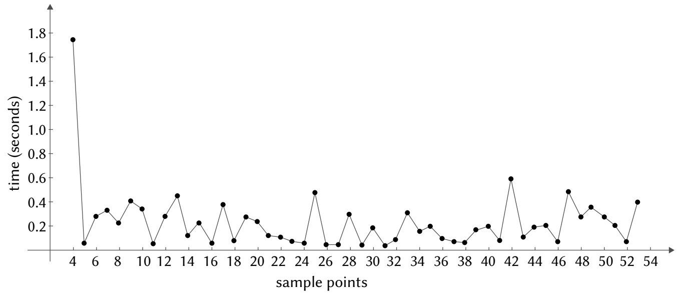


Fig. 16. Plot of time versus sample points: showing the response of OMPRSS over change in sample points on M3 (Fig. 14(c)) with source at (10, 10) and goal at (470, 470).

TABLE I. Comparison of Responses of the Planner Over Different Range Factors in Different Scenarios. The Maps Used Are Given in Fig. 14 As M_1, M_2 and M_3

Sl. No.	Map	Source	Goal	Range	Cost	Time (seconds)	Result
1	M_1	(20,20)	(450,450)	R_1	**802.6841**	0.04598	Pass
2	M_1	(20,20)	(450,450)	R_2	*941.3936*	0.05159	Pass
3	M_1	(20,20)	(450,450)	R_3	1006.8177	0:02251	Pass
4	M_1	(100,100)	(10,300)	R_1	*895.1789*	5.13672	Pass
5	M_1	(100,100)	(10,300)	R_2	**783.3984**	3.51435	Pass
6	M_1	(100,100)	(10,300)	R_3	930.3145	0.01224	Pass
7	M_2	(10,10)	(450,450)	R_1	741.5306	0.76386	Pass
8	M_2	(10,10)	(450,450)	R_2	**727.5929**	0.56532	Pass
9	M_2	(10,10)	(450,450)	R_3	*769.5605*	0.13955	Pass
10	M_2	(50,300)	(450,100)	R_1	**664.7577**	0.76386	Pass
11	M_2	(50,300)	(450,100)	R_2	*708.8577*	0.081249	Pass
12	M_2	(50,300)	(450,100)	R_3	729.6059	0.43471	Pass
13	M_3	(100,70)	(450,470)	R_1	*716.774*	0.062411	Pass
14	M_3	(100,70)	(450,470)	R_2	**709.3188**	0.014065	Pass
15	M_3	(100,70)	(450,470)	R_3	1373.9801	0.073104	Pass
16	M_3	(100,70)	(230,200)	R_1	---	---	Fail
17	M_3	(100,70)	(230,200)	R_2	---	---	Fail
18	M_3	(100,70)	(230,200)	R_3	**754.2889**	1.12045	Pass

3. Altering the Depth of Recursion

Keeping the number of sample points constant including the map, range factor, and the source and goal locations while altering the depth of recursion, OMPRSS responded differently. The experiment was performed over M_3 (Fig. 14(c)). Number of sample points kept constant at 4, source at (10, 10) and goal at (470, 470) and half of d_L as the range factor. The depth of recursion was initiated at 4 and not at 1 because at a depth of 1, the planner is likely to fail. The reason behind starting at 4 is to start off at a likely possible successful planning followed by increasing the depth to observe the response over incrementally increasing the depth. The experimental results are shown graphically in Fig. 17 and Fig. 18.

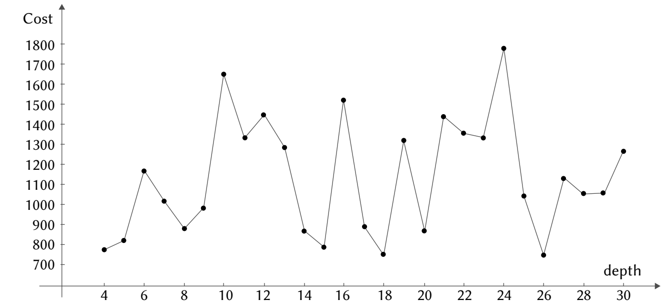


Fig. 17. Plot of cost versus depth: showing the response of OMPRSS over change in depth of recursion on M_3 (Fig. 14(c)) with source at (10, 10) and goal at (470, 470).

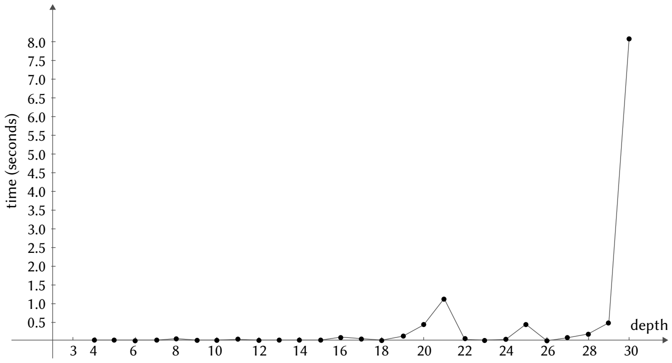


Fig. 18. Plot of time versus depth: showing the response of OMPRSS over change in depth of recursion on M_3 (Fig. 14(c)) with source at (10, 10) and goal at (470, 470).

C. Comparison of OMPRSS With PRM Variants

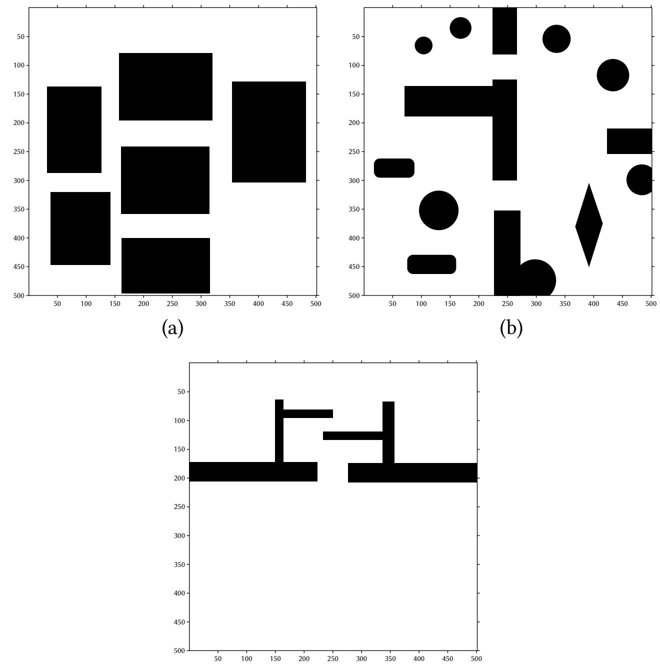
In this section we will look into the performance of OMPRSS against some of the variants of probabilistic roadmap method (PRM). The PRM variants used are:

- Bridge test PRM
- Gaussian sampling-based PRM
- Obstacle-based PRM
- Uniform cost search PRM

The comparison was made based on three outcomes:

- Cost inferring the path length
- Time required for planning
- Success rate specifying the number of successful planning

The experiment was performed to view the response of both the PRM variants and OMPRSS by increasing the sample points. The first comparison experiment was performed on M_6 from Fig. 19, the location of source at (10, 10) and goal at (450, 450).



(a)

(b)

(c)

Fig. 19. 500 × 500 bit maps used for comparison of PRM variants and OMPRSS (a) as M_4, (b) as M_5 and (c) as M_6.

The initial number of sample points for the PRM variants was 200 incremented by 50 per iteration, iterated 80 times resulting to 4150 sample points at the end. Whereas for OMPRSS, the initial number of sample points was 5 incremented by 1 per iteration, iterated 80 times. For both PRM variants and OMPRSS, for a particular number of sample points (that is in one iteration), say S_i, the planner was executed 25 times, and from the generated results, of each S_i, the average of the cost, the execution time and the success rate were considered. For OMPRSS, the range factor used was half of d_L and the depth of recursion was kept at 4. If any planner fails to find a path in any of the 2000 times executed (i.e. 80 × 25 times), the cost can be taken to be a high value - in the experiment, cost was taken as 9999 if any planner fails. And also, if a planner fails to find any path for the 25 times executed per S_i the success rate would be 0. On the other hand, if a planner succeeds to find paths in all 25 times, the success rate would be 1.

Fig. 20, Fig. 21 and Fig. 22 show the graph for PRM variants of cost, time and success rate versus increasing sample points respectively. Whereas Fig. 23, Fig. 24 and Fig. 25 show the result of OMPRSS for cost, time and success rate against the increasing sample points. In order to view the successful planning and cost convergence against time, the data generated from both PRM variants and OMPRSS were plotted. Fig. 26 and Fig. 27 shows the scatter plot of cost versus time and success rate versus time generated by PRM variants respectively, and Fig. 28 and Fig. 29 for OMPRSS.

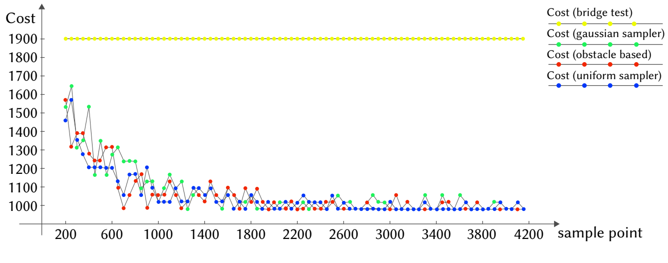


Fig. 20. Graph of cost versus increasing sample points generated by the PRM variants in M_6.

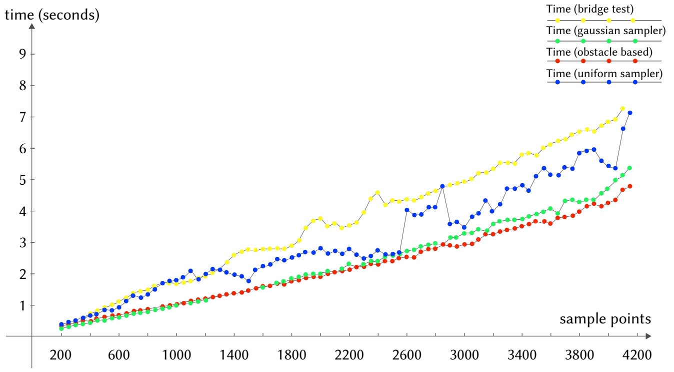


Fig. 21. Graph of time versus increasing sample points generated by the PRM variants in M_6.

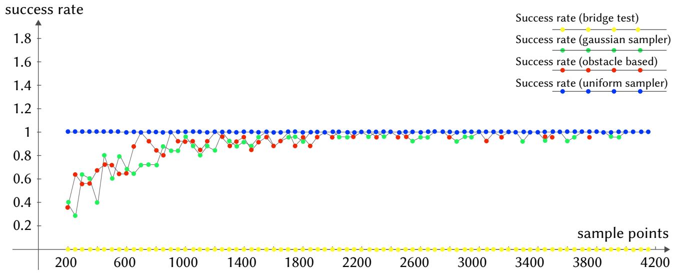


Fig. 22. Graph of success rate versus increasing sample points generated by the PRM variants in M_6.

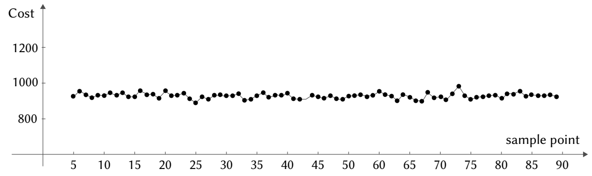


Fig. 23. Graph of cost versus increasing sample points generated by the OMPRSS in M_6.

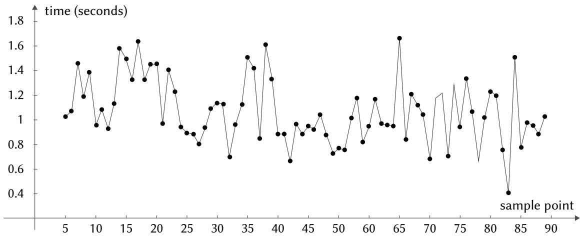


Fig. 24. Graph of time versus increasing sample points generated by the OMPRSS in M_6.

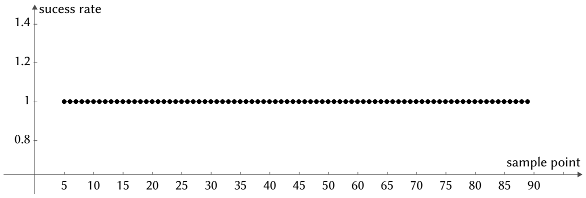


Fig. 25. Graph of success rate versus increasing sample points generated by the OMPRSS in M_6.

TABLE II. Cost Comparison of the PRM Variants and OMPRSS in Different Scenarios (Maps in Fig. 19) at Various Time Steps. Source and Goal Kept at (10, 10) and (450, 450) Respectively for All Maps

Map	Time (sec)	Bridge test	Gaussian sampler	Obstacle-based	Uniform sampler	OMPRSS
M_4	Less than 0.04	9999	9999	9999	9999	9999
M_4	Less than 0.08	9999	9999	9999	9999	**890.5013**
M_4	Less than 0.12	9999	9999	9999	9999	**890.5013**
M_4	Less than 0.16	9999	9999	9999	9999	**890.5013**
M_4	Less than 0.2	9999	9999	9999	9999	**890.5013**
M_4	Less than 0.24	9999	8527.6245	**806.7357**	9999	*890.5013*
M_4	Less than 0.28	9999	7425.3433	**806.7357**	9999	*890.5013*
M_4	Less than 0.32	9999	7425.3433	**806.7357**	9999	*890.5013*
M_4	Less than 0.36	9999	5585.5652	**806.7357**	9999	*890.5013*
M_4	Less than 0.4	9999	5585.5652	*806.7357*	**782.1605**	890.5013
M_5	Less than 0.01	9999	9999	9999	9999	9999
M_5	Less than 0.06	9999	9999	9999	9999	**774.6329**
M_5	Less than 0.11	9999	9999	9999	9999	**774.6329**
M_5	Less than 0.16	9999	9999	9999	9999	**774.6329**
M_5	Less than 0.21	9999	9999	9999	9999	**774.6329**
M_5	Less than 0.26	9999	9999	9999	9999	**774.6329**
M_5	Less than 0.31	9999	3750.8509	*1160.7313*	9999	**774.6329**
M_5	Less than 0.36	9999	1178.5719	*794.7315*	9999	**774.6329**
M_5	Less than 0.41	9999	1166.7064	794.7315	*786.6865*	**774.6329**
M_5	Less than 0.5	9999	1166.7064	790.1796	*786.6865*	**774.6329**
M_6	Less than 0.1	9999	9999	9999	9999	9999
M_6	Less than 0.2	9999	9999	9999	9999	9999
M_6	Less than 0.3	9999	**6338.6594**	*6730.6351*	9999	9999
M_6	Less than 0.4	9999	**4146.1292**	*4169.2192*	5995.6477	9999
M_6	Less than 0.5	9999	*4146.1292*	6730.6351	4905.0506	**957.2462**
M_6	Less than 0.6	9999	*2672.9671*	3435.7925	4527.0484	**957.2462**
M_6	Less than 0.7	9999	2671.515	*1970.3431*	4527.0484	**910.5138**
M_6	Less than 0.8	9999	2671.515	**862.0898**	4527.0484	*906.7916*
M_6	Less than 0.9	9999	1930.6015	**862.0898**	4164.7177	*891.2265*
M_6	Less than 1	9999	1930.6015	**858.9613**	4164.7177	*891.2265*
M_6	Less than 1.5	9999	**831.5308**	*858.9613*	4164.7177	900.8705
M_6	Less than 2	9999	**828.2267**	*844.9996*	4164.7177	922.8004

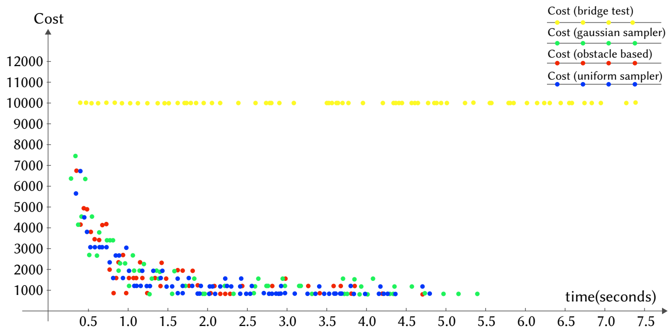


Fig. 26. Graph of cost versus time generated by the PRM variants.

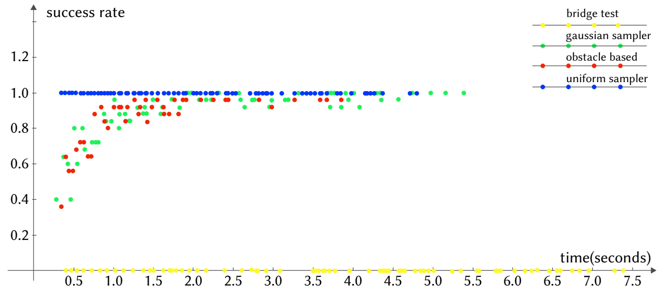


Fig. 27. Graph of success rate versus time generated by the PRM variants.

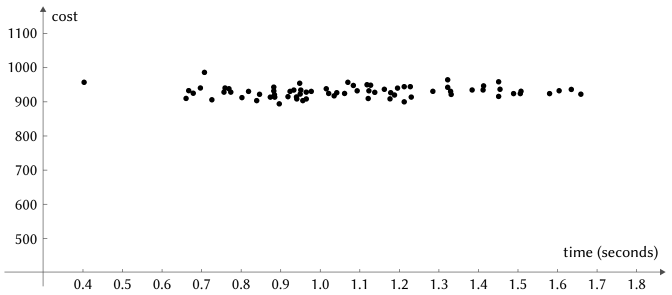


Fig. 28. Graph of cost versus time generated by the PRM variants.

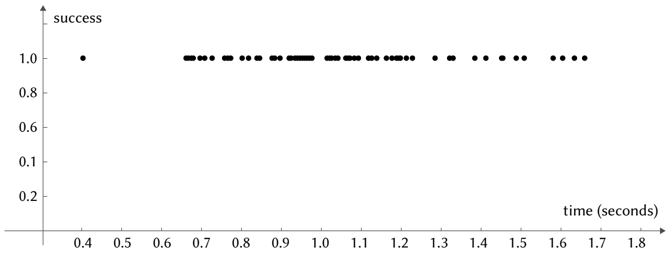


Fig. 29. Graph of success rate versus time generated by the PRM variants.

Table II and Table III shows a broader view of the experiment performed shown in Fig. 26 to Fig. 29. Both the PRM variants and

TABLE III. Success Rate Comparison of the PRM Variants and OMPRSS in Different Scenarios (Maps in Figure 19) at Various Time Steps. Source and Goal Kept at (10, 10) and (450, 450) Respectively for All Maps

Map	*Time (sec)*	*Bridge test*	*Gaussian sampler*	*Obstacle-based*	*Uniform sampler*	*OMPRSS*
M_4	*Less than 0.04*	*0*	*0*	*0*	*0*	*0*
M_4	*Less than 0.08*	*0*	*0*	*0*	*0*	*1*
M_4	*Less than 0.12*	*0*	*0*	*0*	*0*	*1*
M_4	*Less than 0.16*	*0*	*0*	*0*	*0*	*1*
M_4	*Less than 0.2*	*0*	*0*	*0*	*0*	*1*
M_4	*Less than 0.24*	*0*	*0.16*	*1*	*0*	*1*
M_4	*Less than 0.28*	*0*	*0.28*	*1*	*0*	*1*
M_4	*Less than 0.32*	*0*	*0.28*	*1*	*0*	*1*
M_4	*Less than 0.36*	*0*	*0.48*	*1*	*0*	*1*
M_4	*Less than 0.4*	*0*	*0.48*	*1*	*1*	*1*
M_5	*Less than 0.01*	*0*	*0*	*0*	*0*	*0*
M_5	*Less than 0.06*	*0*	*0*	*0*	*0*	*1*
M_5	*Less than 0.11*	*0*	*0*	*0*	*0*	*1*
M_5	*Less than 0.16*	*0*	*0*	*0*	*0*	*1*
M_5	*Less than 0.21*	*0*	*0*	*0*	*0*	*1*
M_5	*Less than 0.26*	*0*	*0*	*0*	*0*	*1*
M_5	*Less than 0.31*	*0*	*0.68*	*0.96*	*0*	*1*
M_5	*Less than 0.36*	*0*	*0.96*	*1*	*0*	*1*
M_5	*Less than 0.41*	*0*	*0.96*	*1*	*1*	*1*
M_5	*Less than 0.5*	*0*	*0.96*	*1*	*1*	*1*
M_6	*Less than 0.1*	*0*	*0*	*0*	*0*	*0*
M_6	*Less than 0.2*	*0*	*0*	*0*	*0*	*0*
M_6	*Less than 0.3*	*0*	*0.4*	*0.36*	*0*	*0*
M_6	*Less than 0.4*	*0*	*0.64*	*0.64*	*0.44*	*0*
M_6	*Less than 0.5*	*0*	*0.64*	*0.68*	*0.56*	*1*
M_6	*Less than 0.6*	*0*	*0.8*	*0.72*	*0.6*	*1*
M_6	*Less than 0.7*	*0*	*0.8*	*0.88*	*0.6*	*1*
M_6	*Less than 0.8*	*0*	*0.8*	*1*	*0.64*	*1*
M_6	*Less than 0.9*	*0*	*0.88*	*1*	*0.64*	*1*
M_6	*Less than 1*	*0*	*0.88*	*1*	*0.64*	*1*
M_6	*Less than 1.5*	*0*	*1*	*1*	*0.64*	*1*
M_6	*Less than 2*	*0*	*1*	*1*	*0.64*	*1*

OMPRSS were executed in three maps M_4, M_5 and M_6 given in Fig. 19, the location of the source and the goal kept at (10, 10) and (450, 450) respectively. The planners were executed 25 times per iteration of increasing sample points and the average of both cost and the success rate were recorded and arranged in certain time steps as shown in Table II and Table III for all the planners.

V. Discussion

The experimental results of OMPRSS show some intriguing outcomes. The initial experiment shows the basic working of OMPRSS in which we can observe that though the path length is not optimal, the time of planning is adequate. It also somehow tends not to waste time on C_{free} areas rather focus on obstacle dominated areas. OMPRSS has the flexibility of changing various parameters which includes the range factor which determines the area of generating the random sample points and the space for increasing or decreasing the number of sample points and the depth limit of recursion. These parameters open up more dimensions for experiments. From the theoretical point of view, though the sample points are generated randomly, the larger the area of generating points (larger range factor) the chance of getting a larger path length is higher and this holds for the other way around as well. The experiment of changing the range factor was performed as to view whether this condition would hold. Table I shows the result of changing the range factor in which cost generated by R_3 is almost always higher compared to R_1 and R_2. But there is a better part of R_3. We see this in *Sl. No.* 16 to 18 of Table I. In M_3, for source at (100, 70) and goal at (230, 200), R_1 and R_2 fails to find the path, whereas R_3 does. This shows that R3 can work better in cases where the destination is augmented by obstacles, for example, dealing with concave obstacles. On the other hand, altering the number of sample points and the depth of recursion can result in different responses. Theoretically, increasing the sample points in OMPRSS can increase the chances of getting a successful planning faster but we can see in Fig. 16 that time really does not decrease with increasing sample points, rather fluctuates. The cost also does not either increase or decrease with increasing sample points (Fig. 15). In Fig. 16, the worst case of time required for planning is about 0.175 seconds and below 0.01 seconds at best. As OMPRSS takes fairly short time for planning, the planner can be executed multiple times and the best cost can be considered. On the contrary, increasing the depth is worse than increasing the sample points. We see this in Fig. 17 and Fig. 18 where the cost on average is higher whereas the time required for planning has the probability of shooting up high with increasing depth.

The final stage of the experiment was the comparison of OMPRSS with the variants of PRM. In Fig. 20, the PRM variants start with a higher cost which gradually decreases with increasing sample points S_i, except for Bridge test PRM. The Gaussian sampler and Obstacle based gradually gets higher success rate with S_i and the uniform sampler has a consistent success rate (Fig. 22). The time on the other hand increases with S_i (Fig. 21). Whereas for OMPRSS, neither the cost and time increases or decreases with S_i, rather oscillate with a consistent success rate. And from Fig. 26 to Fig. 29 and Table II and Table III we can observe that OMPRSS tends to converge faster with adequate cost. In all the experiments, the Bridge test PRM failed in most cases as it was developed to generate sample points only at narrow corridors, due to which a single connected roadmap could not be constructed.

VI. Conclusion

This paper introduces a new approach of a deliberative sampling-based motion planning which tries to find a path from source to destination expectantly using randomly generated collision free sample points and straight lines. When the straight-line path from source to goal is collision prone, the planner tries to find the path around the obstacles recursively resulting in a planning which focuses more on obstacle outweighing areas and spending lesser time on the collision free areas. Comparison with PRM variants shows that the proposed planner gives solutions at a faster rate.

Though OMPRSS can deliver fast planning, there are downsides to the algorithm. OMPRSS mostly fails in highly obstacle-dominant maps and to plan in such maps, higher depth of recursion is required. But the planner can produce lots of sample points which can lead to unnecessary turns and also take a large amount of time when it tries to find a path around an obstacle especially when the depth is high. In addition, OMPRSS is not designed for dynamic environments. The algorithm also tends to struggle in narrow corridors especially if the length of the corridor is large. In the experiments performed, OPMRPSS is not tested in high dimensional spaces and also not tested on real robots.

Optimistic motion planning using recursive sub-sampling is at its initial phase, as a result it still has some drawbacks, like the path length can be high and somewhat far from being optimal, finding path in obstacle dominant areas can take a large amount of time and having to choose ideal parameters for different scenarios is still done manually. In addition, the experiments were performed in 2D environments using a point robot. Implementing the OMPRSS in higher dimensional environments and also addressing to the drawbacks mentioned above are the future perspectives for improving the proposed approach.

Acknowledgement

This work is funded by the Indian Institute of Information Technology, Allahabad.

References

[1] Latombe, J.-C., "Motion Planning: A Journey of Robots, Molecules, Digital Actors, and Other Artifacts," *The International Journal of Robotics Research*, 1999, 18, (11), pp. 1119-1128. doi:10.1177/02783649922067753.

[2] Kavraki, L. and Latombe, J., "Randomized Preprocessing of Configuration for Fast Path Planning," in, *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, (1994)), pp. 2138-2145 vol.2133. doi:10.1109/ROBOT.1994.350966.

[3] Elbanhawi, M. and Simic, M., "Sampling-Based Robot Motion Planning: A Review," *IEEE Access*, 2014, 2, pp. 56-77. doi:10.1109/ACCESS.2014.2302442.

[4] Lavalle, S.M., "Rapidly-Exploring Random Trees: A New Tool for Path Planning," October 1998, Technical Report,Computer Science Dept., I.S.U. (TR 98–11). doi: 10.1.1.35.1853.

[5] Tsianos, K.I., Sucan, I.A., and Kavraki, L.E., "Sampling-Based Robot Motion Planning: Towards Realistic Applications," *Computer Science Review*, 2007, 1, (1), pp. 2-11. doi:10.1016/j.cosrev.2007.08.002.

[6] Kumar, A. and Kala, R., "Linear Temporal Logic-Based Mission Planning," *International Journal of Interactive Multimedia and Artificial Intelligence*, 2016, 3. doi:10.9781/ijimai.2016.375.

[7] Montana, F.J., Liu, J., and Dodd, T.J., "Sampling-Based Reactive Motion Planning with Temporal Logic Constraints and Imperfect State Information," in Petrucci, L., Seceleanu, C., and Cavalcanti, A. (eds.), *Critical Systems: Formal Methods and Automated Verification*, (Springer International Publishing, 2017)), pp. 134-149. doi:10.1007/978-3-319-67113-0_9.

[8] Kavraki, L.E., Svestka, P., Latombe, J., and Overmars, M.H., "Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces," *IEEE Transactions on Robotics and Automation*, 1996, 12, (4), pp. 566-580. doi:10.1109/70.508439.

[9] Amato, N.M. and Wu, Y., "A Randomized Roadmap Method for Path and Manipulation Planning," in, *Proceedings of IEEE International Conference on Robotics and Automation*, (1996)), 1, pp. 113-120 vol.111. doi:10.1109/ROBOT.1996.503582.

[10] Persson, S.M. and Sharf, I., "Sampling-Based a* Algorithm for Robot Path-Planning," *The International Journal of Robotics Research*, 2014, 33, (13), pp. 1683-1708. doi:10.1177/0278364914547786.

[11] Sucan, I.A., Moll, M., and Kavraki, L.E., "The Open Motion Planning Library," *IEEE Robotics & Automation Magazine*, 2012, 19, (4), pp. 72-82. doi:10.1109/MRA.2012.2205651.

[12] Bohlin, R. and Kavraki, L.E., "Path Planning Using Lazy Prm," in, *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, (2000)), 1, pp. 521-528 vol.521. doi:10.1109/ROBOT.2000.844107.

[13] Karaman, S. and Frazzoli, E., "Sampling-Based Algorithms for Optimal Motion Planning," *The International Journal of Robotics Research*, 2011, 30, (7), pp. 846-894. doi:10.1177/0278364911406761.

[14] Dobson, A. and Bekris, K.E., "Sparse Roadmap Spanners for Asymptotically near-Optimal Motion Planning," *The International Journal of Robotics Research*, 2014, 33, (1), pp. 18-47. doi:10.1177/0278364913498292.

[15] Hsu, D., Tingting, J., Reif, J., and Zheng, S., "The Bridge Test for Sampling Narrow Passages with Probabilistic Roadmap Planners," in, *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, (2003)), 3, pp. 4420-4426 vol.4423. doi:10.1109/ROBOT.2003.1242285.

[16] Boor, V., Overmars, M.H., and Stappen, A.F.v.d., "The Gaussian Sampling Strategy for Probabilistic Roadmap Planners," in, *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, (1999)), 2, pp. 1018-1023 vol.1012. doi:10.1109/ROBOT.1999.772447.

[17] Amato, N.M., Bayazit, O.B., Dale, L.K., Jones, C., and Vallejo, D., "Obprm: An Obstacle-Based Prm for 3d Workspaces," in, *Proceedings of the third workshop on the algorithmic foundations of robotics on Robotics : the algorithmic perspective: the algorithmic perspective*, (A. K. Peters, Ltd., 1998 of Conference, Edition edn.), pp. 155–168. doi:10.1201/9781439863886-19.

[18] Geraerts, R. and Overmars, M.H., "Reachability-Based Analysis for Probabilistic Roadmap Planners," *Robotics and Autonomous Systems*, 2007, 55, (11), pp. 824-836. doi:10.1016/j.robot.2007.06.002.

[19] Geraerts, R. and Overmars, M.H., "Sampling and Node Adding in Probabilistic Roadmap Planners," *Robotics and Autonomous Systems*, 2006, 54, (2), pp. 165-173. doi:10.1016/j.robot.2005.09.026.

[20] Kuffner, J.J. and LaValle, S.M., "Rrt-Connect: An Efficient Approach to Single-Query Path Planning," in, *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, (2000)), 2, pp. 995-1001 vol.1002. doi:10.1109/ROBOT.2000.844730.

[21] Kala, R., "Rapidly Exploring Random Graphs: Motion Planning of Multiple Mobile Robots," *Advanced Robotics*, 2013, 27, (14), pp. 1113-1122. doi:10.1080/01691864.2013.805472.

[22] Janson, L., Ichter, B., and Pavone, M., "Deterministic Sampling-Based Motion Planning: Optimality, Complexity, and Performance," *The International Journal of Robotics Research*, 2017, 37, (1), pp. 46-61. doi:10.1177/0278364917714338.

[23] Solovey, K. and Kleinbort, M., "The Critical Radius in Sampling-Based Motion Planning," *The International Journal of Robotics Research*, 2019, 39, (2-3), pp. 266-285. doi:10.1177/0278364919859627.

[24] Ichter, B., Schmerling, E., Lee, T.W.E., and Faust, A., "Learned Critical Probabilistic Roadmaps for Robotic Motion Planning," in, *2020 IEEE International Conference on Robotics and Automation (ICRA)*, (2020)), pp. 9535-9541. doi:10.1109/ICRA40945.2020.9197106.

[25] Vonásek, V., Pěnička, R., and Kozlíková, B., "Computing Multiple Guiding Paths for Sampling-Based Motion Planning," in, *2019 19th International Conference on Advanced Robotics (ICAR)*, (2019)), pp. 374-381. doi:10.1109/ICAR46387.2019.8981589.

[26] Kim, D., Kwon, Y., and Yoon, S., "Adaptive Lazy Collision Checking for Optimal Sampling-Based Motion Planning," in, *2018 15th International Conference on Ubiquitous Robots (UR)*, (2018)), pp. 320-327. doi:10.1109/URAI.2018.8442203.

[27] Švestka, P. and Overmars, M.H., "Coordinated Path Planning for Multiple Robots," *Robotics and Autonomous Systems*, 1998, 23, (3), pp. 125-152. doi:10.1016/S0921-8890(97)00033-X.

[28] Clark, C.M., "Probabilistic Road Map Sampling Strategies for Multi-Robot Motion Planning," *Robotics and Autonomous Systems*, 2005, 53, (3), pp. 244-264. doi:10.1016/j.robot.2005.09.002.

[29] Yao, Z. and Gupta, K., "Path Planning with General End-Effector Constraints," *Robotics and Autonomous Systems*, 2007, 55, (4), pp. 316-327. doi:10.1016/j.robot.2006.11.004.

[30] Kala, R., "Sampling Based Mission Planning for Multiple Robots," in, *IEEE Congress on Evolutionary Computation (CEC)*, (2016)), pp. 662-669. doi:10.1109/CEC.2016.7743856.

Lhilo Kenye

Lhilo Kenye received his B.Tech. degree in Information Technology from School of Engineering and Technology, Nagaland University, India in 2014. He received his M.Tech. degree in Information Technology, specialization in Robotics from Indian Institute of Information Technology, Allahabad, India in 2016. He is currently pursuing PhD in Information technology, specialization in Robotics, from Indian Institute of Information Technology, Allahabad, India. His current work is mainly in the field of robot navigation and computer vision.

Rahul Kala

Rahul Kala received the B.Tech. and M.Tech. degrees in Information Technology from the Indian Institute of Information Technology and Management, Gwalior, India in 2010. He received his Ph.D. degree in cybernetics from the University of Reading, UK in 2013. He is currently working as an Assistant Professor in the Indian Institute of Information Technology, Allahabad, India. He is the author of four books and over 100 papers. He is a recipient of the Early Career Research Grant from the Department of Science and Technology; Best PhD dissertation award from the IEEE ITS Society; and Commonwealth scholarship from the British Government.

Multimodal Human Eye Blink Recognition Using Z-score Based Thresholding and Weighted Features

Puneet Singh Lamba[1,2], Deepali Virmani[3], Manu S. Pillai[2], Gopal Chaudhary[2]*

[1] University School of Information, Communication and Technology, GGSIPU, Sector 16 C, Dwarka, Delhi (India)
[2] Bharati Vidyapeeth's College of Engineering, Paschim Vihar, New Delhi (India)
[3] Vivekananda Institute of Professional Studies-Technical Campus, New Delhi (India)

Received 15 March 2021 | Accepted 27 August 2021 | Published 8 November 2021

LA UNIVERSIDAD
EN INTERNET

Abstract

A novel real-time multimodal eye blink detection method using an amalgam of five unique weighted features extracted from the circle boundary formed from the eye landmarks is proposed. The five features, namely (Vertical Head Positioning, Orientation Factor, Proportional Ratio, Area of Intersection, and Upper Eyelid Radius), provide imperative gen (z score threshold) accurately predicting the eye status and thus the blinking status. An accurate and precise algorithm employing the five weighted features is proposed to predict eye status (open/close). One state-of-the-art dataset ZJU (eye-blink), is used to measure the performance of the method. Precision, recall, F1-score, and ROC curve measure the proposed method performance qualitatively and quantitatively. Increased accuracy (of around 97.2%) and precision (97.4%) are obtained compared to other existing unimodal approaches. The efficiency of the proposed method is shown to outperform the state-of-the-art methods.

Keywords

Eye Blink, Multimodal, Z score Threshold, Weighted Features.

DOI: 10.9781/ijimai.2021.11.002

I. Introduction

Eye blinking is partly unintended closing and reopening of the eyelid. Muscles that help in closing/opening the eye are Orbicularis oculi and levator palpebrae superioris [1]. Eye blinking assists in cleaning and moistening the eye cornea. Blinking can be categorized into three main classes: spontaneous blinking, reflex blinking, and voluntary blinking. The first two lie under the category of involuntary blinking. On the contrary, voluntary blinking can be invoked intentionally within the control of a subject. The work emphasizes detecting the eye blinks, specifically from voluntary blinking, which can be encrypted to alarm emergency [2]-[3].

Due to its importance in many applications (driver drowsiness, human-computer interaction, micro-expression detection), an exponential rise is seen in the research field related to eye blinking detection. Many innovative, unconventional, and robust methods have been stated in the literature [4]-[11]. In the current scenario, multiple robust real-time facial landmark detectors [12]-[15] that arrest virtually all the distinctive features on a human face are accessible. Several approaches have been proposed to routinely detect blinks [9],[12] from the eyes (from both still [16] and video sequences [17]-[19]). But the disadvantage is that they impose rigid requirements on the setup, in the sense of a relative head alignment, resolution, illumination, etc. This work puts forward an effective method to classify eye-blinks [20]-[22] to overcome the stated complications, extracting five distinct weighted features (explained in detail in section III). In this work, using the unique circles formed from the eyelids landmarks, five novel features (depicting imperative information) have been extracted, which are further assigned weights, thus resulting in an accurate eye blink detection system. The motivation and research question behind the work is depicted in Fig. 1.

A. Key Contribution of the Paper

A novel multimodal approach for eye blink detection using low-level eyelid feature extraction and z-score thresholding is proposed.

i) Accurate face detection and eye localization method is used (MTCNN – Multitask Cascaded Neural Network).

ii) Unique circles are formed from the eyelids landmarks.

iii) The eye blinks are detected using an amalgam of 5 weighted features (Vertical Head Positioning, Orientation Factor, Proportional Ratio, Area of Intersection, and Upper Eyelid Radius) depicting imperative gen (z score threshold), extracted from the circles uniquely formed from the eyelids landmarks.

iv) A fusion technique is described to fuse the extracted five features into a single binary signal.

v) Finally, a Z-score-based thresholding algorithm is proposed to extract peaks from the signal where peaks correspond to eye blinks.

The rest of the work is structured as: Section I familiarizes the blink concept and the essential contributions. Section II reviews the related literature. Feature Extraction of the multimodal eye blink system is elaborated in section III. The methodology used, database, performance parameters, investigational setup, and blink prediction are described in section IV. Experimental results and conclusions are analyzed in sections V and VI.

* Corresponding author.

E-mail address: gopal.chaudhary88@gmail.com

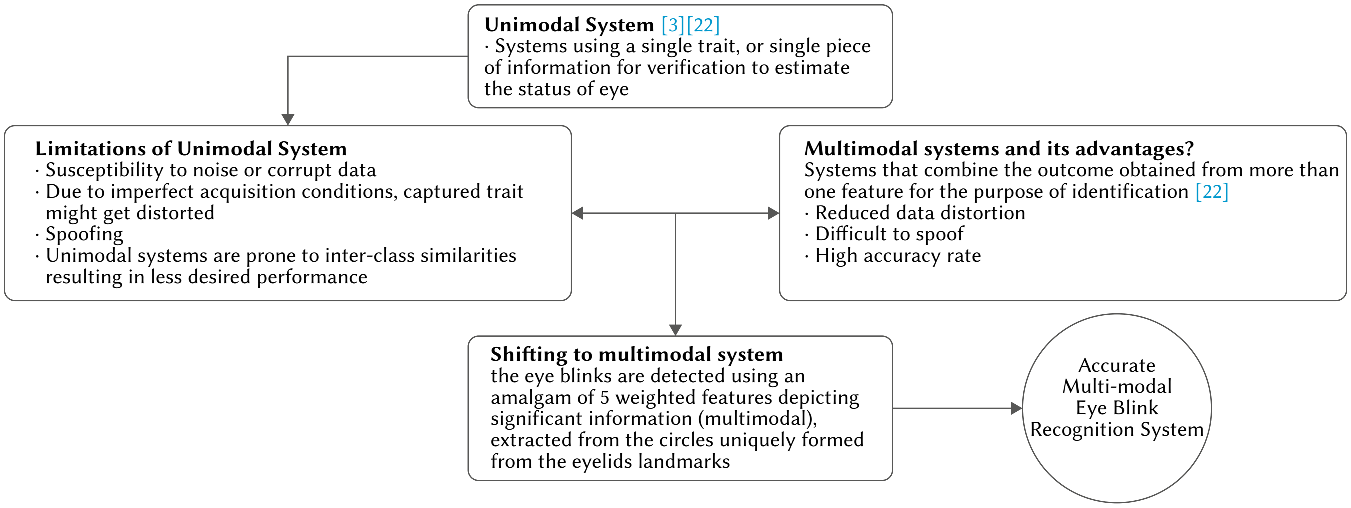

Unimodal System [3][22]
· Systems using a single trait, or single piece of information for verification to estimate the status of eye
Limitations of Unimodal System
· Susceptibility to noise or corrupt data
· Due to imperfect acquisition conditions, captured trait might get distorted
· Spoofing
· Unimodal systems are prone to inter-class similarities resulting in less desired performance
Multimodal systems and its advantages?
Systems that combine the outcome obtained from more than one feature for the purpose of identification [22]
· Reduced data distortion
· Difficult to spoof
· High accuracy rate
Shifting to multimodal system
the eye blinks are detected using an amalgam of 5 weighted features depicting significant information (multimodal), extracted from the circles uniquely formed from the eyelids landmarks
Accurate Multi-modal Eye Blink Recognition System

Fig.1. Motivation and research question.

TABLE I. Related Work With Performance Attributes

S. no.	Author and Year	Year	Dataset Used	Performance	Method employed
Z1	Michael Chau and Margrit Betke. 2005 [39]	2005	Eight test subjects	Accuracy: 95.3	Templates (open/close eye) are used in determining the state of an eye.
Z2	Diego Torricelli et al. 2009 [40]	2009	ZJU eye blink dataset	Accuracy: 95.7	Eye-Blinking is sensed without any constraints on the head pose or target position concerning the camera.
Z3	W.O. Lee et al 2010 [8]	2010	ZJU eye blink dataset	Precision: 94.4	Multiple features are extracted to sense the state of the eye. 1) the fraction of elevation breadth of the eye 2) the aggregate alteration of the no. of black pixels around the eye area using an adaptive threshold in consecutive image frames.
Z4	Kohei Arai and Ronny Mardiyanto. 2010 [21]	2010	Individual users used	N.A	Gabor filters are applied to detect the eye arcs, and eye-blink is sensed by figuring the arcs' distance.
Z5	Tomas Drutarovsky and Andrej Fogelton. 2014 [5]	2014	ZJU eye blink dataset	Accuracy: 93.45	Vertical motions in the eye are examined to sense the blinks.
Z6	Fengyi Song et al. 2014 [6]	2014	ZJU eye blink dataset	Accuracy: 96.8	The strength of multiple features collectively characterizes information (eye patches).
Z7	Tereza Soukupova and Jan Cech. 2016 [1],[7]	2016	ZJU eye blink dataset	N.A	Eye Aspect Ratio (EAR) is calculated to determine the eye status (open/ closed) in a video sequence
Z8	Federico M. Sukno et al. 2016 [20]	2016	AV@CAR	N.A	The blink detection procedure outputs freq, durations, and some fraction metrics.
Z9	P. Singh and D. Virmani. 2018 [3]	2018	ZJU eye blink dataset	Accuracy: 97	The blinks are perceived using a 13-dim sequential window calculated using eye facet correlation (Ratio of diagonal distance and width of eye landmarks), which outclasses the threshold method and [7].
Z10	P. Singh and D. Virmani. 2020 [22]	2020	ZJU eye blink dataset	Accuracy: 99.02 Precision: 99.65	A precise multimodal eye-blink recognition method using feature-level fusion (MmERMFLF) to detect the eye status.
Z11	Al-gawwam, S., & Benaissa, M. [37]	2018	ZJU eye blink dataset, eyeblink and talking face	Precision: 100%, 96.65% and 98.38%	The method approximates the facial-landmark positions and excerpts the vertical distance between eyelids. A Savitzky–Golay (SG) filter is used to level the signal while keeping the peak info to sense eye-blinks.

II. Related Study

In recent times a lot of research has been conducted in different biometric modalities wherein different modalities have proven their prevalence. Recognition systems (RS) (gait, voice, and face) are efficient but lack one or the other aspects. In Gait RS data procurement is ineffectual. Voice RS is an effectual process; but info procurement is a cumbrous errand. Face RS is burdened by the inevitable dispute of aging. But due to their stability and efficiency, eye blink recognition systems does not suffer from the limitations mentioned above. Table I reveals prior work based on functionality engaged in detecting blinks. Z1–Z11 uses the ZJU dataset to detect the blinks accurately. A solitary feature engages in computing an eye status in [Z1, Z2, Z4, Z5, Z7–Z9, Z11]. While multiple features are used in [Z3, Z6, and Z10] to sense blinks, subsequently creating a multimodal environment. This work emphasizes feature-level fusion (FLF) of weighted features extricated to sense the eye blinks accurately.

In recent times eye-blink has been detected using two primary techniques: electrooculography (EOG) [23]-[24] and videooculography (VOG) [25]-[34]. Ag/AgCl electrodes are used in the former technique and equipped around the eye area to detect eye movements and blinks [23], [35], [36], [38]-[43]. Any movement (left, right, up, down) can be perceived. However, the variability of electrooculography rests on factors that are hard to control [24]. The electrode placement also causes troublesomeness for the user, and the method is relatively costly [28]. In the latter technique, the subject is in sight of the line of the camera installed, and the frames are processed one by one, resulting in the eye's final status (open/close) using an algorithm [3], [22]. All significant research in this area nowadays uses the latter method (VOG). Several VOG-based blink detector techniques are available but lack in one or the other area. This paper proposes a novel multimodal approach for eye blink detection using low-level eyelid feature extraction and z-score thresholding. Firstly, accurate face detection and eye localization method are used (MTCNN). Multitask Cascaded Neural Network or MTCNN is a convolutional neural network architecture jointly trained for facial landmark detection and alignment, achieving superior accuracy over the state-of-the-art techniques on the challenging FDDB and WIDER FACE benchmark for face detection, and AFLW benchmark for face alignment, while keeping real time performance [44]-[46].

Further, unique circles are formed from the eyelid's landmarks, and eye blinks are detected using an amalgam of 5 weighted features (Vertical Head Positioning, Orientation Factor, Proportional Ratio, Area of Intersection, and Upper Eyelid Radius) depicting imperative gen (z score threshold). A unique fusion technique is described to fuse the extracted five features into a single binary signal. Finally, a Z-score-based thresholding algorithm is proposed to extract peaks from the signal where peaks correspond to eye blinks. No work has been done on sensing blinks with low-level eyelid feature extraction and z-score thresholding as per our familiarity. Neither multiple features (multimodal system) have been employed to detect the eye status (open/close) from video sequences.

III. Feature Extraction Using Multimodal Eye Blink Detection System

For sensing the eye blinks, feature extraction remains the most prominent step. This section focuses on detecting the face and eyes using landmarks and feature extraction. Face and eye detection using landmarks are explained in sec. A and the proposed feature extracted are explained in detail and relegated in sec. B.

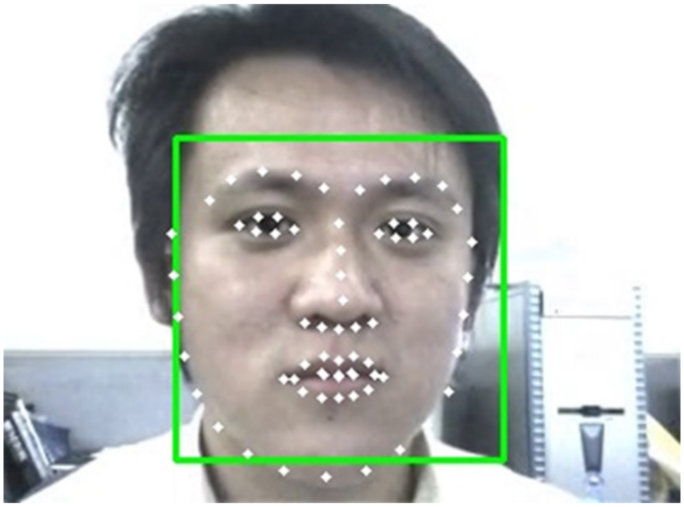


Fig. 2. Face landmarks extracted from the input video stream (ZJU dataset).

A. Face Detection and Eye Landmark Prediction

Traditional systems and approaches use techniques that directly localize eyes in an image or prefer detecting face regions first and then localizing eyes within the detected face region. The proposed approach in this work increases true positives for blinking detection and reduces false positives. An accurate eye blink detection system begins from accurate face detection and eye localization, for which we chose the widely used face detection deep learning model MTCNN. Once the face is localized from the input image, we extract eyelid landmarks from the face region detected using dlib, as shown in Fig. 2.

B. Feature Extraction

For extracting features, the landmarks around the eyelids are used. The 12 landmarks detected as the eyelids (6 for each eye) are separated into upper eyelid landmarks and lower eyelid landmarks. The four edge landmarks of the eyes belong to both groups. We first calculate the line's midpoint joining the two points lying on the eyelids in each group. Then, along with the edge points and the midpoint just calculated, a unique circle that passes through these 3 points is created. Fig. 3 shows the entire procedure where white dots are the original landmarks and the line's midpoint is shown in red. Circle formation for both open and closed eyes are depicted in Fig. 4.

1. Feature Extracted

Five sets of features are extracted from the four circles generated:

a) Vertical Head Positioning (VHP)

For approximating the positioning of the head/eyes, we calculate the distance between the center of the upper and lower eyelid circles for both the eyes.

The mean of these two distances gives *VHP*:

$$VHP = \frac{l_1+l_2}{2} \qquad (1)$$

where l_1 and l_2 are the distances between centers of the left and right eye circles as shown in Fig. 5.

b) Orientation Factor (OF)

For calculating the head/eyes orientation, we calculate the distance between the center of the upper eyelid circle and lower eyelid circle of the left eye and upper eyelid circle and lower eyelid circle of the right eye, respectively.

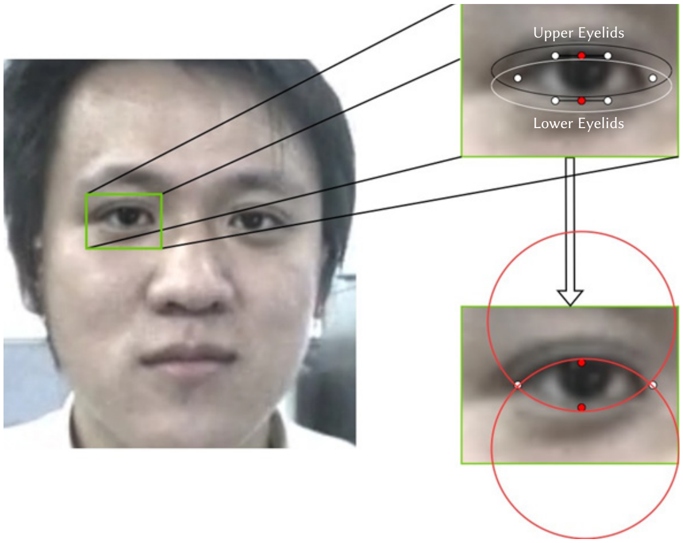


Fig. 3. Upper and lower eyelid landmarks and the circle generated using the points.

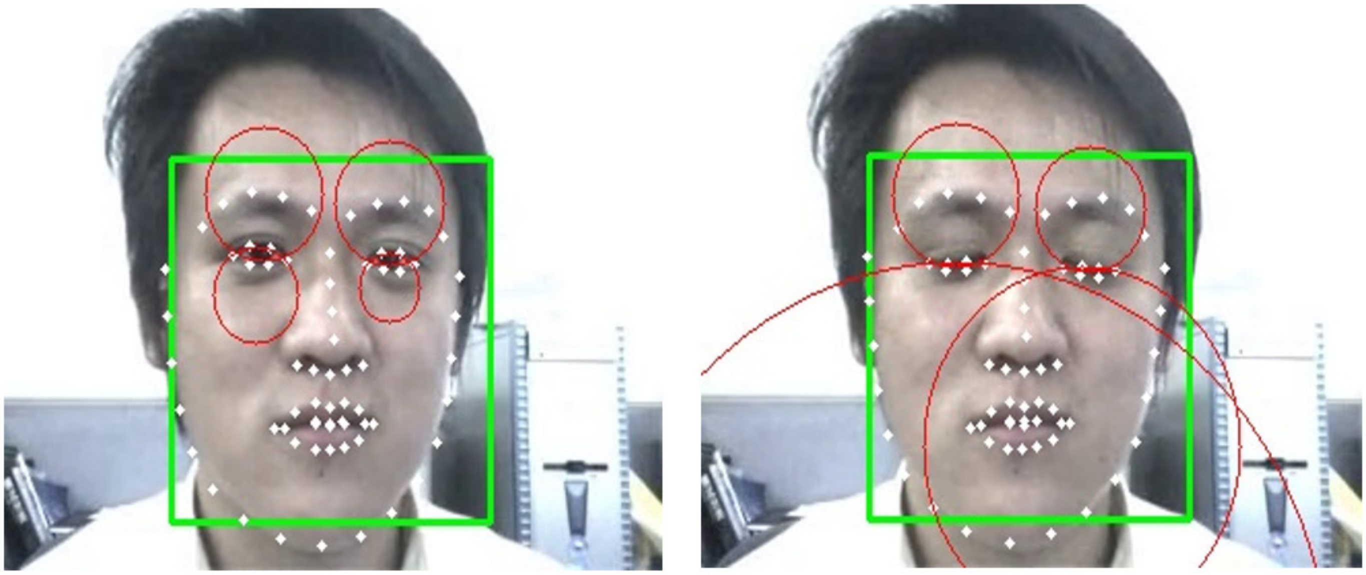


Fig. 4. Circles generated using both the eyelid landmarks for open (left) & closed (right) eyes.

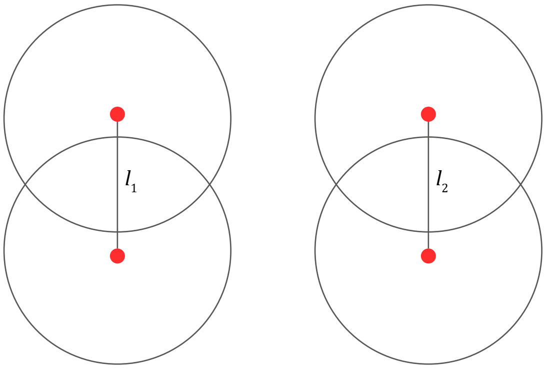


Fig. 5. Distance between the centers of the left and right eye.

The ratio of these two distances gives *OF*:

$$OF = \frac{l_1}{l_2} \tag{2}$$

where l_1 and l_2 are distances between the centers of the circles as shown in Fig. 6.

c) Proportional Ratio (PR)

To calculate the circle size proportional to the triangle perimeter formed by the eyelid's landmarks, we calculate the upper eyelid circle radius to both eyes' lower radius eyelid circle.

The mean of these ratios gives *PR* as follows:

$$PR = \frac{r_1+r_2}{2} \tag{3}$$

$$r_1 = \frac{lr_{up}}{lr_{down}} \tag{4}$$

$$r_2 = \frac{rr_{up}}{rr_{down}} \tag{5}$$

where lr_{up}, lr_{down}, rr_{up} and rr_{down} are radius of left upper eyelid circle, left lower eyelid circle, right upper eyelid circle and right lower eyelid circle respectively as depicted in Fig. 7.

d) Area of Intersection (AOI)

AOI is the area of intersection between the upper eyelid circle and lower eyelid circle of either eye. In our work, we have used the *AOI* of the left eye. For experiment purposes, either of the eyes can be used.

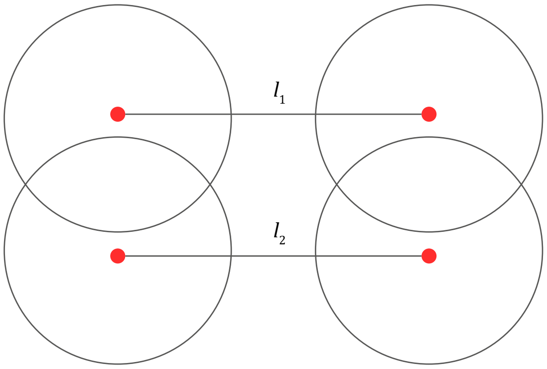


Fig. 6. Distance between the center of the upper eyelid circle and lower eyelid circle of the left eye and upper eyelid circle and lower eyelid circle of the right eye.

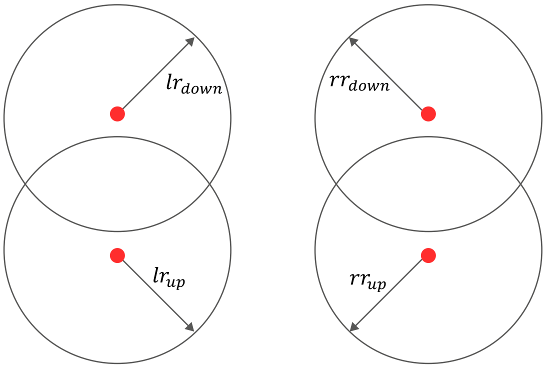


Fig. 7. Radius of left upper eyelid circle, left lower eyelid circle, right upper eyelid circle, and right lower eyelid circle.

We can also take the mean of *AOI* of both the eyes. Fig. 8 shows the *AOI* of an upper and lower eyelid.

e) Upper Eyelid Radius (UER)

UER is the upper eyelid radius of either eye. In our experiments, we have used *UER* of the left eye. Either of the eyes can be used for experiments, as in the case of *AOI*. We can also take the mean of *UER* of both the eyes.

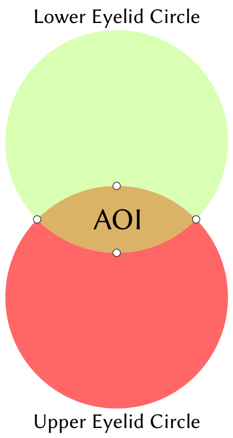


Fig. 8. The *AOI* of an upper and lower eyelid.

IV. Research Methodology Used

This section focuses on the methodology that is employed by the method for sensing the blinks. First of all, the dataset employed for the experimental purpose is discussed (section A). Then the performance parameters that judge the efficiency of the method are explained in section B. Experimental setup, blink detection/prediction are explained in detail in subsequent sections.

A. Dataset Used

One state-of-the-art dataset ZJU (eyeblink) [18], is used to evaluate the proposed method's performance. The dataset consists of 80 video sequences of 20 users, as depicted in Table II. All the video sequences (4 to 5 sec) are recorded in 30 fps having two to six voluntary blinks. In total, there are 255 blinks in the entire dataset.

TABLE II. ZJU Eye-Blink Dataset

No. of Users	Four Clips per person			
	No. of Clips	View	Spectacles	Total Blinks
20	1	Front	N	255
	1	Front	Y	
	1	Front	Y	
	1	Front	N	

B. Performance Parameters

In this paper, eye blinks are detected using an amalgam of 5 weighted features depicting imperative gen (z score threshold). A unique fusion technique is described to fuse the extracted five features into a single binary signal. Finally, a Z-score-based thresholding algorithm is proposed to extract peaks from the signal where peaks correspond to eye blinks. The performance of the method discussed is evaluated using several parameters as shown below:

1. ***Recall* (*R*)** can be defined as the ratio of the correct categorized cases of a class to the total number of class cases.

$$P = \frac{TP}{TP + FN} \tag{6}$$

2. ***Specificity* (*Sp*)** It deals with the proportion of actual negatives (true) that are correctly identified as such.

$$Sp = \frac{TN}{FP + TN} \tag{7}$$

3. ***Precision* (*P*)** can be defined as the ratio of correct positive predictions to the total number of positive predictions.

$$P = \frac{TP}{TP + FP} \tag{8}$$

4. ***Negative Predictive Value* (*NPV*)**: The ratio of accurate negative class predicted to the over-all amount of negative predictions.

$$NPV = \frac{TN}{TN + FN} \tag{9}$$

5. ***False Positive Rate* (*FPR*)**: The ratio of false-positive class predictions to the overall negatives.

$$FPR = \frac{FP}{FP + TN} = 1 - Sp \tag{10}$$

6. ***False Discovery Rate* (*FDR*)**: The ratio of false-positives predictions to the overall positives.

$$FDR = \frac{FP}{FP + TP} \tag{11}$$

7. ***False Negative Rate* (*FNR*)**: The ratio of false-negatives predictions to the aggregate number of positives.

$$FNR = \frac{FN}{FN + TP} \tag{12}$$

8. ***Accuracy*** can be defined as the ratio of correctly classified instances to the total number of instances.

$$P = \frac{TP + TN}{P + N} \tag{13}$$

9. ***F1 Score* (*F1*)** can be defined as weighted avg. of precision and recall

$$P = \frac{2 * TP}{* TP + FP + FN} \tag{14}$$

10. ***AUC Score*** is used to measure the performance of a model at different threshold settings.

(**Note**: TP: True +, TN: True -, FP: False +, and FN: False -)

C. Experimental Setup

We have used ZJU (eye-blink) dataset (explained in sec. A) for our experiments. The five extracted set of features supports blinking with sudden peaks in their respective plots. Fig. 9 shows the plot (five extracted features (explained in sec. III) vs. the number of frames) from one of the dataset's input videos.

It can be inferred from the plots that frames 36 to 43 exhibit a peak in all four plots except Orientation Factor (OF) plot. It remains uniform with a value greater than 1, indicating standard behavior of other features. Similar trends can be seen at frames 85 to 92 and 130 to 140, indicating blinks at those frames in the input video sequence. The case discussed in Fig. 9 is a clear case where all the features show a positive consent as the subject is sitting in the camera's line of sight and not wearing any spectacles. On the contrary, Fig. 10 shows the plot of all five features for another input video sequence of the dataset with disturbance and other perturbations. The subject in the input video sequence is blinking in a dark background. Here, feature OF is not uniform, and in certain intervals, its value drops below 1. This behavior is an indication that other features are not behaving normally and are affected by the perturbation in the input video sequence, except Upper Eyelid Radius (UER), which still produces peaks at four intervals supporting blinks in the video input. With further experiments, we found that when the value of Orientation Factor (OF) drops below 1, Upper Eyelid Radius (UER) is the most dominant one to report a blink with a peak, while the other three features produce sloppy peaks. We used weighted signal averaging or convolution conditioned on OF to implement this behavior in an automated system. The same behavior is seen when the subject is wearing spectacles also.

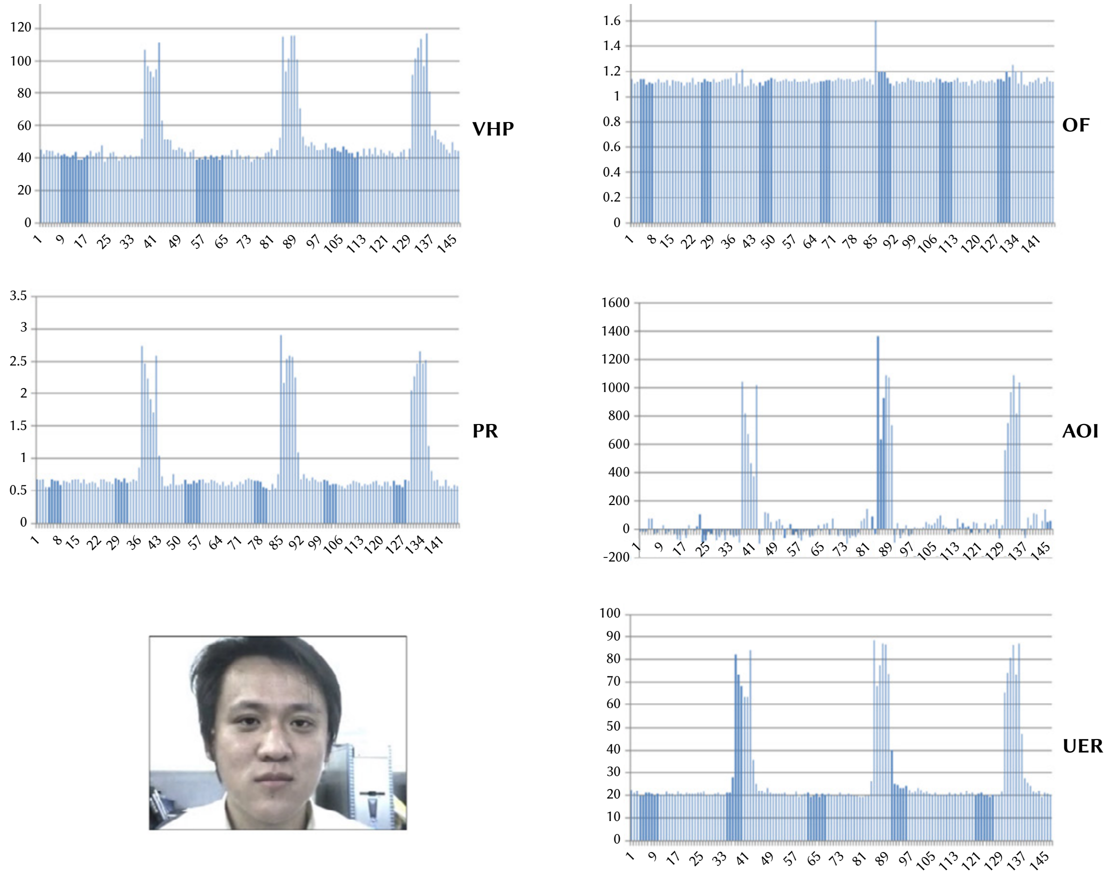


Fig. 9. The plot of all five features extracted vs. frames from an input video sequence along with a frame from the video.

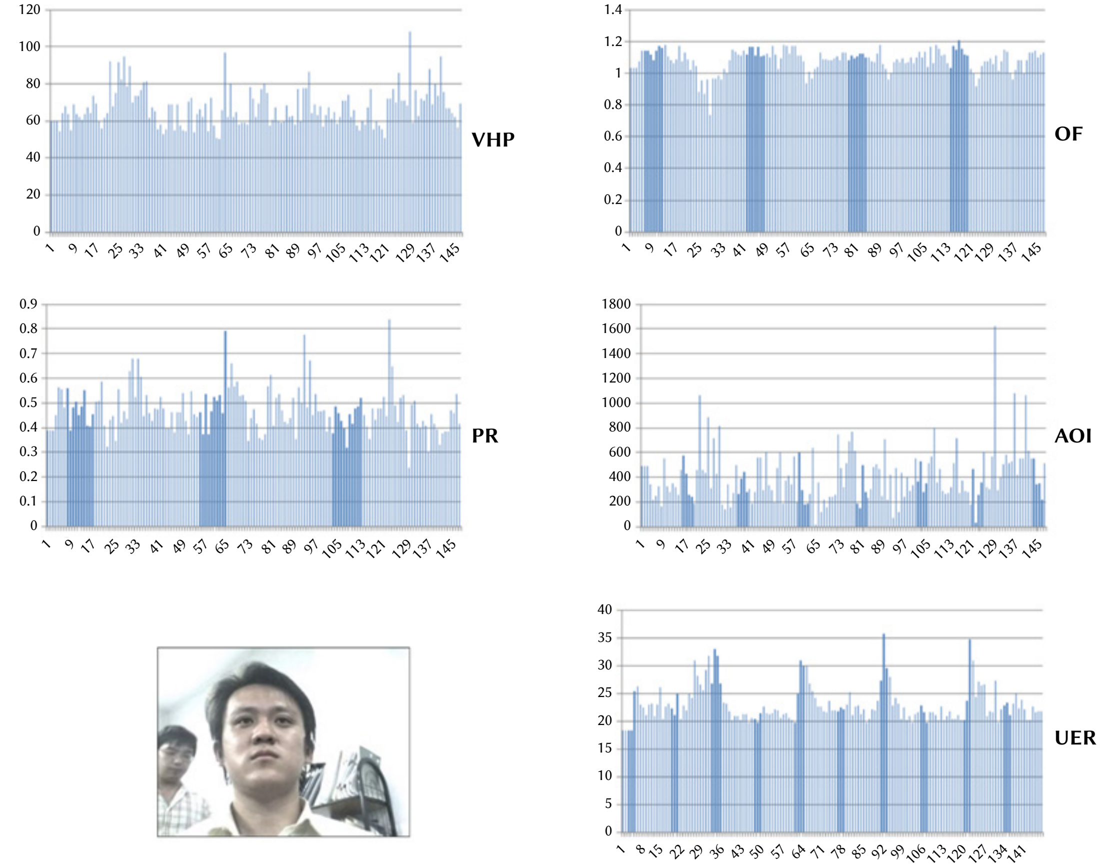
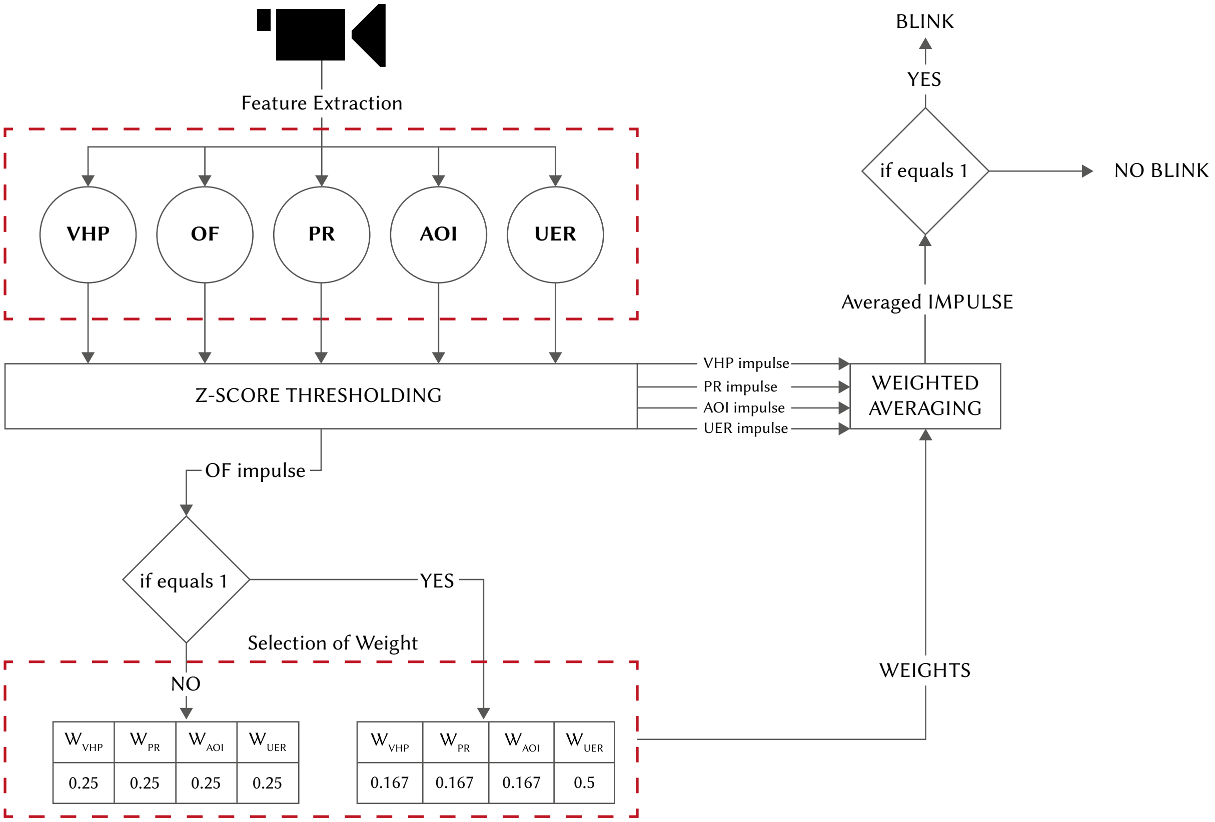


Fig. 10. The plot of all five features for an input video sequence with disturbance and other perturbations along with a frame from the video.



W_{VHP}	W_{PR}	W_{AOI}	W_{UER}
0.25	0.25	0.25	0.25

W_{VHP}	W_{PR}	W_{AOI}	W_{UER}
0.167	0.167	0.167	0.5

Fig. 11. Flowchart of the multimodal eye-blink detection system.

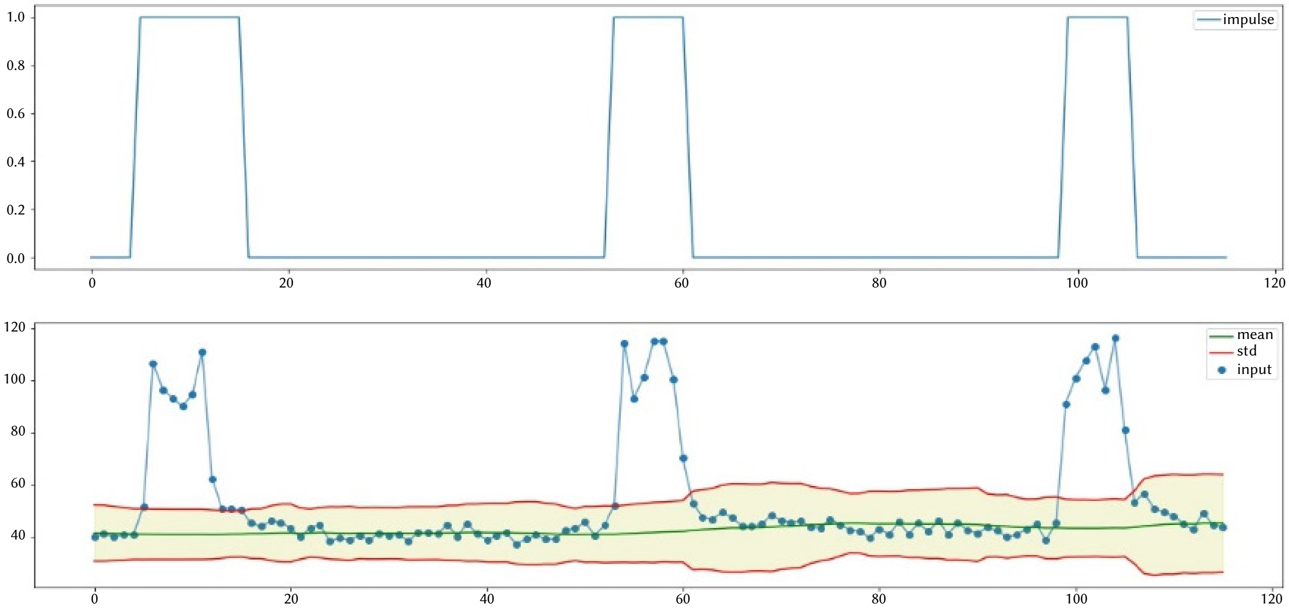


Fig. 12. An arbitrary input signal and its generated unit impulse signal using the method.

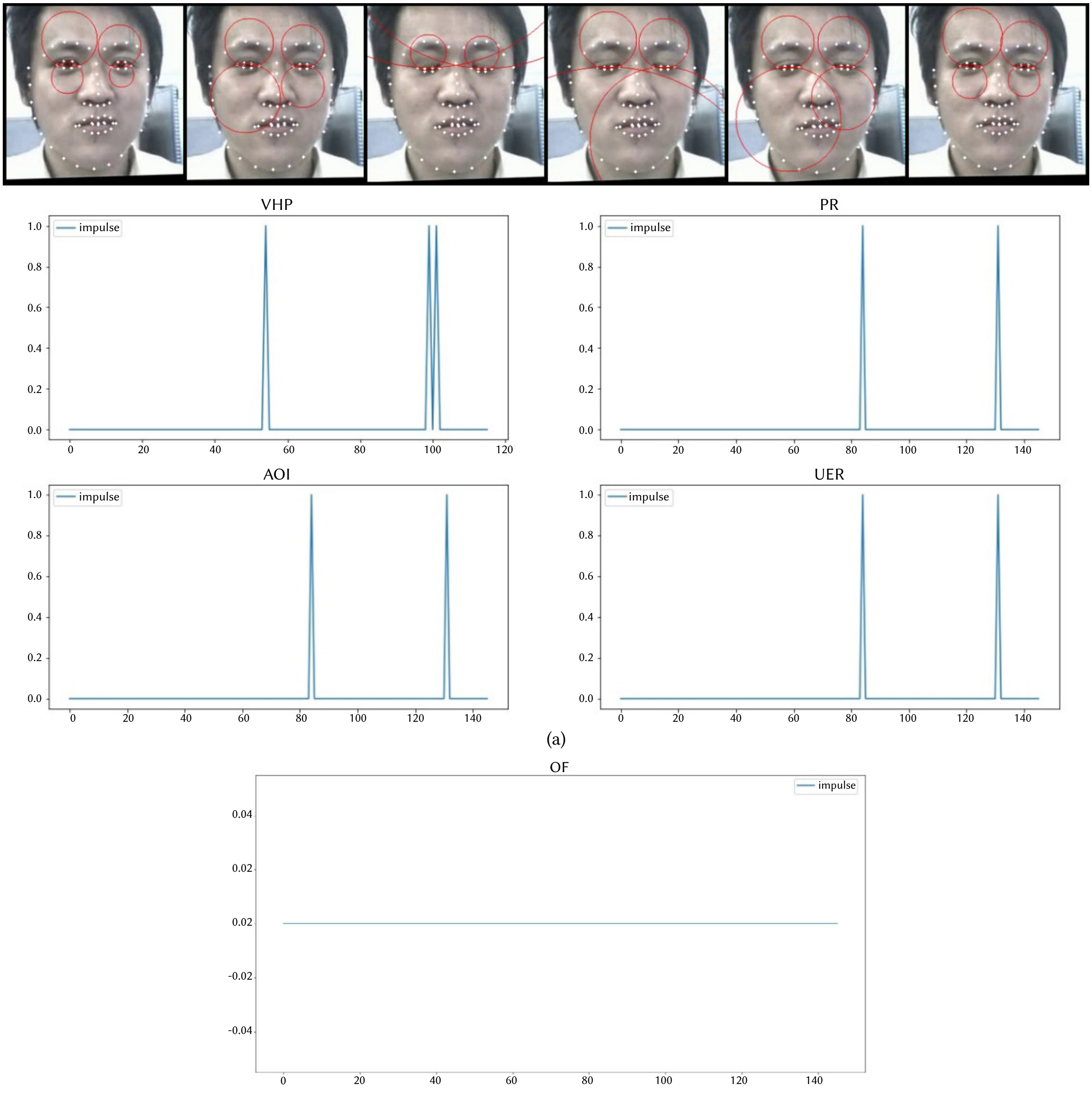
D. Blink Detection

Automated blink detection is a methodology consisting of peak detection and weighted averaging. We generate a unit impulse signal for peaks in all four features individually and interpolate these signals conditioned on the impulse signal of Orientation Factor (OF). Fig. 11 depicts the entire workflow with a flowchart.

a) **Peak Detection**: A simple way to detect peaks in a signal is hard thresholding with a predefined value. But in our experiments, we found that the scale of the extracted features depends vastly on multiple factors, like, camera position, head positioning, landmark detector accuracy, etc. Therefore, direct hard thresholding won't be able to detect peaks for our input in real-time settings. Therefore, we derived a more robust, yet simple z-score based peak detection algorithm inspired by probability distributions properties [41]-[43] for peak detection. Rather than keeping a set threshold value for signaling peaks in a given input signal, the algorithm calculates the moving mean and moving standard deviation of the input signal. When a data point lies the preset threshold times the standard deviation away from the mean, the algorithm signals a peak. The approach of calculating a moving mean and standard deviation keeps the method's integrity at different scales. The algorithm can be made more robust to outliers and signal length by calculating the moving mean and standard deviation by only looking at the last 'k' values, where k is a whole number. Fig. 12 shows an arbitrary input signal and its generated unit impulse signal using our method.

b) **Weighted averaging of signals**: Fig. 13(a) shows the impulse signals of VHP, PR, AOI, and UER, and Fig. 13(b) shows the impulse signals of OF generated using hard thresholding at 1 (if the signal is less than 1, an impulse is generated). When there is an impulse in OF, it means that other features except UER will struggle to report a blink with a peak, whereas when OF has no impulse signals, it suggests the normal behavior which peaks in all other four features. Fig. 14(a) and Fig. 14(b) show the same feature impulses for an input video sequence with abnormal behavior. Table III shows the feature weights we used to average the impulse signals.

For OF = 0, we have averaged over all the four impulse signals, and for OF = 1, we have given UER a weight of 0.5, and the rest three impulse signals are given equal weightage out of 0.5, i.e., 0.5/3 = 0.167.



(a)

(b)

Fig. 13(a). The impulse signals of VHP, PR, AOI and UER. (b). The impulse signals of OF generated using hard thresholding at 1 (if the signal is less than 1, an impulse is generated).

TABLE III. The Feature Weights We Used to Average the Impulse Signals

	if OF = 0				if OF = 1			
	W_{VHP}	W_{PR}	W_{AOI}	W_{UER}	W_{VHP}	W_{PR}	W_{AOI}	W_{UER}
w_i	0.25	0.25	0.25	0.25	0.167	0.167	0.167	0.5

Equation (15) gives the final output averaged signal that reports a peak for a blink in the input video sequence. The step-by-step process is depicted in algorithm 1.

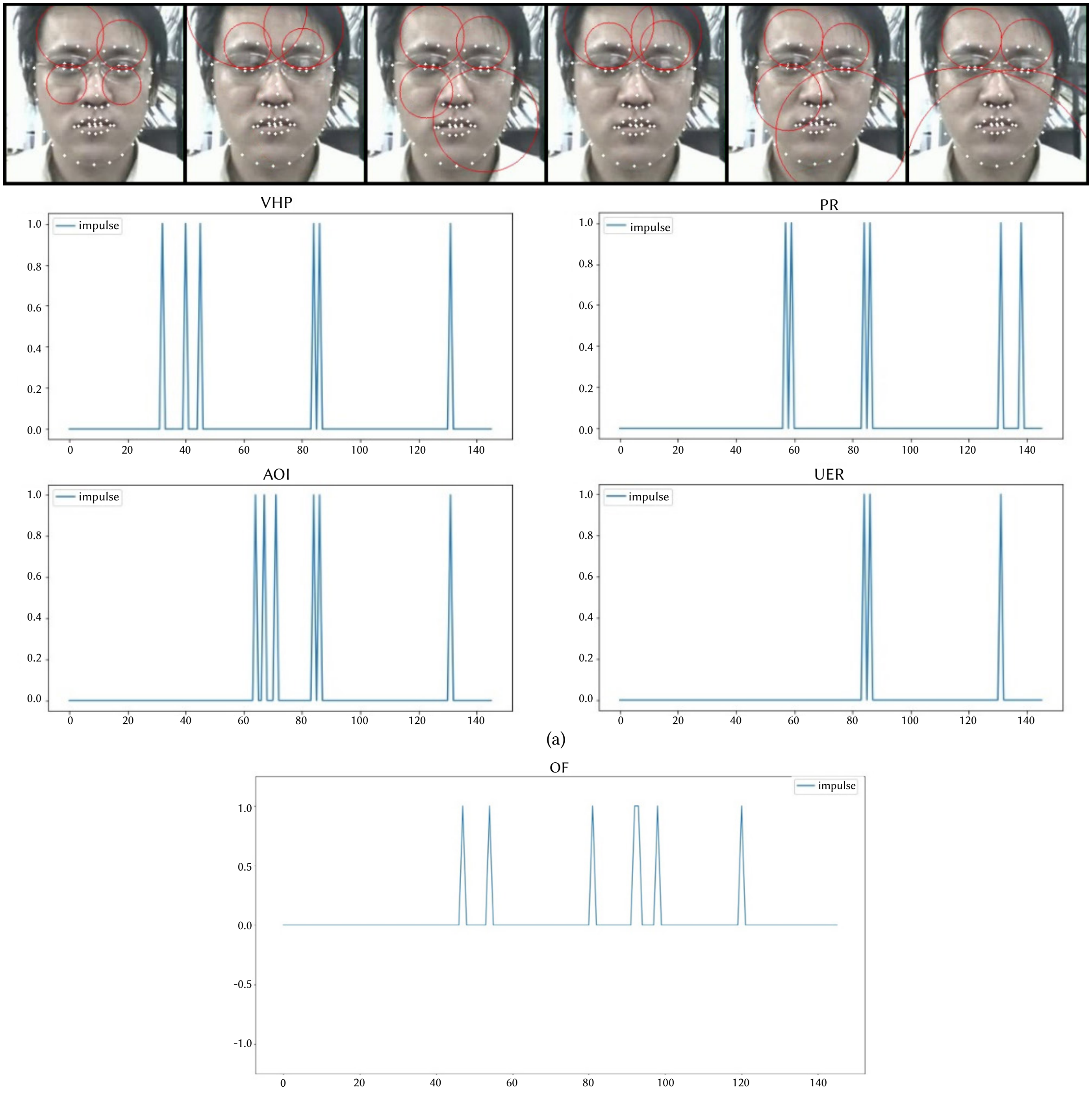
$$y^t = \sum_{i=1}^{4} W_i x_i^t \qquad (15)$$

Where y^t: output averaged signal at time t

W_i: the weight of feature i from Table III, which is conditioned on OF feature at time t.

E. Blink Prediction

Once the final signal y^t is obtained, we hard threshold the signal with a value 0.75 such that when the signal value is greater than or equal to 0.75, a unit impulse is considered, and a blink is predicted. The threshold we used favors predicting a blink only when 75% of features



(a)

(b)

Fig. 14(a). The impulse signals of VHP, PR, AOI and UER. (b). The impulse signals of OF generated using hard thresholding at 1(if the signal is less than 1, an impulse is generated).

report a peak. Future implementations can use various threshold values for the confidence of prediction while incorporating more handcrafted and deep learning extracted features.

Algorithm 1: Algorithm for Multimodal Eyeblink Detection system:

```
1. Capture video stream frames from the input camera.
2. For each frame:
    a. Localize face from the frame using Multitask Cascaded Neural Network (MTCNN).
    b. Detect Eye landmarks from the extracted face region of the frame.
3. For each eye:
    a. Separate the 12 eye landmarks detected into the upper eyelid and lower eyelid landmarks.
    b. For each group, calculate the midpoint of the line joining the two points lying on the eyelids.
    c. Then, along with the edge points and the midpoint just calculated, create a unique circle that passes through these 3 points using the algorithm presented in appendix A.
    d. Using the two circles created, calculate the five sets of features: Vertical Head Positioning, Orientation Factor, Proportion Ratio, Area of Intersection, and Upper Eyelid Radius.
    e. Perform Z-Score thresholding on all these five feature sets.
    f. Select the set of weights to be applied to normalize the impulse signals resulting from the z-score thresholding using table III and perform the weighted averaging.
4. Combine the final signal for both the eyes using signal averaging. (Optional)
5. Report a blink if the final signal's magnitude is greater than the threshold (0.75) chosen.
```

V. Experimental Results

A multimodal human eye blink detection system is implemented using five weighted features, namely Vertical Head Positioning, Orientation Factor, Proportional Ratio, Area of Intersection, and Upper Eyelid Radius. The five unique features are extracted from a unique circle formed from eye landmarks. The features extracted are used to depict the positioning and orientation of a subject's head/eyes being tested. While observing the features experimented on ZJU dataset, two sets of patterns have resulted, as shown in Fig. 9 and 10. In Fig. 9, a clear pattern is observed wherein all features contributed equally and showed expected yields. Whereas in Fig. 10, a different pattern is observed where the UER feature is given more weightage than the rest of the features. A clear pattern is observed when the subject is not wearing spectacles in decent light conditions.

On the contrary, when a subject is wearing spectacles or the light conditions are not good, a blurred pattern is observed. A multimodal eyeblink detection system is proposed and implemented to balance out the two patterns, as shown in Fig. 11. For testing the proposed system's performance, a state-of-the-art dataset, ZJU, is used. Table IV shows how the proposed system has performed in terms of performance parameters. A reputable accuracy and precision of 97.2% and 97.4% are achieved. For testing purposes, 60% of the dataset is employed. ROC graph along with an AUC score of 0.972 resulted is shown in Fig. 15.

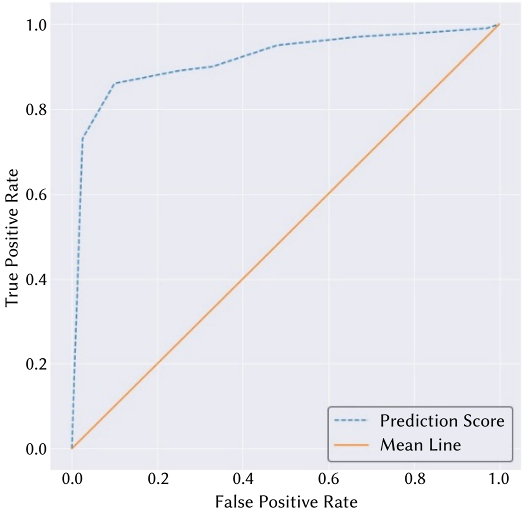


Fig. 15. ROC Curve.

A. Analysis of the Results

Table V shows a comparative accuracy comparison of the proposed work with existing methodologies. It can be clearly seen that the proposed method has outperformed the existing methodologies by a margin. Increased accuracy and precision of 97.2% and 97.4% is achieved when a multimodal approach with a weighted feature set is used. A remarkable false positive, discovery and a negative rate of 0.025, 0.025, and .003 is perceived in Table IV.

TABLE IV. Performance Parameters for the Multimodal Eye-blink Detection System

Recall	Specificity	Precision	Negative Predictive Value	False Positive Rate	False Discovery Rate	False Negative Rate	Accuracy	F1 Score	AUC score
0.97	0.975	0.974	0.970	0.025	0.025	0.03	0.972	0.972	0.972

TABLE V. Comparative Accuracy Comparison of the Proposed Work With Existing Methodologies

S. NO.	Author and Work Reference	Performance	Accuracy and precision of the proposed work
1	Michael Chau and Margrit Betke. 2005 [39]	Accuracy: 95.3	Accuracy: 97.2 Precision: 97.4 Recall: 97.0
2	Diego Torricelli et al. 2009 [40]	Accuracy: 95.7	
3	W.O. Lee et al 2010 [8]	Precision: 94.4	
4	Tomas Drutarovsky and Andrej Fogelton. 2014 [5]	Accuracy: 93.45	
5	Fengyi Song et al. 2014 [6]	Accuracy: 96.8	
6	Tereza Soukupova and Jan Cech. 2016 [7]	Recall: 92.9	
7	Federico M. Sukno et al. 2016 [20]	Frame classification acc.:97.1	
8	P. Singh and D. Virmani. 2018 [3]	Accuracy: 97	

VI. Conclusion

A multimodal eye blink recognition system using Z-score-based thresholding and weighted features was presented in this work. The eye blinks were detected using a blend of 5 weighted features (Vertical Head Positioning, Orientation Factor, Proportional Ratio, Area of Intersection, and Upper Eyelid Radius) depicting imperative gen (z score threshold), extracted from the circles uniquely formed from the eyelids landmarks. For testing the performance of the method, ZJU eye-blink dataset was used. While implementing the proposed method with the said dataset, it was observed that when there is an impulse in OF, it means that other features except UER will struggle to report a blink with a peak. In contrast, when OF has no impulse signals, it suggests the expected behavior, which peaks in all four features. The multimodal system's performance was increased to 97.2% (accuracy) with a precision of 97.4%. Other performance parameters also showed a decent routine. As a future scope, more features can be incorporated to increase the performance attributes further.

Appendix

Procedure to create a unique circle given 3 points:

The algorithm to create a circle given 3 points is as follows:

1. Let $(x_a, y_a), (x_b, y_b)$ & (x_d, y_d) be 3 points from which a circle is to be created. Now, consider the general equation of a circle as $(x - x_c)^2 + (y - y_c)^2 - r^2 = 0$ where (x_c, y_c) is the center of the circle, and r is the radius.
2. Given 3 points, plug the values into the equation of the circle.

$$(x_a - x_c)^2 + (y_a - y_c)^2 - r^2 = 0 \tag{A1}$$

$$(x_b - x_c)^2 + (y_b - y_c)^2 - r^2 = 0 \tag{A2}$$

$$(x_d - x_c)^2 + (y_d - y_c)^2 - r^2 = 0 \tag{A3}$$

3. Solve the linear equations formed for x_c, y_c by subtracting equation (A1) from (A2) and (A3).
4. Plug the values for the center (x_c, y_c) in any of the three quadratic equations and solve for r

An example:

$$\begin{aligned} x_a, y_a &= 1, 1 \\ x_b, y_b &= 2, 4 \\ x_d, y_d &= 5, 3 \end{aligned}$$

$$\begin{aligned} (1 - x_c)^2 + (1 - y_c)^2 - r^2 &= 0 \\ (2 - x_c)^2 + (4 - y_c)^2 - r^2 &= 0 \\ (5 - x_c)^2 + (3 - y_c)^2 - r^2 &= 0 \end{aligned}$$

$$\begin{aligned} -2x_c - 6(y_c - 3) &= 0 \\ (y_c + 7) - 6x_c &= 0 \end{aligned}$$

$$(x_c, y_c) = (3, 2)$$

$$(1 - 3)^2 + (1 - 2)^2 - r^2 = 0$$
$$5 - r^2 = 0$$
$$r = \sqrt{5}$$

References

[1] T. Soukupova, "Eye Blink Detection using Facial Landmarks," Diploma Thesis, Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University, Prague, 2016.

[2] Puneet Singh Lamba and Deepali Virmani, "Information Retrieval from Emotions and Eye Blinks with Help of Sensor Nodes," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 8, no.4, pp. 2433-2411, 2018.

[3] Puneet Singh Lamba and Deepali Virmani, "Reckoning number of eye blinks using eye facet correlation for exigency detection," *Journal of Intelligent & Fuzzy Systems*, vol. 35, no. 5, pp. 5279-5286, 2018.

[4] Stephen Milborrow and Fred Nicolls, "Locating Facial Features with an Extended Active Shape Model," *Computer Vision – ECCV 2008, Lecture Notes in Computer Science*, Springer, 5305 (2008), pp. 504-513.

[5] T. Drutarovsky and A. Fogelton, "Eye blink detection using variance of motion vectors," In: *Computer Vision - ECCV Workshops, 2014.*

[6] F. Song, X. Tan, X. Liu and S. Chen, "Eyes closeness detection from still images with multi-scale histograms of principal oriented gradients," *Pattern Recognition, The Journal of the Pattern Recognition Society*, vol. 47, pp. 2825-2838, 2014.

[7] T. Soukupova and J. Cech, "Real-Time Eye Blink Detection using Facial Landmarks," *21st Computer vision Winter Workshop, Luka Cehovin, RokMandeljc, VitomirStruc (eds.) RimskeToplice*, Slovenia, February 3-5, 2016.

[8] W.H. Lee, E.C. Lee and K.E. Park, "Blink detection robust to various facial poses," *Journal of Neuroscience Methods*, vol. 193, no. 2, pp. 356-372, 2010.

[9] N. Alsaeedi, and D. Wloka, "Real-Time Eyeblink Detector and Eye State Classifier for Virtual Reality (VR) Headsets (Head-Mounted Displays, HMDs)," *Sensors*, vol. 19, no. 5, 2019.

[10] R. Sanyal and K. Chakrabarty, "Two Stream Deep Convolutional Neural Network for Eye State Recognition and Blink Detection," *2019 3rd International Conference on Electronics, Materials Engineering & Nano-Technology (IEMENTech)*, Kolkata, India, 2019, pp. 1-8.

[11] C. Zhu and C. Huang, "Adaptive Gabor algorithm for face posture and its application in blink detection," *2019 IEEE 3rd Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, Chongqing, China, 2019, pp. 1871-1875.

[12] Muhammad Tayab Khan, Hafeez Anwar, Farman Ullah, et al., "Smart Real-Time Video Surveillance Platform for Drowsiness Detection Based on Eyelid Closure," *Wireless Communications and Mobile Computing*, 2019.

[13] A. Asthana, S. Zafeoriou, S. Cheng, and M. Pantic, "Incremental face alignment in the wild," in *Conference on Computer Vision and Pattern Recognition*, 2014.

[14] Gianluca Donato, Marian Stewart Bartlett, Joseph C. Hager, Paul Ekman, and Terrence J. Sejnowski, "Classifying Facial Actions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 10, pp.974-989, 1999.

[15] Ying-li, Takeo Kanade, and Jeffrey, "Recognizing Action Units for Facial Expression Analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 1-19, 2001.

[16] Choi, I., Han, S., & Kim, D. (2011, July), "Eye detection and eye blink detection using adaboost learning and grouping," In *2011 Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN)*, 2011, pp. 1-4. IEEE.

[17] Singh, H., & Singh, J., "Real-time eye blink and wink detection for object selection in HCI systems," *Journal on Multimodal User Interfaces*, vol. 12, no. 1, pp. 55-65, 2018.

[18] G. Pan, L. Sun, Z. Wu, and S. Lao, "Eyeblink-based anti-spoofing in face recognition from a generic webcamera,", In *ICCV*, 2007.

[19] L. Tan, K. Yu, F. Ming, X. Cheng & G. Srivastava, "Secure and Resilient Artificial Intelligence of Things: a HoneyNet Approach for Threat Detection and Situational Awareness," *IEEE Consumer Electronics Magazine*, 2021.

[20] F. M. Sukno, S.-K. Pavani, C. Butakoff and A.F. Frangi, "Automatic assessment of eye blinking patterns through statistical shape models," In *ICVS*, 2009.

[21] K. Arai, R. Mardiyanto, "Real Time Blinking Detection Based on Gabor Filter," *International Journal of Human Computer Interaction*, vol. 1, no. 3, pp. 33-45, 2010.

[22] Lamba, P.S., Virmani, D. & Castillo, O., "Multimodal human eye blink recognition method using feature level fusion for exigency detection," *Soft Comput* vol. 24, pp. 16829–16845, 2020, https://doi.org/10.1007/s00500-020-04979-5

[23] Usakli, A. B., Gurkan, S., Aloise, F., Vecchiato, G., & Babiloni, F., "On the use of electrooculogram for efficient human computer interfaces," *Computational intelligence and neuroscience*, 2010.

[24] Pander, T., Przybyla, T., & Czabanski, R. (2008, May), "An application of detection function for the eye blinking detection," In *2008 Conference on*

Human System Interactions, 2008, pp. 287-291. IEEE.

[25] Królak, A., & Strumiłło, P., "Eyeblink detection system for human–computer interaction," *Universal Access in the Information Society*, vol. 11, no. 4, pp. 409-419, 2012.

[26] Kraichan, C., & Pumrin, S. (2014, May), "Face and eye tracking for controlling computer functions," In *2014 11th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, 2014, pp. 1-6. IEEE.

[27] MacKenzie, I. S., & Ashtiani, B., "BlinkWrite: efficient text entry using eye blinks," *Universal Access in the Information Society*, vol. 10, no. 1, pp. 69-80, 2011.

[28] Arai, K., & Mardiyanto, R. (2011, April) "Eye-based HCI with full specification of mouse and keyboard using pupil knowledge in the gaze estimation," In *2011 Eighth International Conference on Information Technology: New Generations*, 2011, pp. 423-428. IEEE.

[29] Khilari, R. (2010, December)," Iris tracking and blink detection for human-computer interaction using a low resolution webcam," In *Proceedings of the Seventh Indian Conference on Computer Vision, Graphics and Image Processing, 2010,* pp. 456-463.

[30] Grauman, K., Betke, M., Lombardi, J., Gips, J., & Bradski, G. R., "Communication via eye blinks and eyebrow raises: Video-based human-computer interfaces," *Universal Access in the Information Society*, vol. 2, no. 4, pp. 359-373, 2003.

[31] Siriluck W, Kamolphiwong S, Kamolphiwong T, "Blink and click," In: *Proceedings of the 1st international convention on rehabilitation engineering and assistive technology: in conjunction with 1st Tan Tock Seng Hospital neurorehabilitation meeting*, 2007, pp. 43–46.

[32] Pimplaskar, D., Nagmode, M. S., & Borkar, A., "Real time eye blinking detection and tracking using OpenCV," *Technology*, vol. 13, no. 14, pp. 15, 2015.

[33] Missimer, E., & Betke, M. (2010, June), "Blink and wink detection for mouse pointer control," In *Proceedings of the 3rd International Conference on PErvasive Technologies Related to Assistive Environments*, 2010, pp. 1-8.

[34] Su, M., Yeh, C., Lin, S., Wang, P., & Hou, S. (2008, July), "An implementation of an eye-blink-based communication aid for people with severe disabilities," In *2008 International Conference on Audio, Language and Image Processing*, 2008, pp. 351-356. IEEE.

[35] Venkataramanan, S., Prabhat, P., Choudhury, S. R., Nemade, H. B., & Sahambi, J. S. (2005, January)," Biomedical instrumentation based on electrooculogram (EOG) signal processing and application to a hospital alarm system," In *Proceedings of 2005 International Conference on Intelligent Sensing and Information Processing*, 2005, pp. 535-540. IEEE.

[36] Kumar, D., & Poole, E. (2002, October), "Classification of EOG for human computer interface," In *Proceedings of the Second Joint 24th Annual Conference and the Annual Fall Meeting of the Biomedical Engineering Society][Engineering in Medicine and Biology,* Vol. 1, pp. 64-67. IEEE.

[37] Al-gawwam, S., & Benaissa, M., "Robust eye blink detection based on eye landmarks and Savitzky–Golay filtering," *Information*, vol. 9, no. 4, pp. 93, 2018.

[38] Han, Y. J., Kim, W., & Park, J. S., "Efficient eye-blinking detection on smartphones: A hybrid approach based on deep learning," *Mobile Information Systems*, 2018.

[39] M. Chau and M. Betke, "Real time eye tracking and blink detection with USB cameras," *Technical Report 2005-12, Boston University Computer Science*, May 2005.

[40] D. Torricelli, M. Goffredo, S. Conforto and M. Schmid, "An adaptive blink detector to initialize and update a view- based remote eye gaze tracking system in a natural scenario," *Pattern Recogn Lett*, vol. 30, no. 12, pp. 1144-1150, 2009.

[41] Trobo, I. P., Díaz, V. G., Espada, J. P., Crespo, R. G., & Moreno-Ger, P., "Rapid modeling of human-defined AI behavior patterns in games," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 7, pp. 2683-2692, 2019.

[42] Khari, M., Kumar, P., Burgos, D., & Crespo, R. G., "Optimized test suites for automated testing using different optimization techniques," Soft Computing, vol. 22, no. 24, pp. 8341-8352, 2018.

[43] García-Díaz, V., Espada, J. P., Crespo, R. G., G-Bustelo, B. C. P., & Lovelle, J. M. C., "An approach to improve the accuracy of probabilistic classifiers for decision support systems in sentiment analysis," *Applied Soft Computing*, vol. 67, pp. 822-833, 2018.

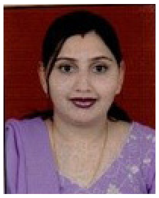
[44] Zhang, K., Zhang, Z., Li, Z., & Qiao, Y., "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499-1503, 2016.

[45] Harish, B. S., Maheshan, M. S., & Nagadarshan, N., "A Convolution Neural Network Engine for Sclera Recognition," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 6, no. 1, pp. 78-83, 2020. http://doi.org/10.9781/ijimai.2019.03.006

[46] Verma, K. K., Singh, B. M., Mandoria, H. L., & Chauhan, P., "Two-Stage Human Activity Recognition Using 2D-ConvNet," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 6, no. 2, 2020. http://doi.org/10.9781/ijimai.2020.04.002

Puneet Singh Lamba

Puneet Singh Lamba received the Master degree (M.tech) in Information Technology from University School of Information And Communication Technology, Guru Gobind Singh Indraprastha University in 2013. He is a PhD candidate in Information Technology at USICT, GGSIPU. His research interests are in the areas of Sensor Networks and Adhoc Networks. He has more than 11 years of teaching experience and is currently working as Assistant Professor in Bharati Vidyapeeth's College of Engineering New Delhi, India. He has published more than 10 research papers in International journals / National journals / International conferences of repute.



Deepali Virmani

Deepali Virmani has done, B. Tech, Computer Science from MDU, Rohtak in 2001, M. Tech in Information Technology in 2005, from GGSIPU and Ph.D Wireless Sensor Network from Delhi University in 2013. She has a teaching experience of 17 years. She is actively engaged in teaching and research in areas of Computer Science since 2001. She has published more than 73 research papers in International journals / National journals / International conferences of repute. Her research interests are in the areas of Sensor Networks, Data Mining and Security. She has guided more than 45 B.Tech projects. Presently she is guiding 5 Ph.D scholars registered with reputed universities like GGSIPU, UPTU. She is branch counselor BPIT -IEEE student chapter and BPIT-CSI student branch. She is on the reviewer panel / editorial board of many International Journals. She has organized many professional activities like FDPs, Workshops, expert lectures. She has been the session chair in National / International conferences.

Manu S Pillai

Manu S Pillai is currently working as an Instructor and Product Engineer for Machine Learning and Data Science in Coding Blocks, New Delhi, India. He is also associated with Information Technology Department, Bharati Vidyapeeth's College of Engineering, Guru Gobind Singh Indraprastha University, Delhi, India. His current research interests include Computer Vision, Image Processing & Deep Learning.

Dr. Gopal Chaudhary

Dr. Gopal Chaudhary is currently working as an assistant professor in Bharati Vidyapeeth's College of Engineering, Guru Gobind Singh Indraprastha University, Delhi, India. He holds a Ph.D. in Biometrics at the division of Instrumentation and Control engineering, Netaji Subhas Institute of Technology, University of Delhi, India. He received the B.E. degree in electronics and communication engineering in 2009 and the M.Tech. degree in Microwave and optical communication from Delhi Technological University (formerly known as Delhi College of Engineering), New Delhi, India, in 2012. He has 50 publications in refereed National/International Journals & Conferences (Elsevier, Springer, Inderscience) in the area of Biometrics and its applications. His current research interests include soft computing, intelligent systems, information fusion and pattern recognition. He has organized many conferences and leading special issues in Taylor and Francis, Springer, IOS Press, Computers Materials & Continua etc.

A Novel Technique to Detect and Track Multiple Objects in Dynamic Video Surveillance Systems

M. Adimoolam[1], Senthilkumar Mohan [2], John A.[3], Gautam Srivastava[4,5]*

[1] Department of Computer Science and Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai (India)
[2] School of Information Technology and Engineering, Vellore Institute of Technology, Vellore (India) 3 School of Computer Science and Engineering, Galgotias University, Greater Noida (India)
[4] Department of Mathematics and Computer Science, Brandon University, Brandon, MB R7A 6A9 (Canada)
[5] Research Center for Interneural Computing, China Medical University, Taichung 40402 (Taiwan)

Received 14 April 2020 | Accepted 8 October 2021 | Published 18 January 2022

unir
LA UNIVERSIDAD
EN INTERNET

Abstract

Video surveillance is one of the important state of the art systems to be utilized in order to monitor different areas of modern society surveillance like the general public surveillance system, city traffic monitoring system, and forest monitoring system. Hence, surveillance systems have become especially relevant in the digital era. The needs of the video surveillance systems and its video analytics have become inevitable due to an increase in crimes and unethical behavior. Thus enabling the tracking of individuals object in video surveillance is an essential part of modern society. With the advent of video surveillance, performance measures for such surveillance also need to be improved to keep up with the ever increasing crime rates. So far, many methodologies relating to video surveillance have been introduced ranging from single object detection with a single or multiple cameras to multiple object detection using single or multiple cameras. Despite this, performance benchmarks and metrics need further improvements. While mechanisms exist for single or multiple object detection and prediction on videos or images, none can meet the criteria of detection and tracking of multiple objects in static as well as dynamic environments. Thus, real-world multiple object detection and prediction systems need to be introduced that are both accurate as well as fast and can also be adopted in static and dynamic environments. This paper introduces the Densely Feature selection Convolutional neural Network – Hyper Parameter tuning (DFCN-HP) and it is a hybrid protocol with faster prediction time and high accuracy levels. The proposed system has successfully tracked multiple objects from multiple channels and is a combination of dense block, feature selection, background subtraction and Bayesian methods. The results of the experiment conducted demonstrated an accuracy of 98% and 1.11 prediction time and these results have also been compared with existing methods such as Kalman Filtering (KF) and Deep Neural Network (DNN).

Keywords

Convolutional Neural Network, Machine Learning, Object Detection, Video Surveillance.

DOI: 10.9781/ijimai.2022.01.002

I. Introduction

The human visual system detects and recognizes objects within dense groups of multiple objects very efficiently. But this task proves to be difficult and is riddled with challenges when it comes to artificial systems. Modern surveillance systems have been used in public civil monitoring by implementing object detection and motion tracking. Object detection is a subsidiary topic under the field of Computer Vision which is a study of how computers detect and classify different types of objects in an image or a video. There are numerous applications of such systems in the modern world that detect and track objects in a region such as surveillance systems for military use, modernized traffic control systems, public weather observation systems, etc. [1], [2], [3]. Researchers are working on various techniques to increase the speed and overall accuracy of such object recognition and tracking. Recent advancements in the field of information technology have increased the need for more robust and intelligent surveillance systems with better speed and accuracy. Therefore, object detection and tracking have shown great potential while emerging as an important technology in the field of surveillance related to security.

Unlike humans, computers see images as several clusters. Each pixel in an image contains data corresponding to the colour values i.e. red, green and blue. If an image contains all three colour values then there are three channels present in that image. A grayscale image contains only one channel [4]. Determining the location of an object and the region of interest is a challenging task in the field of computer vision. In general, two methods are used for determining the location of the object and they are object detection and object

* Corresponding author.
E-mail address: srivastavag@brandonu.ca

tracking. To locate an instance of the object in images or videos, the object detection technique is used. The popular Convolutional Neural Networks (CNN) [5] was trained using a large set of labelled data and it is used to detect a region of interest within a completely new given image. Hence using this method one can determine the location of an object within a given image, whereas in the case of object tracking only the pixel information of the region of interest is provided and the region having the highest amount of similarity is searched.

During the process of object detection, objects are detected based on various points such as objects of interest, face, colour, shape, and skin. The process involves the extraction of frames from an image or video. Subsequently, various such features of objects are extracted for video surveillance systems and this is discussed in detail [6]. The detected object is then continuously tracked in the input video stream. Numerous factors make it difficult to track objects after their detection. Several times, the object is occluded by its surroundings which makes it difficult for the tracking algorithm to track the object in real-time. Additionally, sudden movements which lead to changes in the shape or size of the object or changes in the observed scene are a few of the factors that can affect object-tracking. Researchers are continuously working on improving these algorithms for the better tracking of objects despite the above-mentioned hindrances that affect the procedure.

Real-time object tracking algorithms are being studied where the detector learns about all the changes in the object and its environment and uses it to better track the object. Some of the popular object detectors are Region-based CNN (R-CNN), Faster R-CNN, Single Shot Detectors and You Only Look Once (YOLO). Among these object-detectors, Faster R-CNN and Single Shot Detectors have greater accuracy, while YOLO has better speed. In this paper, we look at the benefits and drawbacks of two-stage detectors and single-stage detectors. Moreover, we explore how to improve the speed and accuracy of modern security surveillance systems,. Furthermore we fine-tune object detection with direct comparison to state-of-the-art detecting techniques.

The topics discussed in this research paper in the subsequent sections are as follows. Section II discusses current and related work on object detection methods, single-stage and multistage as well as multi-object target methods. Section III delves into the working methodologies and the multi-object detection approaches in DFCN-HP. Section IV contains the implementation and performance analysis of the proposed work along with the existing works followed by the conclusion.

II. Related Works

This section discusses the technology related to object detection (detection of the object in the image), classification (classifying objects into different categories such as dog, cat, person), and tracking of the objects. The main body of work done related to object detection, tracking and classifications are the standard methods for the general one-stage object detection and multistage object detection which are expanded further.

A. Object Detection Methods

Various algorithms have been developed to detect objects in images or videos. The goal of every object detection algorithm is to improve the overall accuracy by improving the confidence level of the object detector while minimizing the time taken to detect the object in the image or video. One stage detector and two-state detector are the two key object detection algorithms in use today and they are extensively used in surveillance object detection. Fig. 1 shows the processes involved in an object detection flow. The surveillance image or video has to be pre-processed to detect the object and subsequently, the detected object has to undergo a feature extraction process.

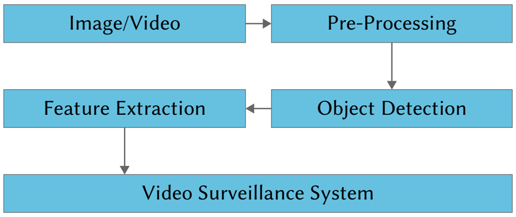


Fig. 1. Object Detection processes.

B. Single-Stage Detectors

The separate region proposal steps are not applied in single-state object detector algorithms. Instead, they consider every position on the image as a potential object and then try to classify each region of interest as an object or background. A few of the popular single-stage object detectors are discussed below.

YOLO: Joseph Redmon along with Ross Girshick and others proposed a new approach for object detection in which they framed object detection as a regression problem. This method divided the entire image into spatially separated bounding boxes with class probabilities associated with them. It could be an optimized end-to-end process since the whole framework was a single network. YOLO gained a remarkably faster speed than its predecessors by processing images in real-time at 45 frames per second [5].

Single-Shot Multibox Detector (SSD): W. Lue and others proposed a single-stage detector (2016) that used a single DNN for detecting objects in images. The output of space-bounding boxes for each grid cell was created to discretize after dividing the images into a grid cell. It has further been trained straightforwardly using SSD. SSD achieved 74.3% mAP(mean average precision) for the input size 300X300 using Visual Object Classes Challenge (VOC) 2007 at 59 frames per second [6].

YOLOv3: Joseph Redmon and Ali Farhadi (2018) proposed an improved version of YOLOv2, YOLOv3. They introduced a few design changes in YOLOv2 to make it better. The SSD runs 3 times faster for the input 320X320 using YOLOv3 and achieved 28.2 mAP just in 22ms [7].

C. Two-Stage Detectors

Two-stage detectors divide the detection of the object into two stages: in stage one, it identifies the subsets of the image that might contain an object (region proposal); and in-state two, it classifies the object for making predictions within the proposed region. The detector identifies the subset of the image which may potentially contain an object during the first state of two-stage detectors. This is done so that every object inside an image can belong to one of the proposed regions. The deep learning model has been applied further in these objects and labels are assigned based on object category and this is called the second stage of two-stage detectors. CNN before R-CNN mainly used a sliding window to generate regions individually with CNN classifiers to produce a set of probabilities. A general region-based CNN has the same approach but instead of selecting a huge number of regions to examine, this independently generates about 2000 regions of interest.

A few of the popular two-stage detectors are discussed below. Ross Girshick et al. proposed a novel two-stage detector using R-CNN (2014) [8]. When compared to the state-of-the-art traditional detectors (40.4% mAP), it obtained a 53.7% mAP performance and it significantly improved overall detection using R-CNN [9]. R-CNN involves three sequences in its pipeline: (i) proposal generation: to find regions in the image that might contain an object, and these regions are called region proposals, (ii) feature extraction: to extract all the CNN features from the region proposals, (iii) region classification: to classify all the objects using the extracted features.

Fast R-CNN [10]: This method is called a multi-task learning detector introduced by Ross Grishick et al. that overruns the R-CNN and also SPP-net [11]. It has R-CNN with ROI (Region of Interest) pooling layer to extract the feature of the region. The Fast R-CNN obtains an accuracy that is better than R-CNN and SPP-net and is of major significance. Fig. 2 shows the representation of Fast R-CNN objects.

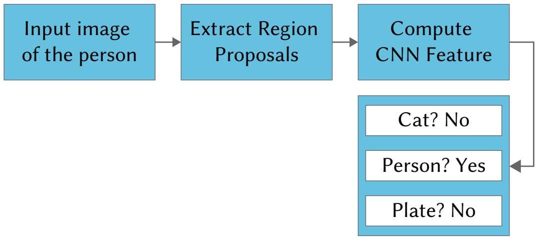


Fig. 2. R-CNN.

Faster R-CNN [12]: Subsequently, Ross Girshick and others proposed the state-of-the-art version of the R-CNN family in 2015. Here, the region proposals have been generated by Region Proposal Network (RPN) in Faster R-CNN. This method generates region proposals directly into the network instead of using an external algorithm. The frame rate of 5 fps for the Very Deep Convolutional Networks (VGG-16 model) [13] has been gained in Faster R-CNN. This performance is a remarkable achievement with an object detection accuracy of mAP 73.2% and 70.4% using Faster R-CNN's object proposal and this mAP has produced coloration with PASCAL 2007 and 2012 respectively. Fig.3 shows the Faster R-CNN working mechanism along with its object components.

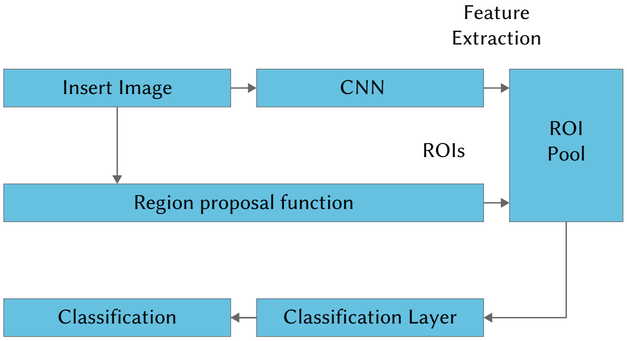


Fig. 3. Fast R-CNN.

Mask R-CNN [14]: The Faster R-CNN has generated a pixel-level mask of an object that has achieved state-of-the-art results. Further, the Mask R-CNN proposed by K. He, et al. has a branch for the prediction of an object which works parallel to the existing branch-box recognition and this parallel object prediction has proved to be more significant than the Faster R-CNN method and has outperformed it in all aspects concerning object detection and prediction.

D. Multi-Object Targets Methods

The geometric constraints method is used for target detection, recognition and tracking of objects using a distributed algorithm, [15]. This work is applied to different applications such as mobile cameras, multiple object detection in Multi-view, etc. The video surveillance techniques such as tracking and multiple object detection are discussed further. In this method, the Bayesian tracking multimodal framework is used without clearly associating object tracking and detection. It is observed to have an errorless performance, with missing detection problems also solved [16]. The real-time multiple objects tracked from the multiple camera surveillance systems were observed. In this work, object tracking and detection were performed using the feature selection parameter [17], with multi-object tracked from the multiple cameras put forward [18] from the surveillance of the video for which tracking and object detection was used [19]. This work uses the Pseudo motion algorithm, the Fourier shift theorem, and the two-stepped morphological operation is used to identify object properties such as region, size, etc. Here the Kalman Filtering (KF) is used for object tracking. The Local Maximal Occurrence Representation (LOMO) feature extraction algorithm is used to feature the representation of the objects and the Hankel matrix is used to manage the target objects, while the IHTLS algorithm is used to estimate the ranking of the objects. The real-time tracking of the objects from the multiple cameras was observed [20]. These works were applied to trace path tracking and trajectory finding using multiple cameras in different positions and multiple tracking of objects using the dual camera used to track it [21]. The geometry, homographic calibration was used for spatial mapping and a pan–tilt–zoom camera was used to detect the objects automatically. Multiple object detections continuously from the multiple cameras using single Target Track-Before-Detect, Particle Filter and predict and update methods are used to track the objects and were observed [22]. The video surveillance system computational cost for object detection was proposed by Rakesh Chandra Joshi et al. (2019) [23]. They use the Kalman Filter Assisted Occlusion Handling (KFAOH) technique for handling occasions. Table I shows the multi-targets tracking methods, with mention of the processing methods and algorithms, etc.

In the previous methods of object tracking, the detection was used for single or multiple objects from single and multiple cameras [35], [36], [37]. Here, it was proposed that multiple objects and multiple cameras be used to track and detect the objects automatically. Most of the works involved manual predictions with only the MGC algorithm having automatic tracking and detection. Hence, the work in question has multiple objects prediction from the multiple cameras atomically in different surveillance systems and from the sequence of videos. The work has also been compared with KF and DNN methods. Ahmad Jalal et al. (2017) [38] proposed a human activity recognition technique for a video surveillance system. The health care application of elder people monitoring was discussed in this work and used Hidden Markov Models (HMM) and robust multi-features model. This model recognized human activity in the experiment. Anahita Ghazvini et al. (2019) [39] discussed counting individuals in video surveillance as multiple object detection. The work used a Convolution Neural Network (CNN) to detect and count several objects in the surveillance video dataset.

III. The DFCN-HP

Multiple object detection from multiple cameras is called DFCN-HP. The DFCN-HP method consists of the following steps to detect multiple objects.

1. Pre-request information
2. Dense Block
3. Feature selection
4. Multiple Object detection and tracking
5. Hyperparameter Tuning
6. Data Acquisitions and Training

The first step is to pre-request information of multiple object detection from multiple cameras having detected multiple objects from multiple channels. Each channel has N objects and is selected. The channel and object combinations are shown in Fig. 4.

TABLE I. Multiple Object Target Tracking Methods

Algorithm	Features	Calibration	Multi-target	Limitation
KFAOH (Kalman Filter Assisted Occlusion Handling) (2019) [23]	Object detection	Manual	Yes	Only a single object detected with multiple cameras
GM (Graph matching) (2009) [24]	2D position, size, velocity	Manual	Yes	Only a single object detected with multiple cameras
CFI (Caratheodory-Fejer Interpolation) (2006) [25]	Pixels, manifold learning	Manual	Yes	Ambiguity occurs when detecting a single object with multiple cameras
GMPHD (Gaussian Mixture Probability Hypothesis Density filter) (2007) [26]	Position, size and colour histogram	Manual	Yes	Limited with object tracking infused video data
MGC (Minimum graph cut) (2009) [27]	Multiple planes occupancy map	Automatic	Yes	Plane view data object alone detected
VA (Viterbi algorithm) (2008) [28]	Colour and motion	Manual	Yes	Single object detection and probability is low
BT (Bayes tracker) (2008) [29]	Head position	Manual	Yes	Dense crowed single object detection and time is more to detect a single object
PF (Particle Filter) (2006) [30]	The vertical axis of the target, ground position	Manual	Yes	Guided particle filtering needed more dataset and time-consuming process
PF (Particle Filter) (2009) [31]	Signal intensity	Manual	Yes	More segmentation noise in object detection
NCA(neighbourhood components analysis) (2018) [32]	Posture change, Pedestrian tracking	Manual	Yes	Poor performance for low-quality videos object detection
KF(Kalman filtering) (2019) [33]	Multi-Object detection	Manual	Yes	Only Aerial Imagery data objects are detected
DNN (Deep Neural Network) (2019) [34]	Tracking multiple objects	Manual	Yes	A deep neural network takes more time to detect a blurred object

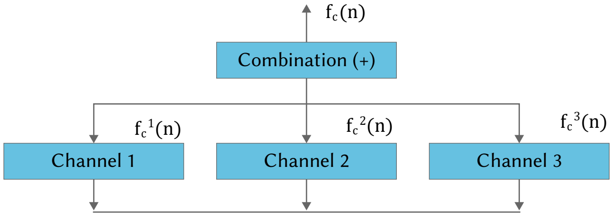


Fig. 4. Multiple channel combinations.

The *n* objects from each channel and the combination of objections are shown in Equation (1).

$$\mathrm{fc(n)} = \mathrm{fc}^1(\mathrm{n}) + \mathrm{fc}^2(\mathrm{n}) + \mathrm{fc}^3(\mathrm{n}) + \cdots + fc^n(\mathrm{n}) \tag{1}$$

Let *D* = {*D1*, ..., *Dk*} denote a set of *k* trained object boundary detectors of objects for a corresponding set of k situations *S* = {*S1*, ..., *Sk*}. Applying the j-th detector *Dj* to an image I give the boundary prediction *Dj* (*I*). The final object detection equation is shown in Equation (2).

$$\mathrm{D(I)}^{\mathrm{k}} = \sum_{\mathrm{j=1}}^{\mathrm{k}} \mathrm{P}\,(Sj(I))Dj(I) \tag{2}$$

Equation (2) denotes the sum of inter-product of probability between the set of various (*k*) situations of images and boundary detection of images.

The second step is a dense block and it is used to increase the prediction of the neural network. An important usage of the dense block is to increase the predictability of objects. Fig. 5 shows the dense block and Equation (3) represents the dense block target function in linear and nonlinear Equations.

$$\mathrm{f(x)} = \mathrm{f(\,w * y + b)} \tag{3}$$

The *f* (*x*) is the target activation function and is used in the entire linear and non-linear prediction of the objects (*y*) concerning weight (*w*), and block (*b*).

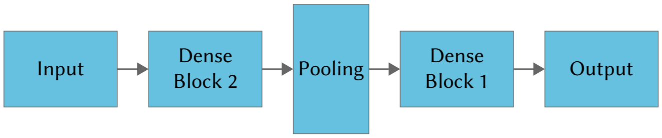


Fig. 5. Dense block.

The dense block connectivity between each layer receives the features of each input. The pooling layer operations are used to change the features from one layer to other layers between each block. Finally, each layer's features are concatenated for final operations. The concatenation operations of each layer's feature are shown in Equation (4).

$$\mathrm{C} = \mathrm{H}(c_1, c_2, c_3, \dots, c_{n-1}) \tag{4}$$

C - Concatenation of layers, *H* - Histogram Values, *M* - Multiple Inputs, (c_1, c_2, c_3, ..., c_{n-1}) are the concatenations of the features.

The third step is object selection and tracking is based on the features of the objects. The important features of the object selection and tracking depend on the following parameters such as (s) similarity of the objects (a) appearance of the objects (c) structure of the objects. The similarity of the objects is measured based on the following Equation (5).

$$\mathrm{Sim(\,I,J)} = \mathrm{Wa.\,a(i,j)} + \mathrm{Ws\,.\,s(i,j)} + \mathrm{Wl.\,l(i,j)} + \mathrm{Wsd.\,Sd(i,j)} \tag{5}$$

Where *Sim*(*I*, *J*) is the cost distance between the similarity of the objects, *W* is the weight of each attribute, *a* - appearance, *s* - structure, *l* - locations, *sd* - size difference. Equation (5) is used to manage each object in consecutive frames.

The appearance of an object is important to track and recognize the object continuously. The appearance depends on the viewpoint change, a correlation between objects (*C*), histogram (*H*) matching against the RGB, object orientation, transformation, keypoint features, etc. Equation (6) is used to match the appearance of the image for continuous detection and tracking in a dense block network from the sequence of frames. It reduces training time.

$$A(i,j) = \sum_{n}^{k} H * C(n,i)C(n.j) \tag{6}$$

Where *k* denotes the combined histogram data in memory from the past data, *n* denotes the number of frames detected from the tracking of frames. The structure of the features is another important clue to track the image and with it, the structure distance is also included. The structure distance is calculated based on the linear binary pattern [40] of the objects. The linear binary pattern captures the structure from the image. LPBH is used to recognize all the structures of the frames such as the face, nose, mouth, etc [41].

The fourth step is object detection and subtraction based on the shelf background subtraction method [42]. This shelf background method learns online information and undertakes foreground object subtraction and background information subtraction from the sequence of frames. The moving objects are detected from the frame regions pixel by pixel using Gaussians represented static sense. The fitting ellipse is used for foreground detection which combines the expectation from maximum methods used to estimate the number of the ellipse and the parameters [43] for the tracking of the objects traced using the Bayesian method. The general tracking of moving objects is represented in notation by Equation (7).

$$\mathrm{T} = [\mathrm{Tt}, \mathrm{n} | \mathrm{n} = 1, .., \mathrm{N}] \tag{7}$$

Where *T* is tracking, *N* is denoting the number of moving objects in each frame with time *t*. The n^{th} frame is denoted by Equation (8).

$$\mathrm{Tt}, \mathrm{n} \;\; = [\mathrm{P}, \mathrm{V}, \mathrm{E}] \tag{8}$$

Where *P* denotes object position, *V* denotes velocity, *E* denotes ellipse object position. Equation (9) represents the posterior probability density function to recursively measure the objects based on the current time slap.

$$\mathrm{P}(\mathrm{xt}|\mathrm{z1}{:}\,\mathrm{t}){=}\,\mathrm{P}(\mathrm{zt}|\mathrm{xt})\,\mathrm{P}(\mathrm{xt}|\mathrm{z1}{:}\,\mathrm{t}-1)/\mathrm{P}(\mathrm{zt}|\mathrm{z1}{:}\,\mathrm{t}-1) \tag{9}$$

Where $P\ (xt\,|\,z1{:}\ t{-}1)$ denotes prior probability, $P\ (zt\,|\,xt)$ denotes likelihood, and $P\ (zt\,|\,z1{:}\ t{-}1)$ denotes normalization factors.

The fifth step is the optimization step using hyperparameter tuning to select a learning process. Before the learning process begins, the values of the hyperparameters are set. Tuning hyperparameters is often a difficult task and is used to train the dense block and their various parameters. Below there are a few of the hyperparameters that are considered. The optimization steps of hyperparameter turning are as following steps.

Step 1: An activation function introduces the non-linear functionality to our network. The activation function helps the neural network to understand something complicated and complex. The main purpose of the activation function is to change the input signal of the sequence of the frame into the output signal.

Step 2: The learning rate controls the rate of learning for each batch of iteration.

Step 3: The Number of Epochs is the number of times training sets are passed into the dense neural network.

Step 4: The batch size parameter denotes the size of the batches that are used during the training process. Mini batch size or frames is preferable in the training process.

Step 5: Step 5 is the backbone for the pre-processing of the dense block of the network.

Step 6: To train the maximum number of regions of interest.

Step 7: This step is the validation of every epoch in the training steps. If the value of the validation steps increases, then the accuracy of validation states will improve, but it will slow down the training.

Step 8: The confidence threshold step is determining how confident it can allow the correct detections to be. It will filter out the non-confident findings by the dense block. This threshold can increase its value to generate more proposals.

The sixth step is data acquisition. Training is used to train the multiple objects and the training data is an acquisition from the huge amount of data. The entire neural network is trained using a stochastic gradient descent using a dense batch size of 64 for 400 and 50 epochs, correspondingly. The starting learning rate is 0.1 divided by 20 at 60% and 85% of training epochs. The dense block network train models 90 epochs using 256 batch sizes. The learning rate start is set to 0.1 and lowered by 10. The graphic processor memory constrains the trained data to a mini-size batch of 156. To compensate for the small batches of frames, we increase the model for 100 epochs and divide the rate by 20 at 90 epoch. Based on the training, the objects are detected from a huge number of datasets.

Finally, the object tracking and detection of the entire process are shown in Algorithm 1.

```
Algorithm 1: Multiple object detection and tracking - DFCN-HP
Input: Sequence of frames from multiple cameras
Output: predicted objects and tracking
Initial: Capture the frames from the multiple cameras
Begin
If the objects are selected from the multiple frames
Combine the objects for tracking (Equation -1)
For each frame of multiple sources
        Use the features
        Predict the objects
        Add objects from multiple frames
End For
End If
For each
        The similarities of the objects are tracked (Equation -6)
        The appearance of the objects are tracked using various
        parameters (Equation -7)
        Objects are tracked using the Bayesian method (Equation 8)
End For
If Objects recursively traced on current time slap (Equation 10)
Return Optimal value
Performing hyperparameter tuning
Else
Continuously trained and Acquisition
End
```

IV. Implementation

The real-time datasets and CIFAR datasets [44] are used for implementations. The real-time datasets are captured from the multiple cameras for the implementation shown in Fig. 6.

Table II gives the configuration details used in image training and, based on the training, the DFCN-HP provides the results. The testing of the data is also associated with real-time and CIFAR datasets. Before testing the real-time and CIFAR datasets, its samples are fine-tuned. The fine-tuning does not increase the object tracking performance. The validation tests of a set of frames are used to verify the validity of the frames of objects. Accordingly, the static and dynamic objects are trained continuously using a dense block model.

Fig. 6. Set of images used for results.

TABLE II. Configurations Details for Training of DFCN-HP

Backbone of network	**Dense block**
Backbone Strides	16, 32, 64
Batch Size	64
The detection of max Instances	100
The detection of Min Confidence	0.9
Detection dense block Threshold	0.3
Frames per Graphics Processor Unit	1
Image Shape	[1024, 1024, 3]
Learning Momentum	0.9
Learning Rate	0.1
Min size of the image	156
Threshold	0.5
Max size of the image	256
Steps Per Epoch	1000
Train ROIs Per Image	200
Validation Steps	50

Most of the previous works used the small size of samples and it is difficult for detection and tracking. But using DFCN-HP, medium-size objects with fine-tuned objects are detected and tracked. Before tuning and after tuning results are shown in Table III. The proposed work of DFCN-HP consists of 50 sequences of frames which are grouped into 25 sequences of frames for training and 25 sequences of frames for testing. The targeted public real-time sequence of frames datasets from the different cameras are different in the following parameters such as viewpoint, camera motion, object density, target motion, object motion direction and objects movement direction, etc.

TABLE III. Parameters Details Before and After Tuning

Parameters	**Before tuning**	**After tuning**
Train Anchors Per Image	256	32
Detection Min Confidence	0.9	0.8
Learning Rate	0.1	0.01
Weight Decay	0.0001	0.1
RPN NMS Threshold	0.5	0.7

The implementation of DFCN-HP is similar to Mask R-CNN with ResNet-101 [19] and YOLOv3 for object tracking and detections from the sequence of frames. The fine-tuning of the process for the entire proposed work is according to the DFCN-HP needs and flow of the work. Here, it was run as a sequence of images similar to Mask R-CNN with ResNet-101 and YOLOv3, although DFCN-HP speed was taken into consideration. Based on the proposed work the object detection and tracking are shown in Fig. 7. For security surveillance systems, accuracy and details are far more important than is the case the field of security.

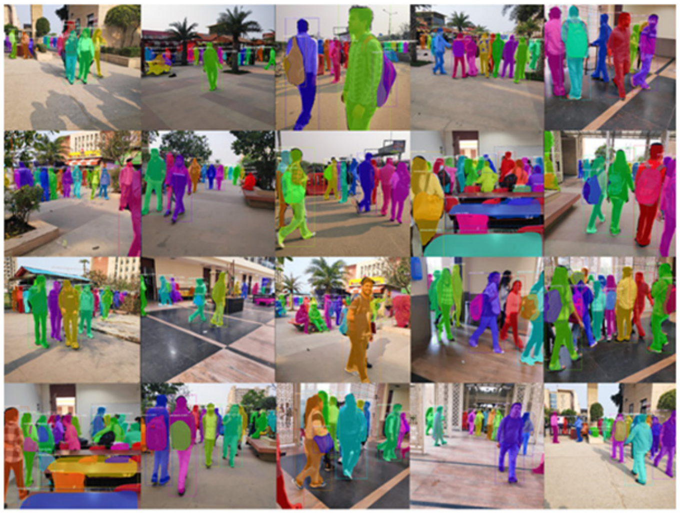


Fig. 7. Objects detection and tracking.

A. Improvement Via Tuning

While working with DFCN-HP, it was observed that reducing the batch size decreased the overall training time. Consequently, reducing the learning rate increased the confidence of predictions. With a minimum confidence threshold set to 0.7, our dense neural network identified a laptop with confidence above 95% but it also identified the display of the laptop as a "tv" with a confidence of 95%. After increasing the confidence threshold to 0.9, our DFCN-HP ignored the regions of the image of "tv" and only predicted the laptop. Using this example, similar object predictions also increased in the DFCN-HP.

In each trial of tuning more hyperparameters, the results were checked and the new results were compared to the old ones. All the tests were performed on a macOS with 8GB of RAM. The dense block per Image was reduced from 256 to 32. To increase the number of proposals, the value of Detection was decreased to a Min Confidence from 0.9 to 0.8 which enabled the detector to predict regions with lower confidence. The increased learning rate from 0.001 to 0.01 was to speed up the learning process. This helped a lot in our test runs. In weight decay, the weights were multiplied by a number slightly less than 1 to prevent the weights from growing too large. This changed weight decay from 0.0001 to 0.1 and worked well. Finally, the value of the dense block threshold was increased from 0.5 to 0.7 to generate more proposals. Many tests were performed on different hyperparameters to pick a certain value that was best for our detector.

Changing the backbone from *one dense block to another dense block* improves the accuracy and speed to a great extent. It was decided to stick to a *dense block* after performing some tests[47]. Here, two tests were performed on each image, one that did not change the hyperparameters and another that tuned the hyperparameters. It was observed that in most of the cases our detector performed well with high confidence and more proposals. The image in Fig. 7 and the chart in Table III and IV clearly show improvements in the detector.

Fig. 7 represents a real-time surveillance system using multiple sequences of frames. Each frame indicates that every object is validated and compared to the trained dataset. If any new objects have been detected in the frames, the prediction and tracking take place.

The detector performs well with Multiple Object Tracking Accuracy (MOTA) for 98% of people, backpacks and handbags. In this, the single iterations for single object predictions rates are 98.187% for people, 97.719% for backpacks and 96.138% for handbags in public places. The Mostly traceable Object (MTO) rate is 99.2%. Overall, various comparison parameters [48] are IDP (ID precision), IDR (ID recall), IDF1 (ID F-score), MOTA-Multiple Object Tracking Accuracy, MOTP - Multiple Object Tracking Precision, Rcll -Recall, Prcn - Precision, MTO-Mostly tractable Object, ML -Mostly Lost Object. The predicted and traced results of DFCN-HP using real-time data are shown in Table IV.

TABLE IV. Results Ff DFCN-HP Prediction Using Real-Time Data

Method	IDF1	IDP	IDR	MOTA	MOTP	RcLL	Prcn	MTO	ML
RNN	-	-	-	82.9	80.3	92.3	95.3	85	15.2
KF	90	90	90	96.4	90.6	98.2	98.2	95.5	3.6
DNN	90.5	90.3	90.6	97.5	88.5	99	98.7	98.9	0
Proposed method	92.5	93.8	94	98	90.8	99.5	99.2	99.2	0

B. Performance and Comparison

The performance of our proposed method DFCN-HP is compared to previous works such as those based on KF [33], DNN [34] and Recurrent Neural Networks [45],[46]. The proposed work outputs are associated with all the detection and tracking scores. The proposed work obtains a recall (RcLL) and precision (Prcn) of 99.5% and 99.2%, respectively, and the prediction results are high when compared with those of KF and DNN methods. The tuning prediction is observed to increase in every frame in the image. The prediction performance is very high. The detection of false-positive alignment is 98.5% and negative detection tracking in the sample is 1.5%. Multiple object detection and distributions are shown in Fig. 8. The results show the prediction distributions of persons, backside bags and handbags, which are 99%, 98.4%, 98% respectively. As shown in Fig. 8, the person's prediction is higher than other objects in various iterations.

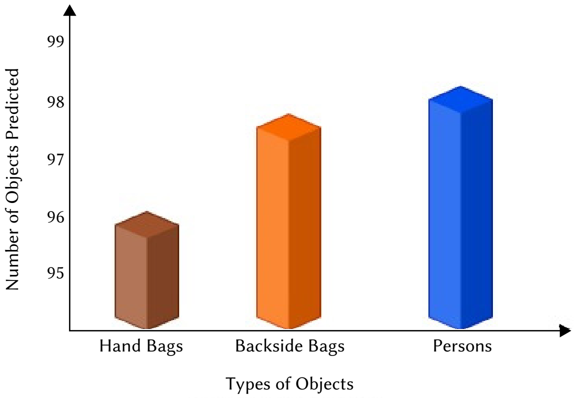


Fig. 8. The number of the objects predicted.

The precision value is associated with the true accuracy of predictions. The recall value is associated with true prediction and tracking found in the sequence of frames. The proposed work prediction is used in tuning and the tuning parameters are shown in Table III and the performance is shown in Fig. 9. The fine-tuning improvement showed a better performance compared to the other works. The time taken to predict an object is a measure in seconds. From Fig. 9, it is clear that the performance of the proposed DFCN-HP is better to predict objects than DNN and KF.

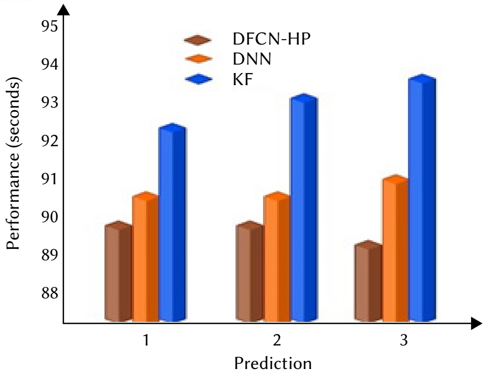


Fig. 9. Performance of Fine-Tuning.

The overall comparison parameters of the proposed work MOTA, MOTP, precision, Recall and MT are compared with those of existing works [33], [34], [45] and are shown in Fig. 10. The proposed work of MT is 99% and compared to the other methods it produces a high tracing rate. All the methods are including spatial information for tracking such as detection bounding areas, appearance, etc. and all the existing methods are not using temporal information. This proposed work considered the delay time and time slap also for prediction and tracing. The proposed work DFCN-HP and existing work DNN [34] are having online trackers with a similar learned motion model. The comparison of the results and multiple parameters is noted and it is shown in Table V and overall performance is shown in Fig. 10.

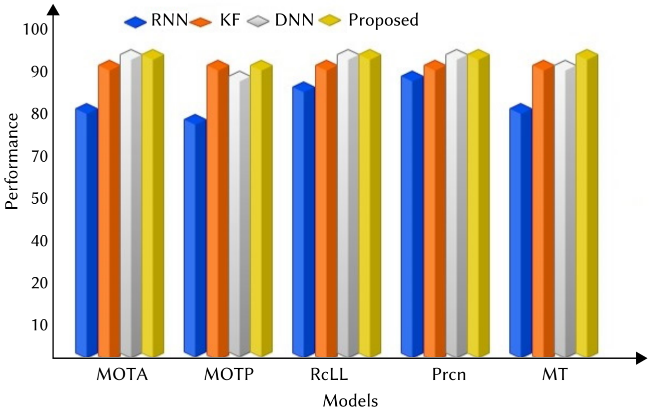


Fig. 10. Performance Comparison of the proposed work.

TABLE V. Comparison to Detection to Tracking

Tracking parameters with Methods											
Precision	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
All assignment (DNN)	0.3	0.422	0.92	0.92	0.93	0.92	0.93	0.93	0.88	0.7	0.5
Detection to track (DNN)	0.9	0.92	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.91
DFCN-HP - all assignment	0.4	0.8	0.93	0.95	0.95	0.95	0.94	0.94	0.93	0.8	0.7
DFCN-HP - Detection to track	0.98	0.98	0.99	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98

The speed of prediction of the proposed work is shown in Fig. 11. The speed of the prediction of work is compared to different standard object detection and tracking methods such as Fast R-CNN, R-CNN, and Faster R-CNN. The speed of prediction of the work DFCN-HP is 0.11 and, compared to the other existing work, this value is considered very low.

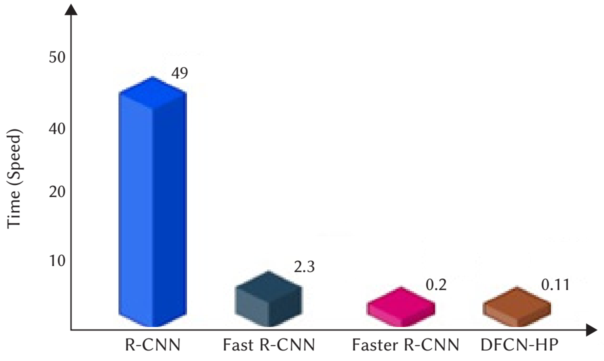


Fig. 11. Performance Comparison

V. Conclusion

Recently, there has been considerable advancement in the field of security and surveillance through different research projects that are being carried out by researchers. The proper utilization of all the new advanced techniques in object detection could dramatically change the field of object detection and open the doors to new research areas. In this research work, keeping surveillance systems for security in mind, the goal was to take a look at different types of static and dynamic object detection and tracking hybrid methods as have been introduced in this work. The main goal of the proposed hybrid DFCN-HP work is to increase the accuracy and decrease the training time to contribute to the area of human security systems. Furthermore, in this work, the hyperparameters have been fine-tuned to increase the speed and accuracy of the model. Several tests were performed to tune the hyperparameters and to evaluate the difference in performance thereof and consequently to pick certain new values of these hyperparameters for video surveillance systems. The proposed hybrid DFCN-HP method was also compared to the KF and DNN methods and was observed to produce better results in terms of multiple parameters such as MTO, ML and Accuracy.

References

[1] Ahn, H., and Cho, H., "Research of multi-object detection and tracking using machine learning based on knowledge for video surveillance system," *Personal and Ubiquitious Computing*, pp. 1-10, 2019, doi:10.1007/s00779-019-01296-z.

[2] A. Raghunandan, Mohana, P. Raghav and H. V. R. Aradhya, "Object Detection Algorithms for Video Surveillance Applications," *International Conference on Communication and Signal Processing* (ICCSP), 2018, pp. 0563-0568, doi:10.1109/ICCSP.2018.8524461.

[3] G. Chandan, A. Jain, H. Jain and Mohana, "Real Time Object Detection and Tracking Using Deep Learning and OpenCV," *International Conference on Inventive Research in Computing Applications (ICIRCA)*, 2018, pp. 1305-1308, doi: 10.1109/ICIRCA.2018.8597266.

[4] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137-1149, 2017, doi:39. 10.1109/TPAMI.2016.2577031.

[5] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779-788, doi: 10.1109/CVPR.2016.91.

[6] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A. C. Berg, "SSD: Single shot multibox detector," *Computer Vision in ECCV*, 2016, pp. 1-17, doi.org/10.1007/978-3-319-46448-0_2.

[7] Redmon, Joseph and Ali Farhadi. "YOLOv3: An Incremental Improvement" *ArXiv* abs/1804.02767, 2018, pp. 1-6.

[8] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," *IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, 2014, pp. 580-587, doi: 10.1109/CVPR.2014.81.

[9] Uijlings, Jasper Sande, K. and Gevers, T. and Smeulders, Arnold, "Selective Search for Object Recognition," *International Journal of Computer Vision,* vol. 104, pp. 154-171, doi:10.1007/s11263-013-0620-5.

[10] R. Girshick, "Fast R-CNN," *IEEE International Conference on Computer Vision (ICCV)*, Santiago, 2015, pp. 1440-1448, doi: 10.1109/ICCV.2015.169.

[11] Kaiming HeXiangyu and ZhangShaoqing RenJian Sun, "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 37, pp. 1904-1916, 2014.

[12] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 39, no. 6, 1137–1149, 2017, DOI:https://doi.org/10.1109/TPAMI.2016.2577031.

[13] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *In International Conference on Learning Representations*, 2015, pp.1-14.

[14] He, Kaiming. "Mask R-CNN," *2017 IEEE International Conference on Computer Vision (ICCV),* 2017, pp. 2980-2988.

[15] Sankaranarayanan, Aswin Veeraraghavan, Ashok Chellappa, Rama, "Object Detection, Tracking and Recognition for Multiple Smart Cameras," *Proceedings of the IEEE*, vol. 96, no. 10, pp. 1606 – 1624, doi:10.1109/JPROC.2008.928758.

[16] C. R. del-Blanco, F. Jaureguizar and N. Garcia, "An efficient multiple object detection and tracking framework for automatic counting and video surveillance applications," *EEE Transactions on Consumer Electronics*, vol.58, no.3, pp.857-862, August 2012, doi: 10.1109/TCE.2012.6311328.

[17] K. S. Kumar, S. Prasad, P. K. Saroj and R. C. Tripathi, "Multiple Cameras Using Real Time Object Tracking for Surveillance and Security System," *3rd International Conference on Emerging Trends in Engineering and Technology*, 2010, pp. 213-218, doi: 10.1109/ICETET.2010.30.

[18] Ray, Kumar S. and Soma Chakraborty, "An Efficient Approach for Object Detection and Tracking of Objects in a Video with Variable Background," *ArXiv* abs/1706.02672, 2017, pp. 1-11.

[19] Wenqian Liu, Octavia Camps, and Mario Sznaier, "Multi-camera Multi-Object Tracking," *ArXiv* abs/1709.07065, 2017, pp.1-7.

[20] Kachhava, Rajendra, Shrivasta, Vivek, Jain, Rajkumar, Chaturvedi, Ekta, "Security System and Surveillance Using Real-Time Object Tracking and Multiple Cameras," *Advanced Materials Research*, vol. 403-408,4968-4973, doi: 10.4028/www.scientific.net/AMR.403-408.4968.

[21] Chen, Chung-Hao, Yao, Yi, Page, David, Abidi, Besma, Koschan, Andreas, Abidi, Mongi, "Heterogeneous Fusion of Omnidirectional and PTZ Cameras for Multiple Object Tracking. Circuits and Systems for Video Technology," *IEEE Transactions on Circuits and Systems for Video Technology,* vol.18, no.8, pp.1052-1063, doi: 10.1109/TCSVT.2008.928223.

[22] Taj, Murtaza Cavallaro, Andrea, "Simultaneous Detection and Tracking with Multiple Cameras," *Studies in Computational Intelligence*, 411, pp 197-214, 2013, doi:10.1007/978-3-642-28661-2_8.

[23] R, Y. Da Xu and M. Kemp, "Fitting multiple connected ellipses to an image silhouette hierarchically," *IEEE Transactions. on Image Processing.* vol.19, no. 7, 1673-1682, Jul 2010.

[24] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu, "Object Detection With Deep Learning: A Review," *IEEE Transactions on Neural Networks and Learning Systems, vol.* 30, pp. 3212-3232, 2019.

[25] Licheng Jiao, Fan Zhang, Fang Liu, Shuyuan Yang, Lingling Li, Zhixi Feng, and Rong Qu, "A Survey of Deep Learning-Based Object Detection," *IEEE Access* vol. 7, pp. 128837-128868, 2019.

[26] Anjum, Nadeem Cavallaro, Andrea . "Trajectory Association and Fusion across Partially Overlapping Cameras," *sixth ieee international conference on advanced video and signal based surveillanc.* pp 201-206,2009. 10.1109/AVSS.2009.65.

[27] V. Morariu and O. Camps, "Modeling Correspondences for Multi-Camera

Tracking Using Nonlinear Manifold Learning and Target Dynamics," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, New York, NY, USA, 2006, pp. 545-552. doi: 10.1109/CVPR.2006.189.

[28] Fleuret, F., Berclaz, J., Lengagne, R., Fua, P. "Multicamera people tracking with a probabilistic occupancy map," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30(2),267–282 (2008)

[29] Eshel, R., Moses, Y. "Homography-based multiple camera detection and tracking of people in a dense crowd," In: *Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition,* Anchorage, AK, USA (June 2008).

[30] Kim, K., Davis, L.S.: "Multi-Camera Tracking and Segmentation of Occluded People on Ground Plane Using Search-Guided Particle Filtering," *European Conference on Computer Vision.* Springer, Berlin, Heidelberg, 2006.

[31] Taj, Murtaza Cavallaro, Andrea. "Multi-camera track-before-detect," *3rd ACM/IEEE International Conference on Distributed Smart Cameras*, ICDSC 2009. 1 - 6. 10.1109/ICDSC.2009.5289405.

[32] Tan, Yihua Tai, Yuan Xiong, Shengzhou. . "NCA-Net for Tracking Multiple Objects across Multiple Cameras," Sensors. 18. 3400. 10.3390/s18103400,2018.

[33] Hossain, Sabir, and Deok-Jin Lee. "Deep Learning-Based Real-Time Multiple-Object Detection and Tracking from Aerial Imagery via a Flying Robot with GPU-Based Embedded Devices," *Sensors (Basel, Switzerland)* vol. 19, 15 3371. 31 Jul. 2019, doi:10.3390/s19153371

[34] Yoon, Kwangjin. "Data Association for Multi-Object Tracking via Deep Neural Networks," *Sensors (Basel, Switzerland)* vol. 19, no. 3 pp. 559. 29 Jan. 2019, doi:10.3390/s19030559

[35] Sikora P, Malina L, Kiac M, Martinasek Z, Riha K, Prinosil J, Jirik L, Srivastava G. "Artificial Intelligence-based Surveillance System for Railway Crossing Traffic," *IEEE Sensors Journal.* 2020 Oct 16.

[36] Vallathan G, John A, Thirumalai C, Mohan S, Srivastava G, Lin JC. "Suspicious activity detection using deep learning in secure assisted living IoT environments," *The Journal of Supercomputing.* 2020 Jul 30:1-9.

[37] Wang X, Srivastava G. "The security of vulnerable senior citizens through dynamically sensed signal acquisition," *Transactions on Emerging Telecommunications Technologies.* 2020 Jul 14:e4037.T.

[38] Ahmad Jalal , Shaharyar Kamal and Daijin Kim, "A Depth Video-based Human Detection and Activity Recognition using Multi-features and Embedded Hidden Markov Models for Health Care Monitoring Systems," *International Journal of Interactive Multimedia and Artificial Intelligence*, Vol. 4, No. 4, pp. 54-62, 2017.

[39] Anahita Ghazvini, Siti Norul Huda Sheikh Abdullah, Masri Ayob, "A Recent Trend in Individual Counting Approach Using Deep Network," *International Journal of Interactive Multimedia and Artificial Intelligence*, Vol. 5, No. 5, pp. 7-14, 2019.

[40] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern Recognition,* vol. 29, no. 1, pp. 51–59, 1996.

[41] Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 12, pp. 2037–2041, 2006.

[42] Z. Zivkovic and F. van der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction," *Pattern Recognition Letters*, vol. 27, pp. 773–780, 2006.

[43] Rakesh Chandra Joshi, Adithya Gaurav Singh, Mayank Joshi, Sanjay Mathur, "A Low Cost and Computationally Efficient Approach for Occlusion Handling in Video Surveillance Systems," *International Journal of Interactive Multimedia and Artificial Intelligence*, Vol. 5, No. 7, pp. 28-38 2019.

[44] Krizhevsky, Alex. "Learning Multiple Layers of Features from Tiny Images," *Technical Report TR-2009, University of Toronto*, Toronto..

[45] Sadeghian, A.; Alahi, A.; Savarese, S. Tracking, "The Untrackable: Learning To Track Multiple Cues with Long-Term Dependencies," *arXiv 2017, arXiv*:1701.01909.

[46] Taj, Murtaza. "Tracking interacting targets in multi-modal," *sensors.* Diss. 2009.

[47] Chen, Muchun, et al. "Real-Time Multiple Pedestrians Tracking in Multi-camera System," *International Conference on Multimedia Modeling.* Springer, Cham, 2020.

[48] Xiaokai, Liu, Wang Hongyu, and Gao Hongbo. "Camera matching based on spatiotemporal activity and conditional random field model," IET Computer Vision 8.6 (2014): 487-497.

M. Adimoolam

Dr. M. Adimoolam received his Under Graduate Degree B.Tech in Computer Science and Engineering Discipline at Pondicherry University. He finished his Post Graduate Degree M.Tech in Information Security at Pondicherry Engineering College, Puducherry and it was sponsored by the Department of Information Technology, India under Project Information Security Awareness and Education. In 2019 he was completed a Ph.D. in Information Security – Computer Science and Engineering discipline at Manononmaniam Sundaranar University, Tirunelveli. Right now he is working as an Associate Professor at Saveetha School of Engineering in Institute of Computer Science. He is the life time member of ISTE, India. He has cleared the University Grant Commission conducting National Eligibility Test 8 times. He has also cleared the Tamilnadu Government conducting State Eligibility Test continuously 3 times. His research areas of interest are computer network, Information and Network Security, machine learning and deep learning.

John A

Dr. John A received his Under Graduate Degree B.Tech in Computer Science and Engineering Discipline at Pondicherry University. He finished his Post Graduate Degree M.Tech in Computer Science and Engineering at Pondicherry University, India. In 2019 he was completed a Ph.D. in Computer Science and Engineering discipline at Manonmaniam sundaranar University, India. Right now, he is working as an Assistant Professor at School of Computer Science in Galgotias University, India. He is the life time member of ISTE, India. His research areas of interest are real time applications, Data analysis and prediction, and Spatial and Temporal Database.

Senthilkumar Mohan

Dr. Senthilkumar Mohan was felicitated with a Ph.D. in engineering and technology from Vellore Institute of Technology in the year 2017. He has obtained his M.Tech in IT from VIT University in the year 2013. He earned his M.S (Software Engineering) degree in computer science and Engineering from VIT University Vellore, in the year2007. He is presently working in the rank of Associate Professor at the Department of Software and System Engineering, Vellore Institute of Technology, School of Information Technology and Engineering, Vellore, India. His area of research includes Artificial Neural Networks, Deep Learning, cloud computing. He has contributed to many research articles in various journals and conferences of repute. He is also a member of various professional societies like CSI, Indian congress, etc.

Gautam Srivastava

Dr. Gautam Srivastava was awarded his B.Sc. degree from Briar Cliff University in U.S.A. in the year 2004, followed by his M.Sc. and Ph.D. degrees from the University of Victoria in Victoria, British Columbia, Canada in the years 2006 and 2011, respectively. He then taught for 3 years at the University of Victoria in the Department of Computer Science, where he was regarded as one of the top undergraduate professors in the Computer Science Course Instruction at the University. From there in the year 2014, he joined a tenure-track position at Brandon University in Brandon, Manitoba, Canada, where he currently is active in various professional and scholarly activities. He was promoted to the rank Associate Professor in January 2018. Dr. G, as he is popularly known, is active in research in the field of Data Mining and Big Data. In his 8-year academic career, he has published a total of 243 papers in high-impact conferences in many countries and in high-status journals (SCI, SCIE) and has also delivered invited guest lectures on Big Data, Cloud Computing, Internet of Things, and Cryptography at many Taiwanese and Czech universities. He is an Editor of several international scientific research journals. He currently has active research projects with other academics in Taiwan, Singapore, Canada, Czech Republic, Poland and U.S.A. He is constantly looking for collaboration opportunities with foreign professors and students. Assoc. Prof. Gautam Srivastava received *Best Oral Presenter Award* in FSDM 2017 which was held at the National Dong Hwa University (NDHU) in Shoufeng (Hualien County) in Taiwan (Republic of China) on November 24-27, 2017.

LIPSNN: A Light Intrusion-Proving Siamese Neural Network Model for Facial Verification

Asier Alcaide[1], Miguel A. Patricio[2], Antonio Berlanga[2], Angel Arroyo[3], Juan J. Cuadrado-Gallego[4,5] *

[1] Ultra Tendency International GmbH, 39326 Colbitz (Germany)
[2] Applied Artificial Intelligence Group, University Carlos III de Madrid, Colmenarejo, Madrid (Spain)
[3] Department of Information Systems. Technical University of Madrid, Madrid (Spain)
[4] Department of Computer Science, University of Alcalá, Madrid (Spain)
[5] Department of Computer Science and Software Engineering, Concordia University, Montreal (Canada)

Received 25 March 2021 | Accepted 20 September 2021 | Published 9 November 2021

LA UNIVERSIDAD
EN INTERNET

Abstract

Facial verification has experienced a breakthrough in recent years, not only due to the improvement in accuracy of the verification systems but also because of their increased use. One of the main reasons for this has been the appearance and use of new models of Deep Learning to address this problem. This extension in the use of facial verification has had a high impact due to the importance of its applications, especially on security, but the extension of its use could be significantly higher if the problem of the required complex calculations needed by the Deep Learning models, that usually need to be executed on machines with specialised hardware, were solved. That would allow the use of facial verification to be extended, making it possible to run this software on computers with low computing resources, such as Smartphones or tablets. To solve this problem, this paper presents the proposal of a new neural model, called *Light Intrusion-Proving Siamese Neural Network*, LIPSNN. This new light model, which is based on Siamese Neural Networks, is fully presented from the description of its two block architecture, going through its development, including its training with the well- known dataset Labeled Faces in the Wild, LFW; to its benchmarking with other traditional and deep learning models for facial verification in order to compare its performance for its use in low computing resources systems for facial recognition. For this comparison the attribute parameters, storage, accuracy and precision have been used, and from the results obtained it can be concluded that the LIPSNN can be an alternative to the existing models to solve the facet problem of running facial verification in low computing resource devices.

Keywords

Deep Learning, Facial Verification, Neural Networks, Siamese Neural Networks.

DOI: 10.9781/ijimai.2021.11.003

I. Introduction

Facial biometrics is a specific biometric mechanism that can allow a person's identity to be determined by analyzing his or her face, which is possible due to the fact that the face is a physical complex characteristic that makes it possible to distinguish and identify people with great accuracy.

Other biometric systems are capable of doing the same and are widely used is fingerprint biometrics. However, although apparently they are similar, they belong to different kinds of biometric systems because of different reasons: from an interaction perspective there are those which need physical contact or collaboration from the user, as in fingerprint biometrics, and those which don't need this, as in facial biometrics. From a security perspective, a fingerprint can be stolen when the person concerned is asleep or unconscios, while facial recognition often requires the eyes to be open and a natural facial expression to be maintained, which results in facial biometrics being considered a robust and accurate bio metric mechanism.

At this point a distinction must be made between the two main facial biometrics uses: *facial verification*, which is the domain of application of the model developed in this research; and facial recognition, as these do not have the same meaning. Facial recognition is based on the comparison of an image of a person against a known database such as, for example, a database of criminals held by the police, and there is not, generally, any output from the facial recognition or direct personal benefit to the individual (1-N). SMNLR model used a multi-class Support Vector Machine classifier, obtaining different results that are used to predict the accurate label from noisy labelled facial images [1]. On the other hand, facial verification is based on the comparison of two face images, and the output of the inference between them allows the result of the comparison to be determined, such as, for example, giving access to a service or space (1-1), and is an action that one is aware that one is doing and in fact is usually an action one chooses to take in order to gain access to some personal benefit. For this reason, accuracy and speed are the key attributes for the facial verification models.

* Corresponding author.

E-mail addresses: asier.martinez@ultratendency.com (A. Alcaide), mpatrici@inf.uc3m.es (M. A. Patricio), aberlan@ia.uc3m.es (A. Berlanga), aarroyo@etsisi.upm.es (A. Arroyo), jjcg@uah.es (J. J. Cuadrado-Gallego).

The performance of these verification models is usually evaluated by applying a confusion matrix. A confusion matrix is a two-dimensional matrix that represents all the evaluation results of a classifier with respect to some test data. The first dimension of the table represents the true class of an input, and the other dimension represents the value assigned by the classifier. In facial verification approaches, the confusion matrix forms 4 elements:

- True Positive, TP. Portion of results that the classifier predicted positive when the truth is indeed positive. Images that are the same person and indeed the model classifies them as two images of the same identity.
- True Negative, TN. Portion of results with a negative detection given that the actual instance is also negative. Images that are not the same person and indeed the model classifies them as different people.
- False Positive, FP. Portion of results with a positive detection given that the actual instance is negative. Images that are not the same person but the model has detected them as the same person.
- False Negative, FN. Portion of results with a negative detection when the actual instance is positive. Images that are the same person, but the model has detected them as different ones.

The wrong predictions in a face verification system are then the FP and FN. FN are important to consider, as facial verification models improve their usability when the user is not constantly and repeatedly trying to access any privilege depending on the specified application. However, FP and FN should not be considered with same importance. A high FP rate would affect the system in a very negative way, as the model will allow access or privileges to people that should not be granted to them.

To solve the problem of facial verification using neural networks, different methods have been published in the literature. Nowadays those methods can be distinguished between deep learning methods, the newest ones, and traditional methods, and the determining feature for this distinction lies in the recognition process followed by the model:

- Traditional methods are carried out in several phases: First a pre-processing phase is needed, followed by a phase of local feature extraction and feature transformation. It is possible that some of these steps can be improved separately, however none of these improvements have resulted in significant growth in accuracy. Furthermore, most of these methods are not capable of extracting stable characteristics that are invariant to real situations [2].
- Deep learning methods use a set of layers that learn different representations at multiple levels. The features obtained from these models are robust to variations in lighting, pose and expression.

The model presented in this paper can be classified as a Deep Learning Method, and for this reason, these will be dealt with in the introduction. One of the first deep learning architectures is the work with the DeepFace network. This architecture is composed of new layers of convolutional neural networks. In recent years, facial verification models have appeared that are built on deep convolutional neural networks (CNN): Facenet [3] maps images of faces to a compact Euclidean space using CNNs and then analyses similarities between faces; in [4], authors introduce a new loss function in the learning process of CNNs, that they call centre loss, which combined with the softmax function allows for greater discriminating power in face recognition systems; in [5], the authors propose normalisation operations on the layers of a CNN, as well as the loss functions necessary for training the normalised features; finally, in [6] they propose a new learning process based on angular softmax loss function in order to learn more discriminative features of their CNN, called SphereFace.

Regarding the results obtained by existing models, DeepFace [7] achieved a verification accuracy of 95.92% with the Labeled Faces in the Wild (LFW) dataset [8] [9]. Since that moment, more complex models of deep learning architectures have been published, such as the models mentioned in the previous paragraph; reaching the latest models published that outperform the previous ones in accuracy. The most recent ones are the ArcFace model [10], Circleloss-ResNet34 [11] and Prodpoly-ResNet [12] models, where a 99.53%, 99.73% and 99.83% in verification accuracy over the LFW dataset are achieved respectively. Typically, these complex models require specialized hardware to run, such as the ”Nvidia Titan” style GPU. These models cannot be run on devices with limited computing capabilities. In [13] the most recent works in this field can be found, including the current challenges of facial biometrics (different poses, changes in lighting or expressions, among others).

To design such powerful neural networks, specialized hardware is needed to reduce the training/inference time. Different proposals have emerged that allow the complexity of these networks to be reduced, such as the use of Binary Networks [14], [15], Network Pruning [16]–[18] or Mimic Networks [19], [20], among others. With these methods the time of training and inference is improved, supposing a small loss of precision. However, in the field of face verification, it is expected that the algorithms to be used are robust and, above all, do not allow the appearance of false positives. Nevertheless, the computation power needed is still too great to be executed in devices with low computation resources. Taking this into account, the aim of this article is the presentation of a new face verification system called Light Intrusion-Proving Siamese Neural Network, LIPSNN, that allows:

1. Creation of a facial verification system capable of being executed in devices with computational limitations.
2. A highly effective facial verification system to be obtained, against possible supplanting of authentication. That is to say, that it minimizes the occurrence of false positives.

The proposed model will use an architecture based on Siamese Convolutional Neural Networks. These architectures are based on the fusion of two parallel networks on which a cost function is applied, whose main task is to classify the characteristics formed from the networks. In consequence, a Siamese network consists of replicating part of the architecture of a neural network, and then merging them into one or more common layers, which allows results to be obtained through the entire previous process of both replications. This allows us to compare two inputs, in our case two images of people, extract the ”characteristics” of each of the inputs and perform any type of classifying method, which defines an output easily interpreted as a result, in this case an affirmative or negative depending on whether the person is correctly verified. The first Siamese models for facial verification emerged at the beginning of this century [21], where they used dimension reduction methods to later compare the characteristics between pairs of images. Later on, new models of Deep Learning appeared, using new ways to extract characteristics from users, as a multi-task learning of False Rejects, FR, and age estimation approach [22]; or a heterogeneous face recognition model published in [23] that consists of a visible and near-infrared pairs of images as input made thanks to a siamese network. However, the structure has always been the same and has not changed in its basic form: a symmetrical and independent part, together with a comparison between the two.

The rest of the paper is organized as follows. Section II presents the proposed Light Intrusion-Proving Siamese Neural Network architecture. Section III Describes the four steps performed to develop the model. Section IV provides the results of the multiple evaluations developed to compare the LIPSNN model performance with many of the traditional and deep learning models published. Section V presents the conclusions and future work.

II. LIPSNN Architecture

The architecture of Light Intrusion-Proving Siamese Neural Network is based on traditional Siamese networks but it introduces new characteristics that improve their performance for face verification fundamentally in two ways: Increasing the detection false positives and Reducing the latency. In addition, this must be achieved with limited computational resources. To do this, an architecture with two consecutive blocks, shown in Fig. 1 is defined, each one of them has the following characteristics and operation:

- Block I: Is a replication of two pre-trained deep learning models with exactly the same architecture, weights and biases. This block extracts the image features of two inputs. Each input is made up of the image of a facial identity, and each image is pre-processed and computed across a Convolutional Neural Network, obtaining as output one vector for each image, called *Bottleneck*.
- Block II: Is a small, light binary classifier, based on a Fully-Connected neural network. Instead of a basic point distance between the two outputs of Block I that traditional Siamese Neural Networks implement, this second block of LIPSNN model takes the two Bottlenecks obtained from the Block I, and *compares* them with the new neural network, to finally obtain the result predicting whether the pairs of images are or not the same person. Also a new penalization technique, which will be explained in detail later, has been implemented during the training phase, with the objective of incrementing the total loss of the batches with false positive cases, as these cases are considered with more priority than those of false negatives.

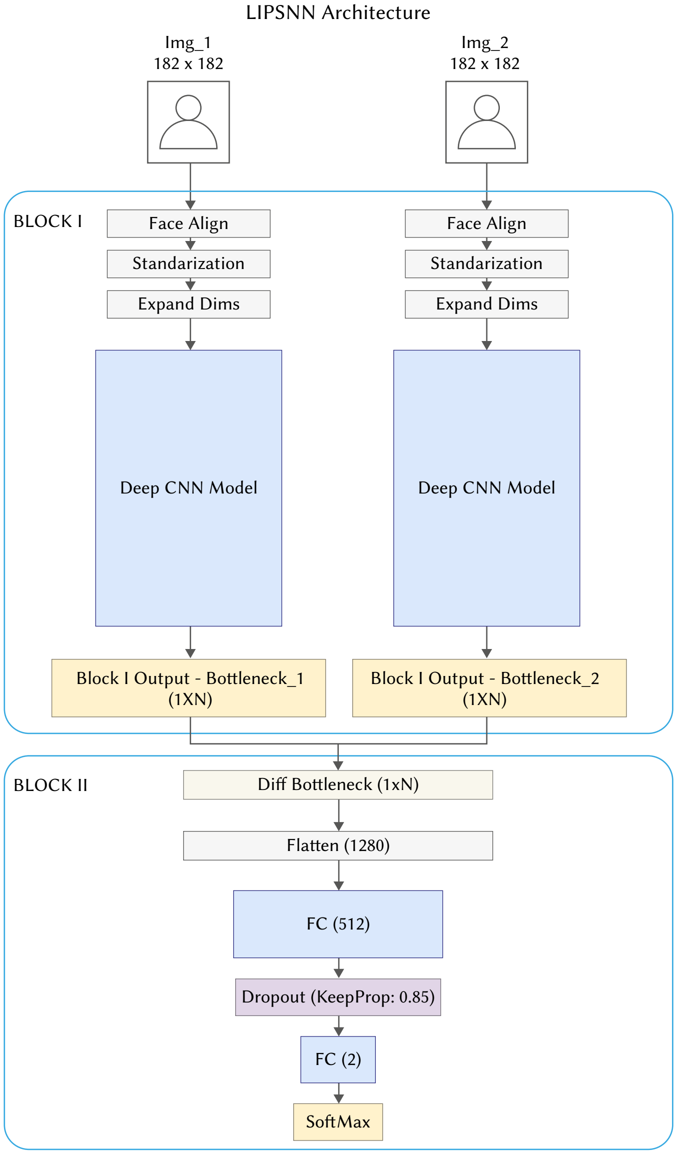


Fig. 1. LIPSNN architecture. Siamese Neural Network basis.

In the following section the architecture of each block is explained in detail.

A. Block I: Feature Extraction

As is well known, feature extraction is a crucial step in Deep Learning training steps and predictions. To deal with the solution of this problem, this block extracts all possible information from its inputs in order of greater or lesser importance depending on the selected convolutional architecture chosen for the model. Block I has the following characteristics:

- Both convolutional networks are completely identical, having the same internal structure, weights and biases.
- Both nets have already been trained and optimized by big organisations for face recognition purposes.
- The two nets have far more parameters and require more computational resources than Block II, made by only a binary classifier.
- The last layers have been removed from both blocks of each architecture made by fully-connected ones, in order to extract the bottleneck of each.
- Both bottlenecks are made of raw feature data extracted by the last convolutional output of the model architectures that has been chosen.

B. Block II: Binary Classifier

The second block is sequentially after the processing of the first Block and, as is shown in Fig. 2, consists of a supervised binary classifier that has as its input the absolute difference between each value of the bottleneck arrays, having the same size as them, this procedure allows the total number of inputs in this block model to be simplified, thus reducing the total number of parameters. As output, the model predicts whether the difference bottleneck obtained from the Block I corresponds to the same person or not. This output is normalized, giving a similarity value that is used to obtain the final prediction.

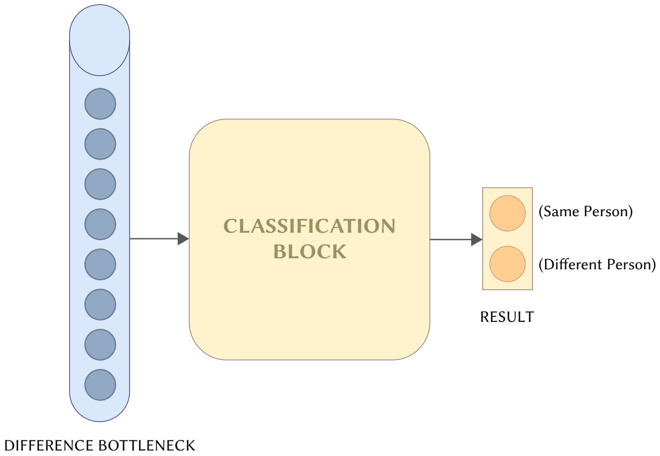


Fig. 2. Block II High-level Architecture.

The structure of the Neural Network consists of two Fully-Connected, FC, layers with 512 and 2 neurons respectively. It has been previously considered to output a single probability instead of two different values for both positive and negative results. However, this architecture has been used in order to collect every specific similarity exclusively when cases are positives, and evaluate them based on the probability threshold. For every positive value that doesn't reach the threshold set, it is automatically discarded as positive and changed into a negative case.

Moreover, as is shown in Fig. 3 a flatten layer has been added before the first FC layer in order to prepare the input dimensions; a dropout regularization technique in order to reduce the over-fitting; and a soft-max layer at the end of the network to normalize the outputs.A Sigmoid activation function has also been used to determine the output of the network.

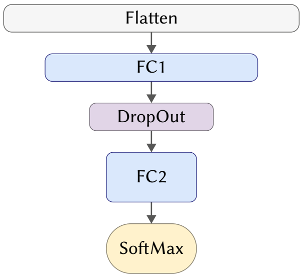


Fig. 3. Block II Low-level Architecture.

III. LISPNN Development

Having established the Light Intrusion-Proving Siamese Neural Network architecture in the previous section, in this section the process used to define will be is presented. This process consists of the following four phases:

A. Neural Network Model Selection

As described before, Siamese Neural Networks need a model that is used for the feature extraction before the comparison step, and in LISPNN this has been implemented with four models of two types specialized in facial purposes, with the objective of outperforming the current results in facial verification and exploiting the strengths of each of them. The four models used for the Block I, classified by type, are the following:

- InceptionResNet. This the first type of model used. From this type two versions have been used:
  1. InceptionResNetV1. This is a combination of two deep learning models with different characteristics: InceptionV3 [24] and ResNet [25].
  2. InceptionResNetV2 [26]. This is a second version and improvement of a combination of two previous architectures: Inception V4 [26] and the residual network techniques of ResNet's [25]. This net gives significant results thanks to the residual techniques, accelerating the training of Inception networks significantly with lower resources compared to others
- MobileNet. This is the second group of models used. They are designed to run in lighter environments, using fewer parameters. Two versions have been used as well:
  1. MobileNetV2. [27] This is an architecture specialised in light and resource-limited devices, an improvement of its previous version MobileNetV1. This net implements the residual connections technique, based on ResNet architectures.
  2. MobileNetV3 [28]. This has overtaken its previous versions in accuracy and latency, latency being one of its biggest improvements. Thus, the Siamese Network implemented in this model is one of the most promising architectures of the moment, considering the low number of parameters.

These pre-trained models can extract a huge amount of information from each facial image used as input compared to traditional Convolutional Neural Networks. In other words, they can extract as much information as traditional ones in a faster way.

B. Pre-Processing

For the training and inference steps or operations, the model needs to receive as inputs the images ready for it. To do so, the images received as inputs are treated using three techniques:

1. Normalization. Align and cropping techniques have been employed, using libraries, such as the Multi-Task Cascaded Convolutional Neural Networks (MTCNN) [29], employing the model to put great effort in noisy data and focus on the main problem of facial verification. During the construction of the training and evaluation data-set, all images have been prepared with this normalization process.
2. Data Augmentation. The data has been *augmented* or, in other words, new multiple data has been created based on the original ones during the training process, creating a more complete dataset. Moreover, random rotating and exposition, sizing, flipping and cropping techniques have been implemented on each training image.
3. Data-set filtering. For proper data preparation, the training data-set has been created with the selection, from the Labeled Faces in the Wild, LFW, data-set (this data set will be described in the following section *Training*), of the classes of people that have more than 15 images each. This technique increases the ease of learning for the model and, to avoid unbalance between classes, an upper limit of images per person has been established.

C. Training

This section presents the features of the LISPNN model training. It begins by describing the data set used, followed by a description of the details of the training of each block and finishes by describing the training of the model as a whole.

1. Training Data Set

The model has been trained with the well-known Labeled Faces in the Wild (LFW) dataset [8] [9], mentioned in the previous section. LFW contains images with faces of famous people obtained through the Internet. The dataset contains 13233 images of 5749 different people, where 1680 people have more than one image. Every person has a varying number of images in the database but, 1680 persons have at least two distinct images.

2. Training Block I

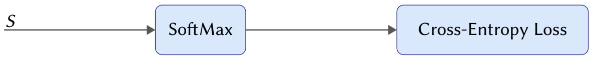
During the training phase, Block I has remained constant, as the architectures of Inception-Resnet and MobileNet are previously trained by external users. Inception-ResNet-V1 has been obtained from a contribution by David Sandberg [30] based on FaceNet [3], a face recognition system developed by researchers at Google with many competitive results. The model was trained by the VGGFace2 [31] training dataset available here. Inception-ResNet-V2, MobileNet-V2 and MobileNet-V3 architectures, weights and bias are obtained from the official GitHub Tensorflow repository [32], trained on the ILSVRC-2012-CLS image classification dataset [33].

These pre-trained models have been used to build the Block I architectures, removing the last classification layers and keeping only the ones that are used to make the feature extraction of the future input of images; and then taking the weights and biases of each of them, excepting those last layers.

3. Training Block II

Block II, however, has been trained in order to optimize its weights and biases so that it can get the best performance related to the minimization of false positive cases. To do so, a new technique has been developed during the training step.

The new technique is implemented in the loss strategy used, in order to reach better back-propagation results. In this case, a Softmax Cross-Entropy Loss strategy (also called Categorical Cross-Entropy loss) has been included, shown in Fig. 4 which consists of a Softmax activation plus a Cross-Entropy loss. By using this technique, the model output will be the probability over the C classes for each image. As the total classes of the full architecture are two (same person, and different person), the model will directly get as output the probability of both images being the same person in one class, and vice versa.



$$f(s)_i = \frac{e^{s_i}}{\sum_j^C e^{s_j}} \quad CE = -\sum_i^C t_i \log\left(f(s)_i\right)$$

Fig. 4. Softmax Cross-Entropy loss function and equation. [34].

Where, for a given class s_i, C is the number of classes; s_j are the scores inferred by the net for each class in C, t_i and s_i are the groundtruth and the CNN score for each class i in C.

The technique implemented in this paper consists of a **loss penalization in case of False Positives**. The model will modify and increment its loss proportionally to the number of False Positives found in every training input batch. The Equation 1 represents the loss function approach proposed.

$$Loss_f = \begin{cases} softmaxCrossEntropyLoss(logits, labels) * \alpha \\ if\ FP > 0 \\ softmaxCrossEntropyLoss(logits, labels) * 1 \\ if\ FP = 0 \end{cases} \tag{1}$$

Alpha determines the number of False Positives found per batch of predictions during the training phase; and *softmaxCrossEntrophyLoss* function calculates the loss between the predicted array results (logits) and the real array labels. This process allows the appearance of False Positives in the LIPSNN architectureto be penalized, this being this one of the main objectives of our proposal. With regards to mini-batches with false positives, the correct predictions are not only the ones without false positives, but also the ones with false negatives, suffering a considered loss that the model will process. However, in our approach, any prediction obtained from the ones that have false positives is considered as a common cross-entropy loss output, and each false positive found will linearly increment the total loss of the corresponding batch. Thus a proportionally higher loss is obtained if the total number of false positive cases increments. A batch with no false positives will generate usual loss according to every prediction. A variation of this α parameter permits a simple calibration of false positive/negative proportional rates to be generated. By increasing this value, it will proportionally increase the loss of the batches where false positives are detected, being stricter in the intrusion cases compared with false negative ones, and vice versa.

4. Training LIPSNN Model as a Whole

The training consists of a multiple hyper-parameter optimization for each of the possible combinations. These (hyper-)parameters are the following:

- Architecture of Block I: These are the pre-trained models used for the feature extraction of each image. Four models have been added:
  1. Inception-ResNet-V1
  2. Inception-ResNet-V2
  3. MobileNet-V2
  4. MobileNet-V3
- Seed: integer used for the weights and biases initialization state of a pseudo random number generator. Used for randomization control. Seeds set used:

  $$seed \in \{13, 25, 29, 31, 42, 51, 67, 80, 90\}$$
- Batch size: number of images per each training and evaluation iteration. Used:

  $$batch\_size \in \{8, 16, 32\}$$
- Max steps: maximum number of iteration in each training and evaluation step. Parameters set from 250 to 2000 steps.

  $$max\_steps \in \{250, 500, 1000, 2000\}$$
- Dropout: regularization technique for reducing overfitting in neural networks by preventing complex co-adaptations on training data. In other words, it consists of *dropping out* random neurons from the net. It has been kept in a 0.85 dropOut-Keep-Probability, which means a 15% of dropout.
- Learning Rate: determines the step size at each iteration while moving toward a minimum of a loss function. It represents the learning speed of a model. 0.01, 0.001 and 0.0001 Learning Rates have been used.

  $$learning\_rate \in \{0.01, 0.001, 0.0001\}$$

By combining these hyper-parameters, a total of 576 models have been trained in a laptop “Xiaomi Mi Laptop Pro 15,6 inch Intel Core i7-10510U NVIDIA GeForce MX250 16GB DDR4 RAM”. The software libraries and frameworks: Python 3.6.8, Tensorflow 1.14.0, Numpy 1.16.4, and OpenCV 3.4.2 .

D. Model Architectures: Comparison

In this section, experimental results for the evaluation of the four architectures implemented in LIPSNN model are given. Two different branches have been considered:

- Efficacy. Efficacy is related to the real performance, depending on the model precision, accuracy, etc.
- Efficiency. Efficiency is related to the model performance depending on the latency times, which mostly has the same behaviour as the total number of parameters per architecture.

For each model represented, their effectiveness and efficiency are presented and compared. After that, a unique architecture is chosen for the model results.

For the collection of the results, the evaluation dataset used is the official LFW test dataset [8], which consists of a thousand face pairs similar to the entire LFW dataset; five hundred image pairs with positive results (both images are the same person) and another five hundred with negative results (face images are not the same person). Each image pair also contains the result of the comparison, whether is positive or negative (1 and 0 respectively). It can be considered that, for a better robustness level of the results, a test dataset based on ten thousand image pairs with same number of positive and negative results has been used during the evaluation. Those image pairs obtain a better approximation of the effectiveness of the model.

TABLE I. Hyper-Parameter Selection of Each Architecture and Their $F_{0.5}$ Values

Architecture	BatchSize	MaxSteps	DropoutKeepProb	LearningRate	Seed	*MaxF$_{0.5}$Score*
InceptionResNetV1	8	500	0.85	0.001	29	0.974555
InceptionResNetV2	16	250	0.85	0.001	13	0.959028
MobilenetV2	8	250	0.85	0.001	13	0.959071
MobileNetV3	16	250	0.85	0.001	80	0.958183

1. Efficacy Evaluation

For the evaluation of the efficacy part, the $F_{0.5}$ score has been used as the most relevant metric. It is based on the F_β metric. Chinchor[53] defines this metric which will be used further on:

$$F_\beta = \frac{(\beta^2+1)\cdot PR}{\beta^2\cdot P + R} \quad (0 \le \beta \le +\infty) \tag{2}$$

Where P refers to the precision and R refers to the Recall obtained on each evaluation, and β is the parameter used to give a proportional weight importance between the Precision and Recall. Both P and R (True Positive Rate) are inversely related to the proportion of False Positives and False Negatives respectively.

For this study, a case that the False Positive errors are crucial to minimize, the β chosen has a value of 0.5. Thus, a bigger fluctuation in the value of F score will be obtained when the proportion of False Positives are modified when this is compared with a same fluctuation of False Negatives. Thanks to this metric number, the performance of the models that minimize the False Positive Rate can be better justified without ignoring the False Negatives Rate, a problem that appears when a hyper-parameter combination that minimizes the False Positives Rate is only considered, generating inappropriate results of the false negatives that are really high.

From the $F_{0.5}$ values obtained on each architecture, those that maximize this metric have been used. The Table I shows the hyper-parameter combination that maximizes each architec ture evaluation, Inception-ResNet-V1 obtaining the highest value.

Moreover, by getting the best hyper-parameter selections of each of the four architectures, an evaluation of the FPR and TPR has been made by making a complete variation of the similarity threshold and a representation of the results in the Receiver Operating Characteristic (ROC) curve in Fig. 5.

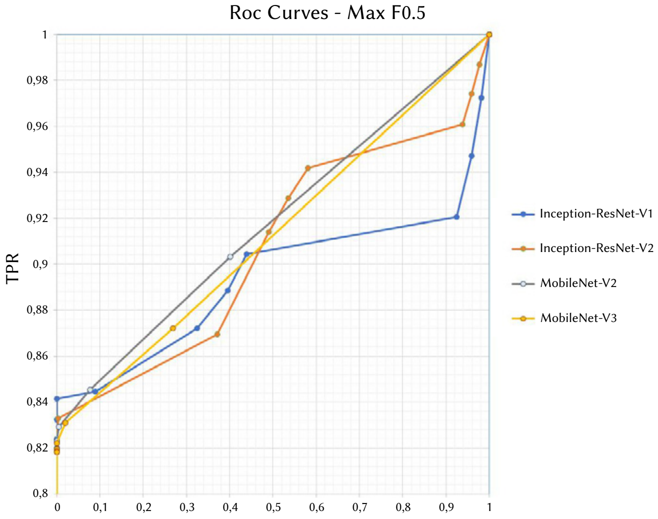


Fig. 5. ROC Curves of each architecture that maximize F0.5 values.

All architectures used for the LIPSNN model have similar results in this curve. As mentioned before, this happens as the training process has been carried out only in the second part, Block II, which consists of a basic neural network binary classifier.

After looking into the ROC curve, in Table II the way the Area Under the Curve (AUC) behaves can be seen.

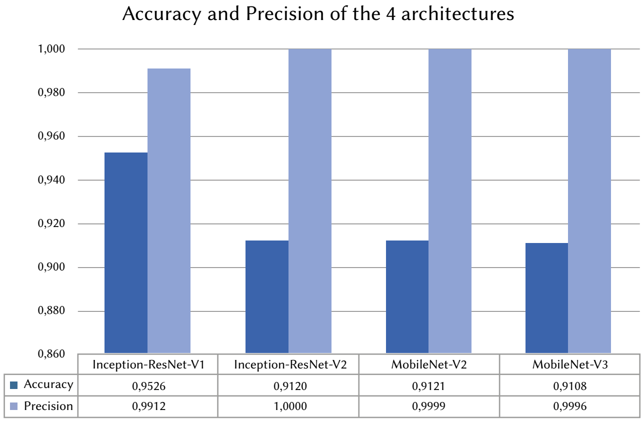
TABLE II. AUC Values of Each Architecture That Maximize $F_{0.5}$ Values

Architecture	AUC (Max $F_{0.5}$) (M)
InceptionResNetV1	0.903
InceptionResNetV2	0.921
MobilenetV2	0.957
MobileNetV3	**0.965**

Surprisingly, the Mobile-Net architectures reach a total of 0.965 and 0.95 values, outperforming the Inception-ResNet ones, with a total of 0.92 and 0.90. This anomaly happens due to the generalization that is made up from the AUC compared to the individual selection of hyper-parameters obtained in the table I. In other words, the models in the table are made up exclusively of one hyper-parameter setting per architecture, obtaining the best performing results possible, whilst the ROC and AUC may contain models of each architecture that decreases the total results per architecture, as they are based in multiple results depending on the similarity threshold.

Once the models are represented and analysed through the $F_{0.5}$ metric, it has been considered an accuracy and precision analysis used for the final conclusions of the architecture performance evaluation in terms of efficacy.

In Fig. 6, it is clearly seen that the results of both metrics are based on the hyper-parameter selection that maximizes $F_{0.5}$.



	Inception-ResNet-V1	Inception-ResNet-V2	MobileNet-V2	MobileNet-V3
■ Accuracy	0,9526	0,9120	0,9121	0,9108
■ Precision	0,9912	1,0000	0,9999	0,9996

Fig. 6. Accuracy and precision of the four architectures, using parameters that maximize $F_{0.5}$ values according to Table I.

The three Inception-ResNetV2, MobileNet-V2 and MobileNet V3 architectures have obtained similar results from both attributes, having almost 100% precision and around 91% accuracy each. On the other hand, the Inception-Resnet-V1 has obtained better accuracy by reducing the precision of the model. This contrast needs to be evaluated, as the main goal and purpose of this work consists in proposing a model that can ensure biometric authentication security by reducing access of undesirable intruders (False Positives), but at the same time it should remain an effective access for people that are allowed to access the hypothetical system.

2. Efficiency Evaluation

$F_{0.5}$, Box Plots, ROC curve and AUC, and Accuracy and Precision attributes are those used for an efficacy performance evaluation. In addition, efficiency evaluation is half of the importance when evaluating this model focused on mobile devices, where the latency and computation times are crucial for a proper evaluation.

Consequently, two tables are represented below in order to visualize the size of each architecture and their latency. Table III represents the total number of parameters that form each architecture, and Table IV shows the inference times they gave during the evaluation step, made in an isolated environment that allows the correct comparison between them.

TABLE III. Total Parameters Per Architecture Implemented for LIPSNN

Architecture	Parameters (M)
InceptionResNetV1	43
InceptionResNetV2	55.8
MobilenetV2	**6**
MobileNetV3	**5.4**

In Table III each of the sizes of the four architectures can be compared with the previous tables [27], [28], [35]. The Inception-ResNet-V1 value is represented as an approximation range due to the lack of information found in other publications. Szegedy, in his publication [26], affirms that this first version of Inception- Resnet has a lower number of parameters than its successor. In any case, it can be seen that both MobileNet architectures have values approximately ten times lower than Inception-ResNet ones.

TABLE IV. LIPSNN Inference Times of Each of the Four Architectures Used in Ms

Architecture	BlockI Inf	BlockII AvgInf	BlockII StdInf	Total TotalInf
InceptionResNetV1	121.2	3.5	0.3	124.7
InceptionResNetV2	271	1.3	0.2	272.2
MobilenetV2	52.5	3.4	0.5	**56**
MobileNetV3	67.1	1.2	0.1	**68.3**

In Table IV, the most relevant aspects to consider are the huge difference between the latency of Inception-ResNets and Mobile- Net ones. The first ones, with 124.7 and 272.2 milliseconds of processing time for each prediction. The second ones, surprisingly, reach a latency of 56 and 68.3 milliseconds. It is also interesting to see that most of the times are related directly to the times of Block I, as this block is the one with the pre-trained architectures, which is bigger than the binary classification model included in Block II. Moreover, it is important to mention that the total inference time in Block I is considered as a sequential computation of the two inferences of each image to be compared included as inputs. This means that the model considered as the worst case of inference can perfectly be implemented in a parallel computing hardware in order to make the inferences at the same time.

After all the analysis of the different architecture performance evaluations of efficacy and efficiency, considering the total accuracy and precision, the ROC curve and AUC, and the total number of parameters per architecture as well as the total number of hyper-parameters, the two networks that perform best in this analysis are the Mobile-Net architectures. Moreover, based on multiple publications [27], [28], [36], [37], the Mobile-Net-V3 generates much better performance than MobileNet-V2 in similar fields such as facial recognition and segmentation [28]. Thus, for the public model performance comparison shown in the following section, the Mobile-Net-V3 is the chosen architecture to be used as for the LIPSNN model evaluation.

IV. LIPSNN Performance Benchmarking

In this section, the results of the various evaluations that have been carried out in order to reach a comparison between the Light Intrusion-Proving Siamese Neural Network model developed and some of the best known models published currently for dealing with this problem are presented. The results obtained from the LIPSNN model to compare with the rest of publications are based on the ROC curve, a metric frequently used in this area. It can represent the False Positive rate, as well as indirectly the False Negative one.

A. Performance Benchmarking Attributes

Taking the purpose of this study into account, four attributes are used for the benchmarking of this research:

- Number of Parameters of the model. This attribute measures the model size with the number of parameters that the model needs. A higher number of parameters would decrease the efficiency of the model, as it would proportionally increase its inference time. Thus, a lower number of parameters of the model allows inferences to be executed faster than a model with high number of them. The metric used to measure this attribute is millions of parameters used, (Millions).
- Storage Space. This attribute measures the model size in information units. As the number of parameters, a higher storage space would increase the total evaluation time and hence the efficiency of the model. That means, the model will be slower in terms of each facial verification. The metric used for this attribute is Megabytes (MB).
- Accuracy. This attribute describes how the model performs across all classes: positives or negatives. It is calculated as the ratio between the number of correct predictions to the total number of predictions, and it is ranged between 0 and 1, or in percentage. A number close to 1 means a model with many correct predictions, either positive or negative. Thus, as mentioned in previous sections, the efficacy of the model would be incremented with an accuracy close to 1 and hence the verification error cases will be reduced. The metric used for this attribute is Percentage (%).
- Precision. This attribute describes the ratio between the number of Positive samples correctly classified to the total number of samples classified as Positive. Unlike the accuracy metric, precision describes how the model can perform with the False Positive cases. It is ranged between 0 and 1, or in percentage. A precision close to 1 proportionally means a low False Positive ratio. The metric used for this attribute is Percentage (%).

As the main objectives of this research is to enhance the facial verification performance with a model with the lowest computational resources used and, considering all previous attribute definitions mentioned above, what would be needed for this case is a model with the highest precision result possible, with the lowest number of parameters and storage space. With regard to the accuracy, a lower result of this attribute would not be crucial for the main objective, as what is really needed is to reduce the number of parameters and the storage space whilst precision remains at least at the same level, avoiding the cases of False Positives. This last attribute is crucial in this facial verification approach in order to minimize these cases.

B. Performance Benchmarking Data Set

For the evaluation and collection of the results, as in the comparison of the architectures used for the model, the LIPSNN model has been evaluated with the official LFW test dataset [38], made up of 1000 face pairs equally divided in 500 same and 500 different face matches. The LIPSNN model has been trained with 5760 face pairs based on the Labeled Faces in the Wild (LFW) dataset, equally distributed between negative and positive matches. Any of these training images are not overlapped with the testing ones, in order to avoid any over-fitting and biases between them.

In the evaluation dataset, each image pair contains the information about each of both images to introduce into the model, and the result of the comparison or label: ’0’ when the images are different people or ’1’ when they are the same person. This dataset is used in other models, whose results are included in the next sections.

C. Traditional Models Benchmarking

The early facial verification models do not implement any deep learning techniques as is done nowadays. The models use multiple facial verification techniques such as the large multiple distance metrics used by the LM3L model [39]; the Linear Discriminant Analysis (LDA) coupled with ”Within Class Covariance Normalization” (WCCN) in the DDML model [40]; or the intra personal focused feature extraction of the Similarity Metric Learning (SML) method [41]. A table of the traditional models obtained is shown in Table V. But although their approximations to solve the problem are quite different, and it is not possible to find in the literature results that include all the attributes defined for this research, because only their accuracy has been published, it is interesting to analyze their results with the same evaluation dataset that this work has used. Those results are in Table V.

TABLE V. Accuracy Benchmarking Between Traditional Models

Model	Accuracy (%)
PCCA	83.80
PAF	87.77
CSML + SVM	88.00
SFRD + PMML	89.35
LM3L	89.57
Sub-SML	89.73
DDML	90.68
VMRS	91.10

From the table it can be seen that although those models achieved acceptable accuracy results, because the models mainly make correct predictions, they still have results that can be outperformed with the new models. Moreover, there is a lack of information in these their results using this approach, as the number of parameters, storage space and precision would enhance the conclusions about them related to this study.

D. Deep Learning Models Benchmarking

Using the four attributes defined, number of parameters, storage space, accuracy and precision, the performance of twelve deep learning models for facial verification, including LIPSNN, have been compared. For most of them, eight models, it has been possible to obtain three of the four attributes used; number of parameters, storage space and accuracy from the published literature [42] [12] [11] [43] [3] [44] [45]; for other, CenterLoss [4] only two attributes have been published, parameters and storage space, with a result of 19.6M for the first and 99.28MB for the second; and for the other three, LBPNet LBPNet [46], High-dim LBP [47], and DeepID2 [48], it has only been possible to obtain their accuracy from the literature, with a result of 94.04%, 95.17% and 95.43% respectively.

The results of the bechmarking of the LIPSNN model, along with the rest of deep learning models for those which have at least three of the four attributes defined, can be found in Table VI. The same results with the addition of the model which only has results for parameters and storage space, that is, CenterLoss, are also presented in Fig. 7. To make the comparison between models easier and the impact of each attribute, and also their combination clearer in the performance of the model, Fig. 7 has been developed using a normalized value, between 0 and 1, of the attributes. This normalization has been calculated for parameters and storage space by dividing the rest of each value minus the maximum value observed for the attribute by the maximum value observed for the attribute minus the minimum value observed for it. To obtain the right scale for this study, in which, the minimum values for those attributes are the best ones, the amount obtained in the previous calculation has been subtracted from one. As attribute accuracy is directly measured in an scale from 0 to 100, its normalization to a range between 0 and 1, is obtained dividing each observed value by 100.

TABLE VI. Attributes Benchmarking Between Deep Learning Models

Model	Parameters (Millions)	Storage Space (MB)	Accuracy (%)
CircleLoss-ResNet34 [11]	60.5M	83MB	99.73
DeepFace [43]	120M	488MB	95.92
FaceNet [3]	140M	186MB	98.87
Light CNN A [44]	3.96M	26MB	97.97
Light CNN B [44]	5.56M	32.8MB	98.80
LIPSNN	5.50M	65.9MB	91.08
Prodpoly-ResNet [12]	14.70M	181.8MB	99.83
VGG [45]	27.75M	533MB	97.27

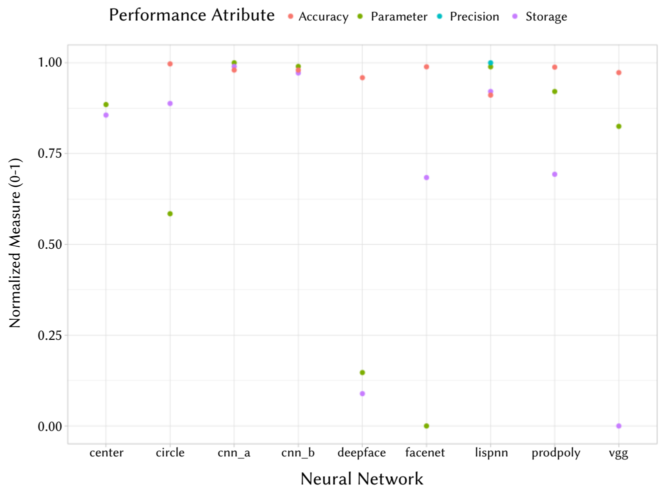


Fig. 7. Attributes Benchmarking between Deep Learning models.

The main purpose of this research was to reduce as far as possible the attribute parameters and storage needed by the deep learning neural network developed in order to be able to operate it in devices with low computation resources, achieving this without a signification loss of accuracy and precision, especially the latter, in the results in its application to facial verification. From Table VI and Fig. 7 it can be seen that both things can be obtained, because the normalized value, between 0 and 1 for parameters is 0.989 and for storage 0.921, both of them being very close to 1, which means that the computational resources needed by LIPSNN are very low; and that when they are compared with the other models are better than most of them, and in some cases significantly better, and are comparable with the best models.

At the same time that the desired results for parameters and storage have largely been obtained, the purpose of maintaining acceptable results on the second two attributes, accuracy and precision has also been achieved. A normalized accuracy of 0.911 is very close to 1. When it is compared with the rest of the models it is slightly lower but it is acceptable taking into account two important related considerations: the first is the good results obtained for parameters and storage, which are better than most of the published models; and the second, even more importantly, is the excellent results obtained for precision, which in its normalized value is 0.9996. It is not known if it has greater or lesser precision than the current models because this value has not been published for them, but with the obtained measure the other models can only present the same or lower value.

At this point is important to point out that the LIPSNN model has focused on a maximization of precision more than in the maximisation of accuracy. This is because the main goal in this work is to reduce the total number of intrusions in the model by reducing the False Positives ratio, and that is achived by increasing precision. Although accuracy must have as high value as possible, it has less importance precision in a facial verification when it is used alone. This is because, as it gives us together the number of the total false positives and negatives of a facial verification study, it doesn't give enough information to make it possible to distinguish between them in the results. For that reason, high measures of accuracy only means a good performance of the model when this is combined with high levels of precision and not only by itself.

As we have seen, LIPSNN is a lightweight network (it can be used in devices with scarce computational resources) with significantly good accuracy. However, the most important advantage of LIPSNN over other architectures is that the False Positive ratio is practically zero, thanks to the loss function used (see Fig. 7). This capability makes it especially useful for the verification of persons in critical services or facilities.

From the presented benchmaking study it can be concluded that the LIPSNN models have a parameters and storage results, with very low measures from both of them, that allow to it to be included within the group of the lightest models and that the LIPSNN model is a viable alternative model to them. If the sizing levels are combined with the precision results obtained, one of the most important conclusions is that, whilst the LIPSNN model doesn't achieve as good performance results in accuracy as if it does in precision, these are, nonetheless, good enough. It has an architecture and development different form the other previously published models, that is adaptable and modular, as is the case, for example, in block I, whose architecture could be replaced by lighter and more accurate and precise future deep learning models, which would allow its future improvement to continue and open up the development of a new alternative deep learning facial verification model, different from the existing previous ones.

V. Conclusions and Future Work

This study presented a solution focused on the design and development of a new deep neural network model for facial verification focused on the computational reduction resources needed for it to operate. The aim was for it to able to be executed in portable devices like mobile phones or tablets without loosing precision in the results. To achieve this the two sided problem of reducing parameters and storage without losing accuracy and precision has been overcome thorough the development of a new architecture and development of a Siamese Neural Network called Light Intrusion-Proving Siamese Neural Networks, LIPSNN.

For the development of the architecture, two blocks have been designed. For the first block, four pre-trained architectures specialized in facial verification were been analyzed in order to extract the face characteristics. For the second block, a binary neural network with a new loss function was developed in order to optimize the false positive cases. After the definition of the architecture, a process with four phases: model selection, pre-processing, training and architecture comparison, to define the neural network was carried out. All these processes were performed with the well-known LFW dataset and the following architecture models: Inception-ResNet-V1, Inception-ResNet-V2, MobileNetV2, MobileNetV3.

Once the LIPSNN design and development was finished, a performance benchmarking study with some of the best known models published nowadays for dealing with this problem was carried out. For this, four attributes were used: Number of Parameters of the model, Storage Space, Accuracy and Precision. The first two measure the size needed to execute the model, the second two, its validity for face recognition. From this comparison study it was concluded that the LIPSNN model requires a low number of parameters and storage needs that allow it to be classified in the set of lightweight models, while at the same time presenting very high levels of accuracy and precision, especially the latter. As the model presents quite different architectures and design to the previous light models published in the literature, it constitutes a new alternative to those ones.

Once a lightweight deep learning model based on a new architecture and design has been obtained, future research will focus on a continuous improvement in the reduction of the parameters and storage measures obtained and increasing accuracy without losing precision. To achieve this fundamentally it will focus on: the incorporation of new pre-trained architecture models in block I, as the LIPSNN model allows the modification of the architecture of the block where the facial features extraction is carried out; training using new modern and sophisticated datasets; and the design and development of new architectures of the binary classifier of Block II.

Acknowledgment

This study was funded by the private research project of Company BQ, the public research projects of the Spanish Ministry of Economy and Competitiveness (MINECO), references TEC2017-88048-C2-2-R, RTC-2016-5595-2, RTC-2016-5191-8 and RTC-2016-5059-8, and the Madrid Government (Comunidad de Madrid-Spain) under the Multiannual Agreement with UC3M in the line of Excellence of University Professors (EPUC3M17), and in the context of the V PRICIT (Regional Programme of Research and Technological Innovation).

References

[1] A. Suruliandi, A. Kasthuris, S. P. Raja, “Deep feature representation and similarity matrix based noise label refinement method for efficient face annotation,” *International Journal of Interactive Multimedia and Artificial Intelligence*, In Press, 2021, doi: 10.9781/ijimai.2021.05.001.

[2] Imatest, “Automated image quality analysis,” 2020. [Online]. Available: http://www.imatest.com/products/test-charts-sfrplus/.

[3] F. Schroff, D. Kalenichenko, J. Philbin, “FaceNet: A unified embedding for face recognition and clustering,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June-2015, 2015.

[4] Y. Wen, K. Zhang, Z. Li, Y. Qiao, “A discriminative feature learning approach for deep face recognition,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9911 LNCS, 2016.

[5] F. Wang, X. Xiang, J. Cheng, A. L. Yuille, “NormFace: L2 hypersphere embedding for face verification,” in *MM 2017 - Proceedings of the 2017 ACM Multimedia Conference*, 2017.

[6] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, L. Song, “SphereFace: Deep hypersphere embedding for face recognition,” in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-January, 2017.

[7] Y. Taigman, M. Yang, M. Ranzato, L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2014.

[8] G. B. Huang, M. Ramesh, T. Berg, E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," University of Massachusetts, Amherst, October 2007.

[9] G. B. Huang, E. Learned-miller, "Labeled faces in the wild: Updates and new reporting procedures," *University of Massachusetts Amherst Technical Report*, 2014.

[10] J. Deng, J. Guo, N. Xue, S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4690–4699.

[11] Y. Sun, C. Cheng, Y. Zhang, C. Zhang, L. Zheng, Z. Wang, Y. Wei, "Circle loss: A unified perspective of pair similarity optimization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6398–6407.

[12] G. G. Chrysos, S. Moschoglou, G. Bouritsas, J. Deng, Y. Panagakis, S. P. Zafeiriou, "Deep polynomial neural networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[13] U. Jayaraman, P. Gupta, S. Gupta, G. Arora, K. Tiwari, "Recent development in face recognition," *Neurocomputing*, 2020, doi: 10.1016/j.neucom.2019.08.110.

[14] M. Rastegari, V. Ordonez, J. Redmon, A. Farhadi, "XNOR- Net : ImageNet Classification Using Binary," *Eccv2016*, 2016.

[15] T. Simons, D. J. Lee, "A review of binarized neural networks," 2019. doi: 10.3390/electronics8060661.

[16] Z. Wang, F. Li, G. Shi, X. Xie, F. Wang, "Network pruning using sparse learning and genetic algorithm," *Neurocomputing*, vol. 404, 2020, doi: 10.1016/j.neucom.2020.03.082.

[17] F. Tung, G. Mori, "Deep Neural Network Compression by In-Parallel Pruning-Quantization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 3, 2020, doi: 10.1109/TPAMI.2018.2886192.

[18] Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, C. Zhang, "Learning Efficient Convolutional Networks through Network Slimming," in *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2017-October, 2017.

[19] Q. Li, S. Jin, J. Yan, "Mimicking Very Efficient Network for Object Detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 7341–7349.

[20] Y. Wei, X. Pan, H. Qin, W. Ouyang, J. Yan, "Quantization Mimic: Towards Very Tiny CNN for Object Detection," in *Computer Vision – ECCV 2018*, Cham, 2018, pp. 274–290, Springer International Publishing.

[21] S. Chopra, R. Hadsell, Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, vol. I, 2005.

[22] X. Wang, Y. Zhou, D. Kong, J. Currey, D. Li, J. Zhou, "Unleash the Black Magic in Age: A Multi-Task Deep Neural Network Approach for Cross-Age Face Verification," in *2017 12th IEEE International Conference on Automatic Face Gesture Recognition (FG 2017)*, 2017, pp. 596–603.

[23] C. Reale, N. M. Nasrabadi, H. Kwon, R. Chellappa, "Seeing the Forest from the Trees: A Holistic Approach to Near-Infrared Heterogeneous Face Recognition," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2016.

[24] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, "Rethinking the inception architecture for computer vision. arxiv 2015," *arXiv preprint arXiv:1512.00567*, vol. 1512, 2015.

[25] K. Z. He, K. Zhang, "X., ren, s. & sun j. deep residual learning for image recognition," *Preprint at https://arxiv. org/abs/1512.03385*, 2015.

[26] C. Szegedy, S. Ioffe, V. Vanhoucke, A. A. Alemi, "Inception- v4, inception-ResNet and the impact of residual connections on learning," in *31st AAAI Conference on Artificial Intelligence, AAAI 2017*, 2017.

[27] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L. C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018.

[28] A. Howard, M. Sandler, B. Chen, W. Wang, L. C. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, R. Pang, Q. Le, H. Adam, "Searching for mobileNetV3," in *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2019- October, 2019.

[29] Z. Mian, W. Hong, "Face verification using gabor wavelets and AdaBoost," in *Proceedings - International Conference on Pattern Recognition*, vol. 1, 2006.

[30] D. Sandberg, "Face Recognition using Tensorflow." https://github.com/davidsandberg/facenet, 2018.

[31] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, A. Zisserman, "Vggface2: A dataset for recognising faces across pose and age," in *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*, 2018, pp. 67–74, IEEE.

[32] N. Silberman, S. Guadarrama, "TensorFlow-Slim image classification model library." https://github.com/ tensorflow/models/tree/master/research/slim, 2016.

[33] J. Deng, A. Berg, S. Satheesh, H. Su, A. Khosla, F. Li, "Large scale visual recognition challenge 2012 (ilsvrc-2012)," 2012.

[34] R. Gomez, "Understanding categorical cross-entropy loss, binary cross-entropy loss, softmax loss, logistic loss, focal loss and all those confusing names," *URL: https://gombru.github.io/2018/05/23/cross_entropy_loss/ (visited on 29/03/2019)*, 2018.

[35] S. Bianco, R. Cadene, L. Celona, P. Napoletano, "Benchmark analysis of representative deep neural network architectures," *IEEE Access*, vol. 6, 2018, doi: 10.1109/ACCESS.2018.2877890.

[36] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," 2017.

[37] J. Hu, L. Shen, S. Albanie, G. Sun, E. Wu, "Squeeze-and-Excitation Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 8, 2020, doi: 10.1109/TPAMI.2019.2913372.

[38] G. B. Huang, M. Ramesh, T. Berg, E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," University of Massachusetts, Amherst, October 2007.

[39] H. Junlin, J. Lu, Y. Junsong, T. Yap-Peng, "Large Margin Multi-metric Learning for Face and Kinship Verification in the Wild," in *Computer Vision – ACCV 2014*, Cham, 2015, pp. 252–267, Springer International Publishing.

[40] O. Barkan, J. Weill, L. Wolf, H. Aronowitz, "Fast high dimensional vector multiplication face recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013.

[41] Q. Cao, Y. Ying, P. Li, "Similarity metric learning for face recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013.

[42] "12th Chinese Conference on Biometric Recognition, CCBR 2017," 2017.

[43] O. M. Parkhi, A. Vedaldi, A. Zisserman, "Deep face recognition," in *Proceedings of the British Machine Vision Conference (BMVC)*, September 2015, pp. 41.1–41.12, BMVA Press.

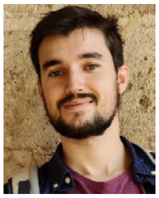
[44] X. Wu, R. He, Z. Sun, T. Tan, "A light CNN for deep face representation with noisy labels," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, 2018, doi: 10.1109/TIFS.2018.2833032.

[45] K. Simonyan, A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[46] M. Xi, L. Chen, D. Polajnar, W. Tong, "Local binary pattern network: A deep learning approach for face recognition," in *Proceedings -International Conference on Image Processing, ICIP*, vol. 2016-August, 2016.

[47] D. Chen, X. Cao, F. Wen, J. Sun, "Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2013.

[48] Y. Sun, Y. Chen, X. Wang, X. Tang, "Deep learning face representation by joint identification-verification," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, K. Q. Weinberger Eds., Curran Associates, Inc., 2014, pp. 1988–1996.



Asier Alcaide

Asier Alcaide has a degree in Computer Engineering and Business Administration from the University Carlos III of Madrid, Spain (2020). He has participated with UC3M and other enterprises in the research and development of artificial intelligence and machine learning technologies, and he was involved in machine learning competitions in the Kaggle community. He is currently focused on big data solutions combined with AI, and new disruptive biometric sensor devices.

Miguel A. Patricio

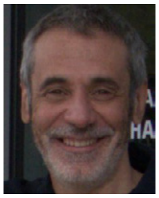
Received his BSc in Computer Science in 1991, his MSc in Computer Science in 1995 and his Ph.D. in Artificial Intelligence in 2002, all from the Universidad Politécnica de Madrid. He is currently an Associate Professor at the Escuela Politécnica Superior of the Universidad Carlos III de Madrid. He is the coauthor of over 100 books, book chapters, journal papers, technical reports, etc., published by organizations including Elsevier, IEEE, ACM, AAAI, Springer Verlag, Kluwer, etc., and most of these present practical and theoretical achievements of data analysis, computer vision and distributed systems.

Antonio Berlanga

Antonio Berlanga holds a Ph.D in Computer Science from Universidad Carlos III de Madrid (Spain) in 2000 and a B.S. degree in Physics from Universidad Autónoma de Madrid (Spain), in 1995. He is Associate Professor at the Universidad Carlos III de Madrid since 2000. His main research lines are foundation of multiobjective evolutionary computation and applications of artificial intelligence and decision support techniques in business and society.

Angel Arroyo

Received his BSc in Computer Science in 1999 from Universidad Carlos III de Madrid and his Ph.D. in Artificial Intelligence in 2011 from Universidad de Alcalá de Henares de Madrid. He is currently a Professor at the Escuela Politécnica Superior de Ingenieros en Sistemas Informáticos in Universidad Politécnica de Madrid. He has a long track record in research projects in the field of Artificial Intelligence, mainly in Computer Vision, Evolutionary Computing and Virtual Environments. In recent years, his main line of research focuses on the study of 3D persistent spatio- temporal environments, in which he has developed cognitive models and software tools for the construction of autonomous agents.



Angel Arroyo

Received a BSc in Physics in 1994 and a MRes in Physics in 1996, both from the University Complutense de Madrid; he also received a Ph.D. in Computer Science Engineering in 2001, from the University Carlos III de Madrid. He is currently an Associate Professor at the Computer Science Department of the University of Alcalá, in Madrid, Spain, and Affiliate Associate Professor at the Department of Computer Science and Software Engineering of the Gina Cody School of Engineering and Computer Science of the Concordia University, in Montreal, Canada. He is author or coauthor of over 100 scientific publications in Computer Science, principally in the fields of Data Science and Software Engineering.

Towards a Robust Thermal-Visible Heterogeneous Face Recognition Approach Based on a Cycle Generative Adversarial Network

Nadir Kamel Benamara[1]*, Ehlem Zigh[2], Tarik Boudghene Stambouli[1], Mokhtar Keche[1]

[1] Laboratoire Signaux et Images, Université des Sciences et de la Technologie d'Oran Mohamed Boudiaf, USTO-MB, BP1505, El M'naouer, 31000, Oran (Algeria)
[2] Laboratoire LaRATIC, Institut National des Télécommunications et des TIC d'Oran, BP 1518, El M'nouer, 31000 Oran (Algeria)

Received 6 March 2021 | Accepted 18 October 2021 | Published 20 December 2021

LA UNIVERSIDAD EN INTERNET

Abstract

Security is a sensitive area that concerns all authorities around the world due to the emerging terrorism phenomenon. Contactless biometric technologies such as face recognition have grown in interest for their capacity to identify probe subjects without any human interaction. Since traditional face recognition systems use visible spectrum sensors, their performances decrease rapidly when some visible imaging phenomena occur, mainly illumination changes. Unlike the visible spectrum, Infrared spectra are invariant to light changes, which makes them an alternative solution for face recognition. However, in infrared, the textural information is lost. We aim, in this paper, to benefit from visible and thermal spectra by proposing a new heterogeneous face recognition approach. This approach includes four scientific contributions. The first one is the annotation of a thermal face database, which has been shared via Github with all the scientific community. The second is the proposition of a multi-sensors face detector model based on the last YOLO v3 architecture, able to detect simultaneously faces captured in visible and thermal images. The third contribution takes up the challenge of modality gap reduction between visible and thermal spectra, by applying a new structure of CycleGAN, called TV-CycleGAN, which aims to synthesize visible-like face images from thermal face images. This new thermal-visible synthesis method includes all extreme poses and facial expressions in color space. To show the efficacy and the robustness of the proposed TV-CycleGAN, experiments have been applied on three challenging benchmark databases, including different real-world scenarios: TUFTS and its aligned version, NVIE and PUJ. The qualitative evaluation shows that our method generates more realistic faces. The quantitative one demonstrates that the proposed TV-CycleGAN gives the best improvement on face recognition rates. Therefore, instead of applying a direct matching from thermal to visible images which allows a recognition rate of 47,06% for TUFTS Database, a proposed TV-CycleGAN ensures accuracy of 57,56% for the same database. It contributes to a rate enhancement of 29,16%, and 15,71% for NVIE and PUJ databases, respectively. It reaches an accuracy enhancement of 18,5% for the aligned TUFTS database. It also outperforms some recent state of the art methods in terms of F1-Score, AUC/EER and other evaluation metrics. Furthermore, it should be mentioned that the obtained visible synthesized face images using TV-CycleGAN method are very promising for thermal facial landmark detection as a fourth contribution of this paper.

Keywords

Deep Learning, Generative Adversarial Network, Heterogeneous Face Recognition, Thermal Sensor.

DOI: 10.9781/ijimai.2021.12.003

I. Introduction

Automatic identification has become a crucial routine task to recognize individuals efficiently without any human interaction, in many applications, such as e-payment [1], people tagging in social media, gaming,...etc. Based on invariant biological and/or behavior characteristics [2], like face, iris, palm, fingerprint signature or gait recognition, biometric technologies are mainly used for security, like access control and criminal identification.

* Corresponding author.
E-mail address: nadirkamel.benamara@univ-usto.dz

Face modality has many advantages. It is a natural recognition procedure, generally accepted by everyone. It is cheaper in comparison to iris or fingerprint recognition modalities, since the price of cameras has become more accessible with the massive development of CMOS/CCD sensors. It is easily done, non-intrusive and above all, it works at a distance and unobtrusively, which makes it suitable for highly populated places, such as airports or bus stations. Therefore, this modality has attracted several scientific researchers. We can cite for example [3]–[6] which have reached a significant level of recognition accuracy for controlled environments. Nevertheless, the performances of those systems, which use the visible light, with wavelength ranging from 0.4 μm to 0.8 μm, are highly dependent on lighting quality and intensity and cannot be used at all for night applications. Furthermore, they face other challenges, such as pose and expression variations, and face disguises. To overcome these challenges, several alternative approaches have been proposed, they are mainly based on 3D or Infrared (IR) imagery.

With invariance to brightness changes as the main asset, IR imaging has emerged as a particularly promising research direction in the facial biometrics field [7]–[10]. It is a burgeoning sensor modality that could further be divided into two categories:

- **Active infrared**: it relies on signal reflected from objects illuminated by an infrared beam. It includes near-infrared (NIR) (0.74 μm - 1 μm) and short-wave infrared (SWIR) (1 μm - 3 μm).
- **Passive infrared**: it is based on body emitted radiation measurements, commonly known as thermal infrared. It comprises middle wave infrared (MWIR) (3 μm - 5 μm) and long-wave infrared (LWIR) (8 μm – 14 μm).

A Face Recognition (FR) system that aims to work under different illumination scenarios, during daytime as well as nighttime, should take into consideration both visible and infrared images. NIR and SWIR use an external active illuminator, which makes the active IR based FR systems improper for highly covert applications, since such an illuminator is easily detectable. Another major drawback for security applications is that NIR and SWIR images do not allow liveness verification natively, which makes the FR systems based on them, like those based on the visible images, vulnerable to spoofing attacks by just a photo or a video record (see Fig. 1). To overcome this issue, [11] has proposed eye-blink liveness checking. It achieves an interesting performance, but it was far away to be robust against the recent spoofing techniques, like 3D mask replica [12], [13]. Based just on body-emitted radiation, MWIR/LWIR remains an efficient solution against these issues (Fig. 2). It should be mentioned that few works have addressed the LWIR-VIS cross-spectral matching problem compared to the NIR-VIS problem, because the challenge intricacy increases proportionally with the spectrum wavelength. The authors in [14] have studied the modality gap based on the structural similarity index (SSIM) as a quantitative measure, and have obtained 0.335 for LWIR-VIS scenario and 0.581 for NIR-VIS scenario. Furthermore, public thermal LWIR face databases are less available than NIR databases [14], which impedes researches in the heterogeneous LWIR-VIS face recognition field.

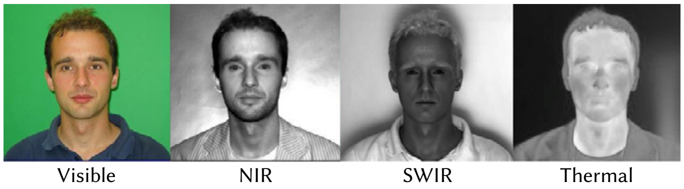


Fig. 1. Different images of a same subject in different imagery bands (Visible and Infrared), from UL-FMTV Database [15].

We have chosen, in this paper, to take up all the challenges cited above. For that, we propose an effective and robust thermal LWIR/Visible heterogeneous face recognition approach that includes four main contributions:

- Full manual annotations of an existent thermal face database are proposed for the scientific community to develop future thermal face detectors.
- A Multi-sensors detector based on the recent YOLO v3 architecture is proposed for face detection in visible as well LWIR imagery.
- A Cycle GAN with modified loss function (called TV-CycleGAN) is proposed to synthesize visible faces from LWIR faces, in different real-world scenarios, reducing the cross-spectral modality gap.
- A promising face landmark detection in thermal images is proposed. It is based on those obtained from the generated face images using our proposed TV-CycleGAN.

II. Background

Several face recognition methods using two spectra have been developed over the last decade [7], [16], [17]. We can distinguish two main approaches: multispectral face recognition (MFR) and heterogeneous face recognition (HFR).

In the MFR approach, the facial recognition system is mainly based on a visible-infrared fusion step, which can be performed at three different levels: data level, feature level or match score level. The first level methods aim to obtain, from two or more coregistered images, one richer image that is used in the feature extraction step [18]–[20]. In the feature level fusion scheme, extracted characteristics from both spectra are merged in order to gather more informative and discriminative feature vectors [21]. This approach has been adopted to fuse extracted characteristics from both visible and LWIR or NIR images to construct a face recognition system in [22], [23]. For the score fusion scheme, after a score normalization process, classification scores are combined in order to improve the recognition performance [24].

In many real-world scenarios, we have images from the infrared spectrum only, since surveillance cameras often capture faces in low light conditions or in total darkness. However, most datasets accessible to law enforcement have been collected in the visible spectrum. Therefore, there exists a need to match IR images to visible face images. We aim in this work to deal with this heterogeneous face recognition challenge.

There are three main approaches in HFR field, which are common subspace, invariant features, and image synthesis. The first approaches aim to project heterogeneous domain spaces to a common subspace to ensure a better measure and more appropriate comparison than the original distributions. For example, the authors in [25] proposed a method, called Common Discriminant Feature Extraction (CDFE), where face images from near-infrared and visible sensors are projected into a common feature space, such that the intra-class gap is minimized and the inter-class gap is maximized. In [26] the canonical correlation analysis (CCA) is used in order to maximize the correlation between the near-infrared and visible imagery domains. Mapping all thermal LWIR and visible images onto a common subspace suffers from being computationally expensive and requires a big amount of pre-processing [27].

In invariant feature domain approaches, scientific researchers use local handcrafted features in order to compare face images gathered from different spectra. [28] for example, proposed light source invariant features (LSIF) to cancel heterogeneities for a NIR-Visible matching. Huang et al. [29] proposed three different modality invariant features, quantized distance vector (QDV), sparse coefficient (SC) and least square coefficient (LSC) as encoding to resolve a NIR-Visible heterogeneous face recognition problem. In the third approach, synthesized images in the reference domain are generated from a probe domain and vice versa to apply traditional homogeneous methodologies. In [27], a joint dictionary learning mapping has been proposed for image reconstruction from visible to NIR domain and vice versa. In [30], using a deep CNN network, visible faces are reconstructed from near-infrared images, by using a cross-modal hallucination at the input and a low-rank embedding at the output. In [31], a cascaded refinement network (CRN) has been proposed for LWIR to visible-like images synthesis. Kantarci et al [32] proposed to first apply the difference of Gaussians filter (DoG) and face alignment, using manually annotated facial landmarks, as a preprocessing stage, then they used a deep auto-encoder architecture based on U-Net network to learn the mapping from thermal face images to visible-like face images.

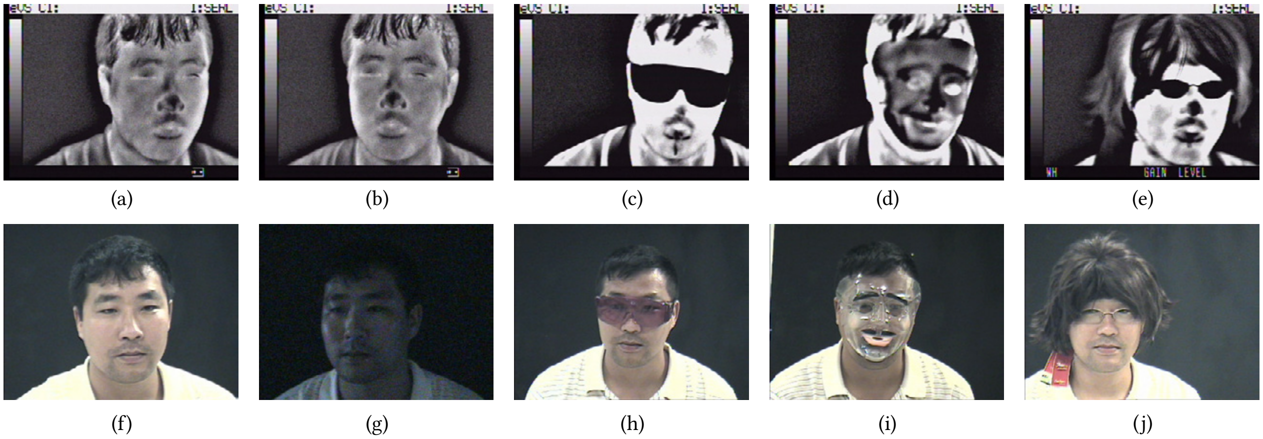


Fig. 2. Thermal imagery advantages under night and spoofing with disguise scenarios, (Up) LWIR Thermal (Down) Visible , (a,f) normal - (b,g) dark - (c,h) disguise with goggle - (d,i) disguise with mask - (e,j) disguise with wig.

Synthesis image approaches in HFR are based on cross-domain translation to reduce the modality gap. Therefore, we believe that generative adversarial network (GAN) [33] is an emerging and prosperous technique that may be applied in this field. Song et al [34], have introduced adversarial learning in raw-pixel and compact feature space to perform a NIR-VIS verification. In regard to the LWIR-VIS matching case, which could be considered as the most complicated case as aforementioned (lowest SSIM), there are very few scientific works in the literature. We can cite for example [35], where they have proposed the semantic guided GAN (SG-GAN) by assigning semantic labels, gathered from a face parsing network to semantic losses, to regularize the adversarial training. In [36], Zhang et al have proposed a TVGAN to generate realistic visible faces from LWIR thermal faces; however, all extremes poses and expressions were excluded from the experiences. Also, they did not provide any details related to automatic facial detection, which it is not a trivial task in LWIR imagery. Recently, Chu et al [37] proposed the multi-scale image synthesis method to translate thermal face images to the visible sensor modality. Their method is based on a GAN model to which they added the feature embedding, the facial landmarks and the identity preservation losses to their baseline loss function.

Other works are based on the polarimetric thermal infrared acquisition. This technique is used to achieve an improved performance since it retrieves the geometric and the textural face details. Xing et al [38] proposed the multi-scale attribute preserving GAN (Multi-AP-GAN) to synthesize visible face images from their corresponding polarimetric thermal face images for cross-modality face verification, by guiding the generator network with extracted attributes from the visible face images using a pre-trained VGG-Face network. He et al [39] proposed a visible face synthesis GAN-based (GAN-VFS) to generate visible faces from polarimetric thermal faces.

The Table I summarizes the related works of the synthesisbased image approaches in LWIR/Polarimetric to Visible HFR with their reported performances, including some comments and drawbacks.

In this scientific research, we propose a new thermal LWIR-VIS image synthesis method based on Cycle GAN, with a modified loss function. It includes all extreme poses and facial expressions in the evaluation protocol. In addition, as the cross-spectral translation depends highly on the pre-processing stage, we propose a simultaneous face detection technique in both spectra (visible and thermal LWIR) to automate the face-cropping phase. It is based on the last YOLO v3 deep architecture, trained using the WIDER visible face DB and the Terravic thermal DB, that we have annotated.

III. Proposed Approach

The proposed heterogeneous face recognition system is shown in Fig. 3. It includes three main parts:

- Multisensor Face Detection
- Face synthesis using TV-CycleGAN
- Face recognition

In addition to the database annotations, our scientific contribution concerns the first two parts.

We notice that the proposed TV-CycleGAN method used to generate visible faces from thermal images has a direct impact on face recognition process. We have chosen to focus on the thermal to visible face synthesis since most existing stored databases only contain visible face imagery of individual of interests [14]. In addition, the LWIR-VIS heterogeneous face recognition scenario is the most interesting for many security applications.

The results of the proposed face synthesis method using TVCycle GAN show also an interesting contribution in the face landmark detection field (Fig. 3). We will show more results in a later section.

A. The Proposed YOLO v3 Based Multi-sensors Face Detector

Face detection aims to locate the face coordinates in an image. Several face detection techniques have been proposed in the literature [41]. They can be categorized into two approaches: a region proposal approach and a regression/classification approach. The first approach, including RCNN [42], Fast-RCNN [43] and Faster- RCNN [44], adopts a pipeline of two stages: Object selection and classification. The first stage consists of selecting similar regions (same color, texture, or features). It is followed by a classification of each selected region. The regression/classification based approach, to which belong YOLO [45] and SSD [46] methods, is one-step straightforward. In this approach, images are firstly divided into grid cells where an object classification is carried out, reducing the time processing for region selection required in the first approach.

Unlike the R-CNN network versions that have used the regional proposal approach for the region selection, the YOLO network is faster and illegible for real-time applications, with comparable detection accuracy. YOLO v3 [47] is the last version, it incorporates the darknet 53 architecture, with skipped connections for feature extraction, and 53 other added convolution layers, for object detection, giving a fully convolutional network of 106 layers. Predictions are made at three different scales. As a result, this new architecture is more robust for the smallest objects than the early YOLO versions and has a better accuracy at overall.

TABLE I. Overview of the Related Works in Heterogeneous Face Recognition (Thermal to Visible) Using the Synthesis-Image Approach (RA: Recognition Accuracy, EER : Equal Error Rate)

Infrared to Visible	Reference (Year)	Synthesis Method	Pretrained Recognition Model	Database	Reported Performance	Comments/ Drawbacks
LWIR to Visible	[36] (2018)	TV-GAN	Mat-Conv-Net (VGG Face)	IRIS	RA: 13.9% (with 1 image/subject in the gallery) RA: 19.9% (with 4 images/subject in the gallery)	• Repeated angles, extreme poses, expressions, and illumination have been excluded from the experiments. - It produces low performances. • The equal Error Rate (EER) is not calculated.
	[31] (2019)	CRN	OpenFace	VIS-TH Eurecom	RA: 15.37%	• The explored database includes just 50 subjects. • The authors used Cascaded Refinement Networks (CRN) to consider multiple scales of images. It is different to generative adversarial networks concept. • Equal Error Rate (EER) is not calculated.
			LightCNN		RA: 57.612%	
	[35] (2019)	SG-GAN	AM-Softmax	Army Research Laboratory (ARL)	EER (LWIR to Visible): 14.24%	• As the number of subjects in the ARL dataset is not large, the recognition model has been trained on a larger MWIR dataset (PCSO), and fine-tuned using the ARL training set.
	[32] (2020)	DoG Filter + Autoencoder	Not Specified	Carl	RA: 48% (with 1 image/subject in the gallery) RA: 88.33% (with all images/subject in the gallery)	• The method uses autoencoders to learn Thermal-Visible mapping. • The used autoencoder is based on a modified U-Net architecture • The preprocessing stage does not include an automatic face detection stage. • The Carl database has been aligned using a manually annotated landmarks. • Equal Error Rate (EER) is not calculated.
				VIS-TH Eurecom	RA: 57.91% (with 1 image/subject in the gallery) RA: 85% (with all images/subject in the gallery)	
				UND-X1	RA: 58.75% (with 1 image/subject in the gallery) RA: 87.2% (with all images/subject in the gallery)	
	[37] (2021)	Multi-scale Image Synthesis	LightCNN	VIS-TH Eurecom	RA: 58.27%	Three losses have been added to the GAN loss function: feature embedding, identity preservation and facial landmarks losses • The equal error Rate (EER) is not calculated.
	[38] (2021)	Multi-AP-GAN	VGG-Face	VIS-TH Eurecom	EER: 25.68%	• The loss function of the generator network includes five losses: multi-scale adversarial loss, perceptual loss, identity loss, attributes loss and target-reconstruction loss.
				TUFTS	EER: 31.14%	
Polarimetric to Visible	[39] (2017)	GAN-VFS	VGG-Face	Army Research Laboratory (ARL)	EER (LWIR to Visible): 27.34 % EER (Polar to Visible): 25.17 %	• Polarimetric thermal imagers are very expensive for a daily life application compared to FLIR thermal imagers. • The polarimetric databases include geometric and textural details of faces that are not present in the thermal faces images.
	[40] (2018)	Multiple-Region Based Method	VGG-Face	Army Research Laboratory (ARL)	EER (LWIR to Visible): 26.25 % EER (Polar to Visible): 21.46%	
	[38] (2021)	Multi-AP-GAN	VGG-Face	Extended Army Research Laboratory (Ext-ARL)	EER (LWIR to Visible): 19.25 % EER (Polar to Visible): 17.81 %	

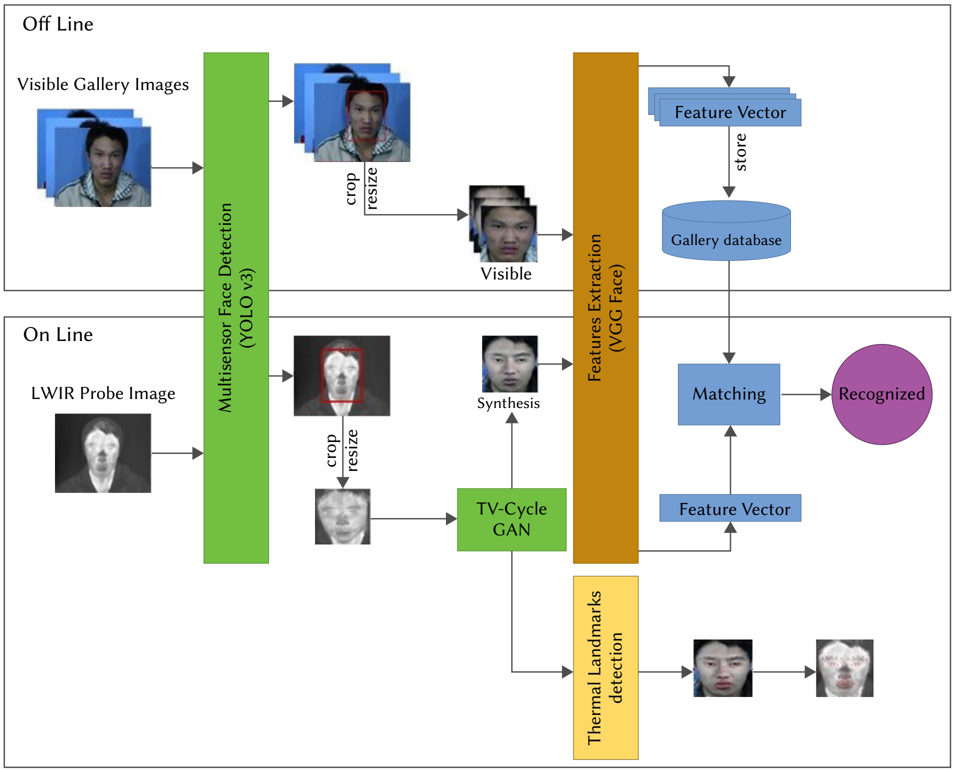


Fig. 3. Flowchart of the proposed Heterogeneous Face Recognition System.

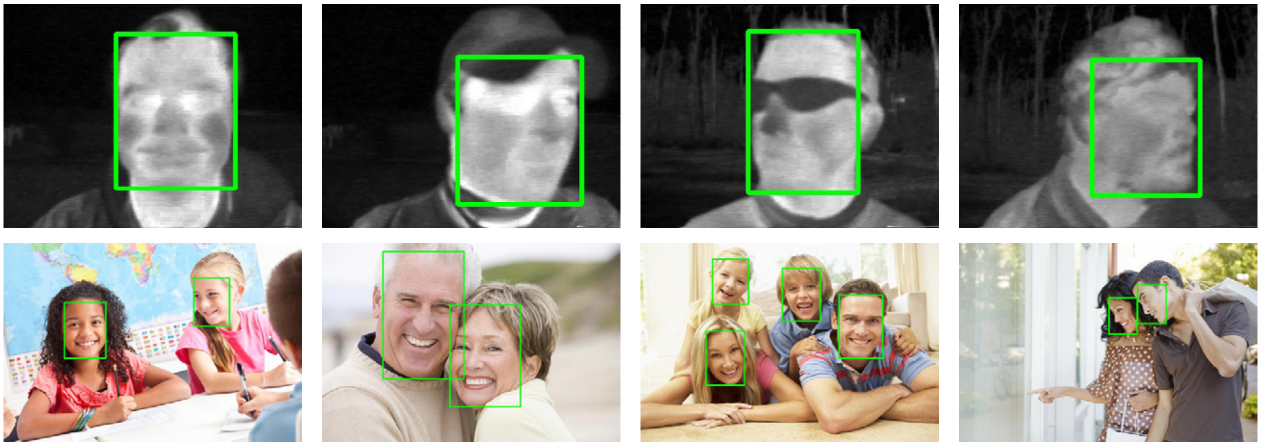


Fig. 4. Training labeled data Samples : (Top) Our manual face annotations regarding thermal LWIR imagery (Terravic DB), (Bottom) Visible face annotations (WIDER DB).

Face detection from visible images has been widely investigated in many scientific researches [48]–[50]. However, only a few works have been dedicated to thermal imagery [51], [52]. According to the best of our knowledge, there is no annotated public thermal face database that allows training a deep network for face detection. To overcome this issue, we have manually annotated the full thermal Terravic Facial IR database in the PASCAL VOC format (see the first row of Fig. 4) which contains a total of 21676 images.

In this paper, a multi-sensor face detection technique, based on the YOLO v3 network architecture, is proposed. It aims to detect faces in both visible and thermal images. For that, we have trained the network in two steps. Firstly, the Terravic Facial IR Database, which we have annotated, is used to train it for face detection in thermal images. Secondly, using a transfer learning technique and the WIDER database [53], a second full training phase has been carried out, for face detection in visible images. This database is fully annotated, including RGB face images with a high degree of variability in scale, pose, occlusion, expression, appearance, and illumination (Fig. 4).

B. Face Synthesis Using Thermal-Visible-Cycle Generative Adversarial Network (TV-CycleGAN)

Recently, Generative Adversarial Networks (GAN) have been proposed as an emerging area in deep learning field, based on two neural networks, a discriminator and a generator respectively. The discriminator acts as a binary classifier that discerns between real and fake images while the generator network learns epoch by epoch to produce images with realistic visual aspect as objective to fool the discriminator. According to application needs, variant GAN architectures have been proposed in the literature such as cGAN [54], Pix2Pix [55] or Cycle GAN [56]. In this work, we are interested in Cycle GAN.

Cycle GAN is based on a pair of vanilla GANs including in total two generators and two discriminators, denoted by (G_T; G_V) and (D_T; D_V) respectively. In order to transform thermal face images t to like visible face images $\hat{v}$, the generator G_T learns the domain translation T (Thermal) to V (Visible) based on the adversarial loss described in equation (1). The negative log-likelihood objective is replaced by a least-squares adversarial loss [57] in order to avoid gradient vanishing problem for training stability and quality generation as mentioned in the original Cycle GAN paper [56], giving the objective loss function described in equation (2). The opposite GAN direction, from domain V (Visible) to T (Thermal), is learned by (G_V ; D_T) according to equation (3).

$$\mathcal{L}_{cyc}(G_V, G_T) = \mathbb{E}_{t\sim p_{data}(t)}\|G_V(G_T(t)) - t\| \\ +\mathbb{E}_{v\sim p_{data}(v)}\|G_T(G_V(v)) - v\| \tag{1}$$

$$\mathcal{L}_{GAN(T\rightarrow V)}(G_T, D_V, T, V) = \mathbb{E}_{v\sim p_{data}(v)}[(D_V(v) - 1)^2] \\ +\mathbb{E}_{t\sim p_{data}(t)}[D_V(G_T(t))^2] \tag{2}$$

$$\mathcal{L}_{GAN(V\rightarrow T)}(G_V, D_T, V, T) = \mathbb{E}_{t\sim p_{data}(t)}[(D_T(t) - 1)^2] \\ +\mathbb{E}_{v\sim p_{data}(v)}[D_T(G_V(v))^2] \tag{3}$$

The main differences between Cycle GAN and Classical GAN are cycle consistency loss Lcyc and identity loss Lid described in equations (4) and (5) respectively. Similar to conventional autoencoders, L_{cyc} corresponds to the L_1 norm of image reconstruction. In case of $T \rightarrow$ V transformation, by injecting a newly generated visible like image $\hat{v}$ from thermal image t using G_T, through the second generator network G_V, leads to reconstruct the original image t, ie: $t \rightarrow G_T(t) \rightarrow G_V(G_T(t))\ \ t$. In order to reduce the translation domain space, the cycle consistency loss is added to the GAN loss function.

The identity loss L_{id} refers to the L_1 norm between the input image and the generated image mapped from its domain, added to preserve the color composition and identity features. The final Cycle GAN loss function is described in equation (6). λ and α are two fixed parameters to control the loss impact on the objective function.

$$\mathcal{L}_{cyc}(G_V, G_T) = \mathbb{E}_{t\sim p_{data}(t)}\|G_V(G_T(t)) - t\| \\ +\mathbb{E}_{v\sim p_{data}(v)}\|G_T(G_V(v)) - v\| \tag{4}$$

$$\mathcal{L}_{id}(G_V, G_T) = \mathbb{E}_{t\sim p_{data}(t)}\|G_T(t) - t\| \\ +\mathbb{E}_{v\sim p_{data}(v)}\|G_V(v) - v\| \tag{5}$$

$$\mathcal{L}_{CycleGAN}(G_V, G_T, D_V, D_T) = \mathcal{L}_{GAN(T\rightarrow V)} + \mathcal{L}_{GAN(V\rightarrow T)} \\ +\lambda\mathcal{L}_{cyc} + \alpha\mathcal{L}_{id} \tag{6}$$

Referring to [14], which the structural similarity index metric (SSIM), defined in formula (7), is used to study the modality gap between visible and infrared sub-bands, we have opted to add the reverse metric (equation 8) to the CycleGAN's loss function to reduce this gap and improve the visual aspect of domain translation from thermal infrared to visible, giving the TV-CycleGAN loss, described in equation (9).

$$SSIM(v,t) = \frac{(2\mu_v\, \mu_t\ + b_1)(2\sigma_{vt} + b2)}{(\mu_v^2 + \mu_t^2 + b_1)(\sigma_v^2 + \sigma_t^2 + b_2)} \tag{7}$$

$$\mathcal{L}_{SSIM(G_V,G_T,v,t)} = (1 - SSIM(G_T(t), v)) \\ +(1 - SSIM(G_V(v), t)) \tag{8}$$

$$\mathcal{L}_{TV-CycleGAN}(G_V, G_T, D_V, D_T) = \\ \mathcal{L}_{CycleGAN}(G_V, G_T, D_V, D_T) + \mathcal{L}_{SSIM} \tag{9}$$

where (μ_v; μt) and (σ_v^2; σ_t^2) denote the mean and the variance, of the respective images v and t. σ_{vt} refers to the cross-covariance. b_1 and b_2 are two constants added to avoid instability when (μ_v^2 + μ_t^2) or ($\sigma_v^2 + \sigma_t^2$) are close to zero.

The proposed TV-CycleGAN pipeline's details are given in Fig. 5.

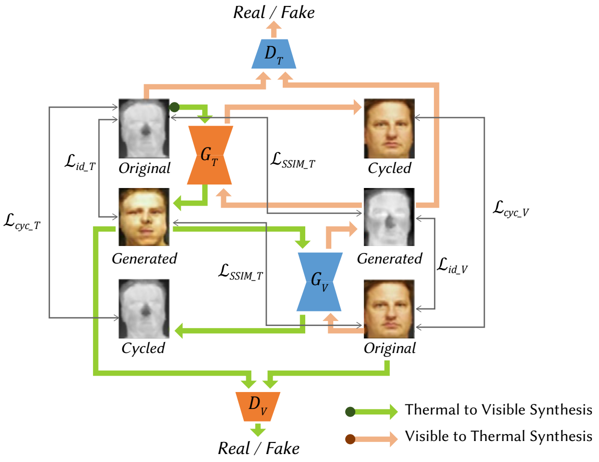


Fig. 5. Flowchart of the proposed TV-CycleGAN, where $L_{cyc} = L_{(cyc_V)} + L_{(cyc_T)}$, $L_{id} = L_{(\,id_V)} + L_{(\,id_T)}$ and $L_{SSIM} = L_{(SSMI_V)}+L_{(SSMI_T)}$.

C. Face Recognition

The proposed TV-CycleGAN is a technique to translate LWIR face images to visible like face images and vice versa, based on a modified adversarial loss. However, the identity information must be preserved to recognize a subject by visible-visible or LWIR-LWIR matching.

For the face recognition task (Fig. 3), we have used two pretrained models on a large-scale face dataset as a feature extraction technique; VGG 16 and Resnet 50. First, the extracted features from the real visible images of the TV-CycleGAN's testing subset, are used to construct the reference embedding. Afterwards, other testing embeddings with the same technique, are formed based on the extracted features from the synthesized images (these synthesis images are obtained from LWIR images using Pix2Pix, TV-GAN, CycleGAN and TV-CycleGAN, respectively).

To evaluate the identity preservation, the nearest neighbour is applied for a fair comparison with the state-of-the-art methods [31], [35], [36], [38]. It is the most commonly used matching process.

It uses the cosine distance, which is calculated for each feature vector in the testing embeddings with those from the reference embedding. Each feature vector from the test set is then classified with the label referred to the lowest distance in the reference embedding and the recognition accuracy is calculated.

The details of each used model are given in the following, and illustrated in the Fig. 6.

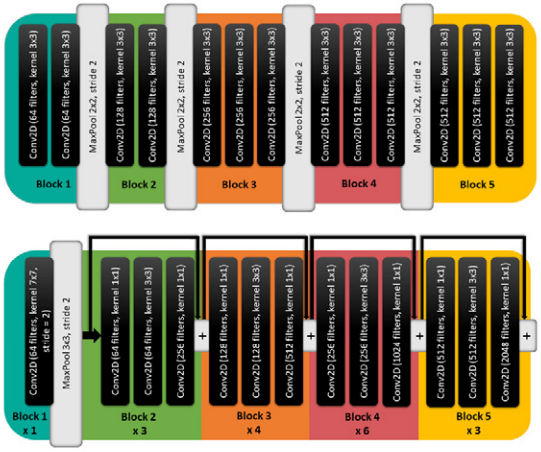

Conv2D (64 filters, kernel 3x3)
Conv2D (64 filters, kernel 3x3)
MaxPool 2x2, stride 2
Conv2D (128 filters, kernel 3x3)
Conv2D (128 filters, kernel 3x3)
MaxPool 2x2, stride 2
Conv2D (256 filters, kernel 3x3)
Conv2D (256 filters, kernel 3x3)
Conv2D (256 filters, kernel 3x3)
MaxPool 2x2, stride 2
Conv2D (512 filters, kernel 3x3)
Conv2D (512 filters, kernel 3x3)
Conv2D (512 filters, kernel 3x3)
MaxPool 2x2, stride 2
Conv2D (512 filters, kernel 3x3)
Conv2D (512 filters, kernel 3x3)
Conv2D (512 filters, kernel 3x3)
Block 1
Block 2
Block 3
Block 4
Block 5
Conv2D (64 filters, kernel 7x7, stride = 2)
MaxPool 3x3, stride 2
Conv2D (64 filters, kernel 1x1)
Conv2D (64 filters, kernel 3x3)
Conv2D (256 filters, kernel 1x1)
+
Conv2D (128 filters, kernel 1x1)
Conv2D (128 filters, kernel 3x3)
Conv2D (512 filters, kernel 1x1)
+
Conv2D (256 filters, kernel 1x1)
Conv2D (256 filters, kernel 3x3)
Conv2D (1024 filters, kernel 1x1)
+
Conv2D (512 filters, kernel 1x1)
Conv2D (512 filters, kernel 3x3)
Conv2D (2048 filters, kernel 1x1)
+
Block 1 x 1
Block 2 x 3
Block 3 x 4
Block 4 x 6
Block 5 x 3

Fig. 6. Details regarding each used pretrained model for the face recognition stage : (Top) VGG16, (Bottom) RESNET 50.

1. VGG16

The VGG16 architecture [58] encompasses 16 deep layers, distributed through five blocks and followed each by a maxpooling layer with a size of 2×2 . The first block comprises two conv layers of 64 filters, and similarly for the second with 128 filters. The following blocks include 3 conv layers each, with 256 filters for the third one and 512 filters for the fourth and fifth blocks. The conv filters use a kernel size of 3×3 with the rectified linear unit (ReLu) activation function. The blocks end with three fully connected layers and a Softmax for prediction probabilities.

For our experiments, the VGG16 model was trained on the VGG Face dataset [59], which includes 2.6 million face images from 2622 subjects. We kept the model up to the fifth block, yielding to extract feature vectors of size 512 attributes.

2. RESNET50

The RESNET 50 belongs to the residual networks [60], including shortcut connections. The network starts with a block of 64 conv filters with kernel size of 7×7 and a maxplooing layer of size of 3×3 with a stride of 2. The rest of the network is distributed through four other blocks of 3 conv layers inter-connected using a shortcut connection each, and kernel sizes of 1, 3 and 1, respectively. The second and fifth blocks are repeated three times while the third, 4 times, and the fourth 6 times, giving in total 50 layers. It is followed by an average pooling layer and fully connected layers of 1000 nodes with a Softmax.

For our experiments, the RESNET 50 model was trained on the VGG Face 2 dataset [61] that comprises 3.31 million face images gathered from 9131 identities. We erased to last layers to extract feature vectors by the last block with 2048 attributes.

IV. Experimental Results

A. Used Material and Tools

All our networks have been implemented using the Keras Python framework. The proposed face detector deep network is trained using the following configuration: learning rate of 0.0001 with Adam optimizer, callbacks such as early stopping and learning rates decay by factors 0.1 for 5 and 2 consecutive unimproved epochs, respectively. A batch size of 4 has been fixed to prevent the used graphical card, which is Nvidia GeForce RTX 2080 GPU with 8GB GDDR6, from overloading.

All our networks have been implemented using the Keras Python framework. The proposed face detector deep network is trained using the following configuration: learning rate of 0.0001 with Adam optimizer, callbacks such as early stopping and learning rates decay by factors 0.1 for 5 and 2 consecutive unimproved epochs, respectively. A batch size of 4 has been fixed to prevent the used graphical card, which is Nvidia GeForce RTX 2080 GPU with 8GB GDDR6, from overloading.

Our TV-CycleGAN adopts the U-Net [62] architecture for its generator, with skipped connections. We have adapted the network to our input images of resolution 128×128 px and used blocks of convolution filters, leaky relu as activation function and instance normalization. For the discriminator network, a fully convolutional neural network using the same blocks as the generator has been used for the binary classification. The TVCycleGAN was trained from scratch for 200 epochs, with the default parameters of the original CycleGAN paper: $\lambda = 10$ and $\alpha = 1$, learning rate of 0.0002 with Adam optimizer and a batch size of 1. Nvidia Tesla K80 was used in this experiment.

B. Used Datasets

Five databases have been used in the experiments that we have carried out; they are presented in the following.

1. Terravic Thermal Database

Terravic Thermal DB[1] provides a set of thermal face images, for 20 different individuals, captured using the Raytheon -3 Thermal-Eye 2000AS thermal sensor, under different scenarios (normal posture, wearing sunglasses/hats, indoor/outdoor and variations in pose). It consists of a total of 21676 images with resolution of 320×240 px. The face images of each subject are put in one folder and split into two subsets: a training set (face folders from 10 to 20), including 16462 images, and a test set of 5197 images (face folders from 01 to 04, and 07 to 09), folders 05 and 06 are not available.

2. WIDER Database

WIDER DB [53] is a public fully annotated database in PASCAL VOC format, it consists of 32203 visible images from 61 different event classes. These images contain a total of 393703 labeled faces including occlusions and several variations in scale and pose. The database is divided into 3 subsets, 40% for training (12880 images), 10% for validation (3226 images) and the remaining 50% for the test set.

3. TUFTS Database

TUFTS DB [63] is one of the largest public[2] databases that provide faces images acquired in different modalities such as visible, thermal, near-infrared, 3D and facial sketch. It consists of over 10000 images, collected from 112 individuals from more than 15 countries, with several age groups. These images include facial expressions and various acquisition angles. In our experiments, we have used the visible and thermal subsets, which include in total 1537 paired images. The thermal images in this database were acquired using the FLIR Vue Pro camera. A new version of this database has been shared in 2020. It includes the same pair images (LWIR-Visible) as the original TUFTS database but in aligned form. In this paper, we have called this database “Aligned TUFTS”.

4. NVIE Database

USTC-NVIE[3] is a multispectral database that comprises spontaneous and posed facial expressions [64], [65]. The posed sub-database consists of over than 7000 face images of 107 different subjects, gathered from the visual and LWIR spectra simultaneously at a distance of 0.75m, using the DZGX25M visible camera and the SAT-HY6850 infrared camera respectively. The images include variations in illuminations and facial expressions.

5. PUJ Database

Pontificia Universidad Javeriana (PUJ) [66] database is a public multispectral face dataset that provides 800 paired and nonaligned images from both visible and LWIR modalities. It was acquired from 40 subjects using the FLIR-T360 according to two illumination protocols (with/out lighting), including variations in pose (front and profile capture) and facial expressions (neutral, surprised, and smiling).

The overview of each used database is given in Table II.

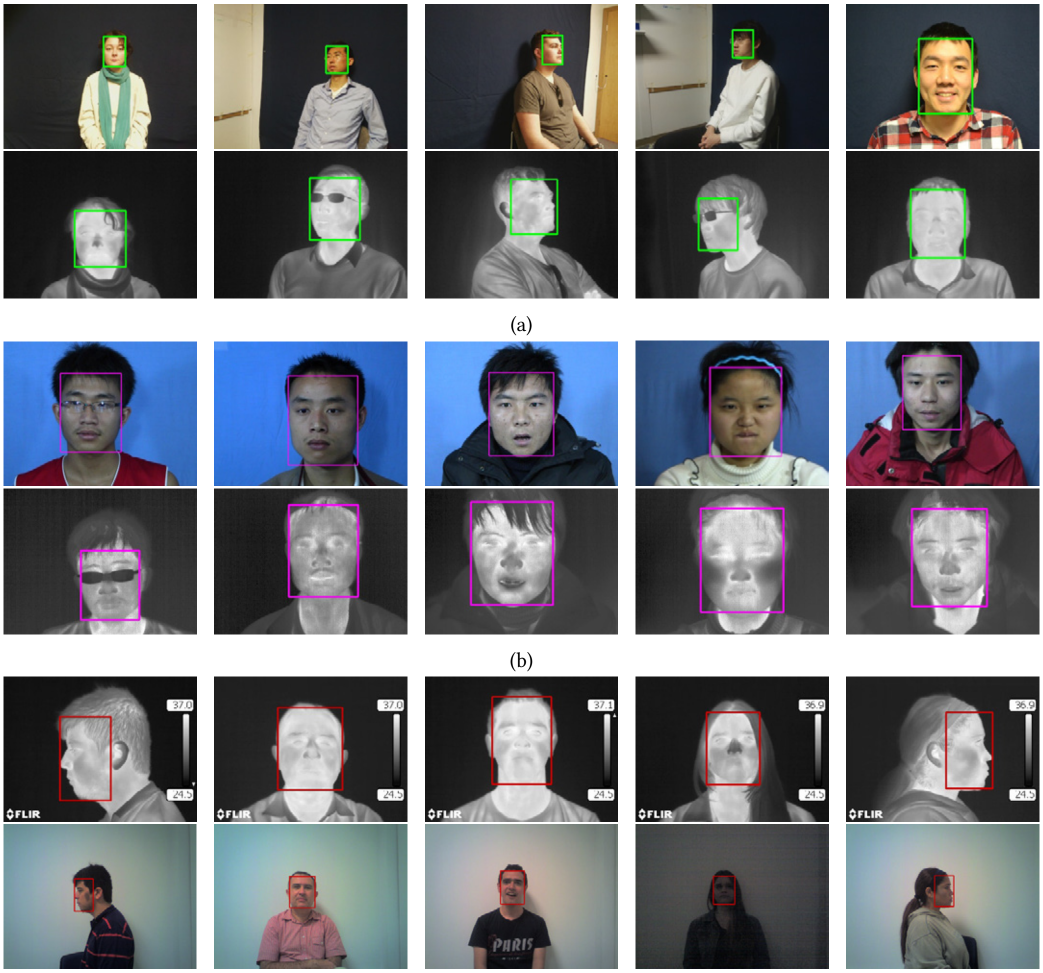
TABLE II. Overview of the Used Face Databases

Database	# Subjects	# Images	VIS	NIR	LWIR
Terravic DB	20	21676	×	×	✓
WIDER DB	-	32203	✓	×	×
TUFTS DB	113	+10000	✓	✓	✓
NVIE DB	107	7329	✓	×	✓
PUJ-T360 DB	40	800	✓	×	✓

[1] http://vcipl-okstate.org/pbvs/bench/Data/04/download.html

[2] http://tdface.ece.tufts.edu/

[3] https://nvie.ustc.edu.cn/



(a)

(b)

(c)

Fig. 7. Results of the proposed multi-sensor face detection technique, applied in different environment scenarios, such as normal posture, presence of facial expressions, slight and extreme angle orientations, and illumination changes, to visible and their corresponding thermal face images, (a) TUFTS database, (b) NVIE database, (c) PUJ database.

C. Protocols

As mentioned beforehand, our new proposed heterogeneous face recognition approach includes three parts: automatic multispectral face detection, visible synthesis using TV-CycleGAN and face recognition (Fig. 3).

Firstly, in the multi-spectral detection step, we have trained a custom YOLO v3 for face detection, using our Terravic face annotations for the thermal modality and the WIDER face annotations for the visible modality. The training process has been carried out in two stages: first, a thermal face detector has been trained using only our Terravic annotations; then, by applying a transfer learning technique, the learned weights have been trained again using the WIDER's face annotations. The proposed multi-spectral face detection technique has been used in the prepossessing stage of our TV-Cycle GAN to automatically crop the region of interest (ROI) for the thermal to visible face translation.

Secondly, in the LWIR to visible translation step using our TVCycle GAN, three multi-spectral databases have been used for evaluation: TUFTS, NVIE and PUJ databases. All thermal/visible face images pairs have been automatically cropped using our own YOLO v3 model and then have been split randomly into two subsets (training/test), according to ratios equal to 95/17 and 92/15, for the TUFTS and NVIE databases, respectively.

Concerning the PUJ database. As it is the smallest one, we have considered the TUFTS database as the training set and we have used all the PUJ images for the test (as probe images). This procedure allows us to test the robustness and the efficiency of the proposed method when the system and acquisition conditions change, since the FLIR Vue Pro has been used for the TUFTS database acquisition where the FLIR T360 has been used for the PUJ database acquisition. Also, we have carried out experiments on Aligned TUFTS database because it is one of the newest available datasets with aligned images, which constitutes an interesting way to show the recognition rate enhancement introduced by our method for aligned faces.

Finally, in the face recognition step, the real-visible face images are enrolled into a pretrained VGG face network, based on the VGG 16 [59] and the RESNET 50 architectures [61], to construct the reference face embedding. Afterwards, in the testing phase, the extracted feature vectors from each probe synthesized-visible image, are compared against those stored in the reference embedding for a face identification related the lowest cosine distance.

Furthermore, we have applied facial landmarks detection on the obtained synthesized images using the proposed TVCycleGAN method. From that, we have easily detected the landmarks on the original thermal ones. Thanks to this result, we have dealt with thermal face landmark detection which is one of the most challenging task.

TABLE III. Quantitative Evaluation of the LWIR to Visible Face Images Synthesis Regarding TUFTS and NVIE Databases

Method	TUFTS					NVIE				
	SSIM	PSNR	MSE	RMSE	MAE	SSIM	PSNR	MSE	RMSE	MAE
Raw Thermal	0.305	9.932	7016.4	82.48	66.7	0.221	10.195	6400.9	79.42	67.28
Pix2Pix	0.332	12.906	3569.6	58.74	44.25	0.274	13.635	2977.6	53.81	42.74
TV-GAN	0.321	12.761	3671.4	59.61	45.22	0.271	13.031	3434.7	57.74	46.34
CycleGAN	0.381	13.902	2978.4	53.03	40.41	0.307	14.254	2510.7	49.75	38.44
TV-CycleGAN (Our)	**0.384**	**13.964**	**2902.3**	**52.50**	**39.97**	**0.324**	**14.28**	**2473.8**	**49.36**	**37.9**

D. Performance Evaluation

1. Face Detection Results

The face detection results obtained by our YOLO v3 based network are illustrated in Fig. 7, using different visible face images and their corresponding LWIR face images. Several acquisition scenarios have been considered to evaluate our multisensors face detector: front pose, slight and extreme orientation, high and low illumination conditions. These results clearly demonstrate the efficacy of the proposed face detector, under the variations cited above.

2. Visible Synthesis Results

To show the performance of our TV-CycleGAN proposed method, for visible face image synthesis, several experiments have been carried out using three multi-spectral face databases, called TUFTS and its aligned version, NVIE and PUJ, respectively. Also, quantitative, qualitative evaluations and comparison to other state-of-the-art methods have been done.

The quantitative evaluation has consisted in computing the Structural Similarity Index (SSIM), the Peak Signal to Noise Ratio (PSNR), the Mean Squared Error (MSE), the Root Mean Squared Error (RMSE) and the Mean Absolute Error (MAE) metrics, for each pair of thermal-visible images and each pair of synthesizedreal visible images, for the TUFTS and NVIE databases. The average value of each metric is reported in Table III.

We notice from the thermal-visible SSIM related to the used databases, that we deal in this paper with the most complicated case, because the LWIR-VIS modality gap corresponds to an SSIM of 0.305 for TUFTS DB and 0.221 for NVIE DB. These gaps are higher than the one reported in the comparative study of [14] where the SSIM is equal to 0.335.

The comparison of different GAN-based methods: Pix2Pix [55], TV-GAN [36], Cycle-GAN [56], and TV-CycleGAN; using the metrics SSIM, PSNR, MSE, RMSE and MAE, shows that the proposed method (TV-CycleGAN) allows the best modality gap reduction for both databases (Table III).

Even if the difference seems small between the results of CycleGAN and TV-CycleGAN (all metrics at Table III), the TVCycleGAN provides a significant improvement on the quality of synthesized faces as shown in Fig. 8 and 9 related to the qualitative evaluation.

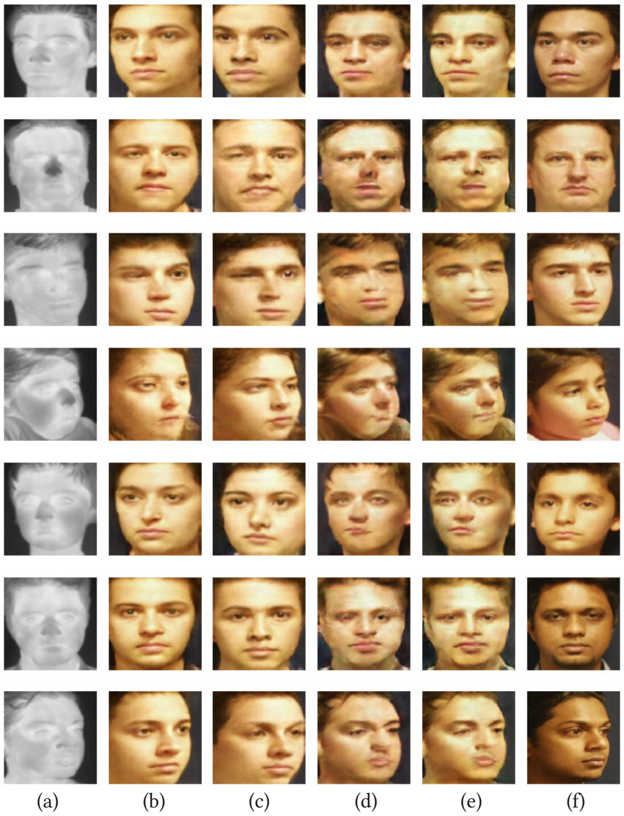


Fig. 8. Qualitative comparative study regarding TUFTS database for the visible face synthesis from LWIR face images: (a) Plain Thermal, (b) Pix2Pix [55], (c) TV-GAN [36], (d) CycleGAN [56], (e) TV-CycleGAN (Ours), (f) Target Visible.

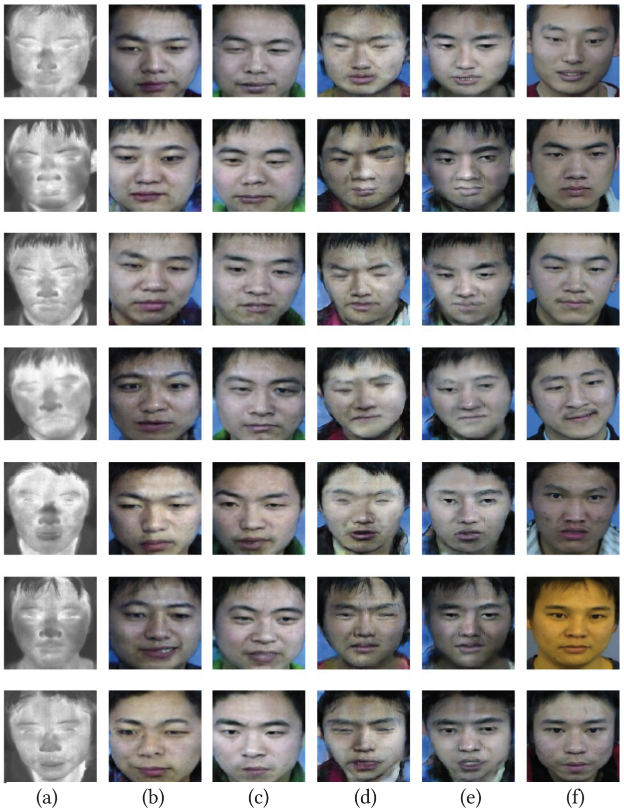


Fig. 9. Qualitative comparative study regarding NVIE database for the visible face synthesis from LWIR face images: (a) Plain Thermal, (b) Pix2Pix [55], (c) TV-GAN [36], (d) CycleGAN [56], (e) TV-CycleGAN (Ours), (f) Target Visible.

The qualitative evaluation is based on visual inspection. We have randomly selected some face image samples to show the results of our proposed visible synthesis method, as shown in Fig. 8 and 9. Even though all synthesizing methods fail sometimes to generate the detailed facial attributes, due to the absence of regularization guiding the GAN training, the proposed one, whose results are shown in the fifth column, outperforms the other state-of-the-art methods and provides a satisfying generation quality. Indeed, the synthesized images shown in this column preserve the persons' identities, they are close to their corresponding ground truth images, shown in column 6.

We can notice from the results related to TUFTS database (Fig. 8), that the TV-CycleGAN, provides a satisfying generation quality, thanks to the loss function similarity incorporated, as it has been mentioned beforehand. Even if the difference seems small between the results of CycleGAN and TV-CycleGAN (Table III), the TV-CycleGAN provides a significant improvement on the quality of synthesized faces. As shown in Fig. 8, 9 and 10 related to the qualitative evaluation, the proposed TV-CycleGAN gives more realistic faces and conserves better than other methods person's identity. For example, in row 2, the TV-CycleGAN is able to better generate nose attributes. In row 4, our method generates the face with better mouth and nose than Cycle-GAN. Also, in row 5, the obtained face from TV-CycleGAN has a better generated mouth.

Similarly, for the NVIE database (Fig. 9), our TV-CycleGAN outperforms the CycleGAN for facial attributes generation in all cases, particularly the eyes, eyebrows, nose, and skin texture, as shown in Fig. 10.

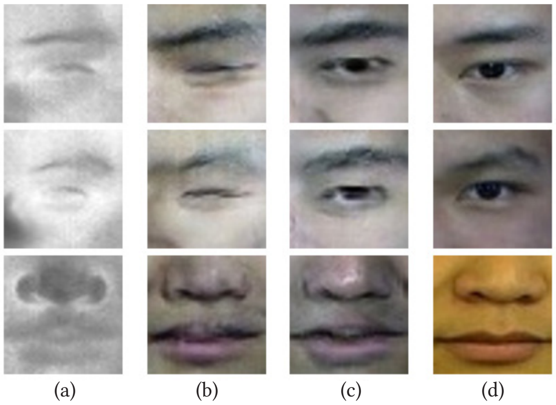


Fig. 10. Enlarged regions of facial attributes, eyes and eyebrows, nose and mouth from Fig. 7 to compare TV-CycleGAN against its main competitor CycleGAN: (a) Plain Thermal. (b) CycleGAN (c) TV-CycleGAN, (d) Target Visible.

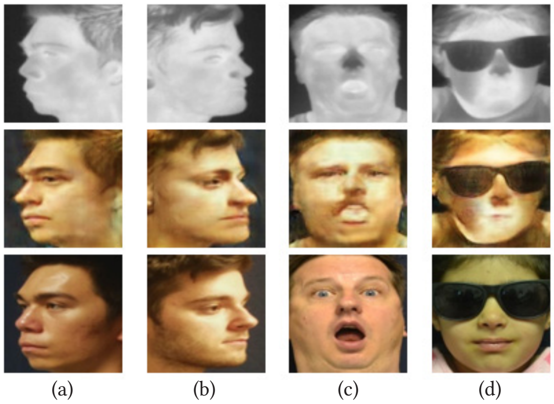


Fig. 11. LWIR to Visible translation using TV-CycleGAN for scenarios including: extreme poses ((a) and (b)), facial expression (c) and glasses (d). first row: Raw Thermal, second row: TV- CycleGAN transformation, third row: Target Visible.

To reinforce the efficacy of the proposed method, we have included extreme poses and facial expressions in our TVCycleGAN training phase (Fig. 11).

In the light of all these results, we can assume that the proposed TV-CycleGAN will have a great impact on the face recognition results as shown in the next section.

3. Face Recognition Results

The proposed TV-CycleGAN's main goal, is reducing the LWIR-Visible modality gap, by synthesizing visible-like face images from thermal ones while preserving as much as possible relevant identity information, to improve the accuracy of heterogeneous face recognition.

In the face recognition step, we have used the pretrained models; VGG16 [59] and the RESNET50 [61] for feature extraction. First, a reference face embedding has been built with the extracted features from the testing real-visible subset images. Afterwards, each synthesized image from LWIR using GANs, is enrolled into one of the pretrained models and then classified with the label of the closest feature vector in the reference embedding, based on the cosine distance. Four metrics have been used to evaluate the face recognition performances as shown in Tables IV and V.

From Table IV, one can figure out that the modality gap has a huge impact on the identification accuracy, since matching directly raw thermal face images with the visible ones, gives a low accuracy, which is 28.99 % in the case of the TUFTS DB and only 17.06 % in the case of the NVIE DB, corresponding to 23.48 % and 8.80 % respectively in terms of F1-Scores, when the model VGG 16 is used.

Synthesizing visible-like images to match the real-visible ones often improves the heterogeneous face recognition performance. The TV-CycleGAN achieved the best recognition rates, which are 57.56 % for the TUFTS DB using the VGG 16 model, and 58.32% for the NVIE DB using the RESNET 50 model. Compared to its main competitor, Cycle GAN, our proposed TV-CycleGAN allows an improvement rate of 0.42% and 1.52% for the TUFTS and NVIE databases respectively. Comparing to a direct LWIR-Visible face matching results, the proposed TV-CycleGAN allows an accuracy enhancement of 10.5 % for TUFTS database, and 29.16% for NVIE database.

This improvement results from the new loss function implemented in the TV-CycleGAN. These results are in good agreement with the quantitative evaluation reported in Table 3; they prove that the modality gap reduction brought by the TVCycleGAN contributes to improve the accuracy of heterogeneous face recognition systems (Tables IV and V). Indeed, compared to the Pix2Pix [55] and TV-GAN [36] methods, the proposed TVCycleGAN has the largest AUC and the lowest EER, for both databases, as shown in Tables 4. In addition, to visualize the contribution of TV-CycleGAN synthesis on the face recognition performances, we have applied the Grad-CAM [67] on the TUFTS database as shown in Fig. 12. This shows that the TV-CycleGAN allows the recognition model to focus more on the facial attributes with larger activated regions (referred to the red regions) compared to the original LWIR face images. The activated regions are closer to the ones activated on the ground truth faces, which demonstrates the accuracy improvements compared to the lowest results obtained with the LWIR to visible face matching.

Furthermore, the obtained results from Table V for the aligned TUFTS database show that an additional face alignment stage contributes to improve the recognition performances. Therefore, the TV-CyleGAN reaches the top accuracy of 63.45 %. This result corresponds to an accuracy enhancement of 18.49% in comparison to a direct Aligned LWIR-Visible face matching. On the other hand, our method outperforms the recent Multiple-APGAN one [38] with an enhancement of 3.92% in terms of EER, even if this last one uses more

TABLE IV. Obtained Face Recognition Results Regarding the TUFTS and NVIE Databases

Method	TUFTS				NVIE			
	Accuracy	F1-Score	AUC	EER	Accuracy	F1-Score	AUC	EER
Raw Thermal – VGG 16	28.99%	23.48%	62.28%	42.63%	17.06%	8.80%	55.57%	46.85%
Raw Thermal – RESNET 50	47.06%	42.99%	71.88%	35.38%	29.16%	27.23%	62.05%	42.73%
Pix2Pix – VGG 16	49.57%	49.69%	73.21%	34.25%	23.97%	16.88%	59.27%	44.56%
Pix2Pix – RESNET 50	39.50%	39.03%	67.86%	38.61%	17.28%	13.05%	55.68%	46.78%
TV-GAN – VGG 16	47.90%	46.15%	72.32%	35.00%	14.25%	9.26%	54.07%	47.74%
TV-GAN – RESNET 50	43.70%	43.46%	70.09%	36.85%	17.28%	14.08%	55.68%	46.78%
CycleGAN – VGG 16	57.14%	55.53%	77.23%	30.57%	46.00%	42.09%	71.07%	35.96%
CycleGAN – RESNET 50	52.94%	54.07%	75.00%	32.65%	56.80%	55.40%	76.86%	30.83%
TV-CycleGAN – VGG 16 (Our)	**57.56%**	55.17%	**77.46%**	**30.36%**	46.65%	45.00%	71.42%	35.68%
TV-CycleGAN – RESNET 50 (Our)	55.88%	**55.79%**	76.56%	31.21%	**58.32%**	**55.60%**	**77.67%**	**30.05%**

TABLE V. Obtained Face Recognition Results Regarding the Aligned TUFTS Version and PUJ Databases

Method	Aligned TUFTS				PUJ			
	Accuracy	F1-Score	AUC	EER	Accuracy	F1-Score	AUC	EER
Raw Thermal – VGG 16	29.41%	22.93%	62.50%	42.48%	14.05%	7.97%	55.92%	46.78%
Raw Thermal – RESNET 50	44.96%	39.43%	70.76%	36.31%	27.27%	24.08%	62.70%	42.56%
CycleGAN – VGG 16	58.40%	55.55%	77.90%	29.93%	25.62%	20.16%	61.86%	43.12%
CycleGAN – RESNET 50	62.60%	61.17%	80.13%	27.69%	42.15%	37.68%	70.33%	36.99%
TV-CycleGAN – VGG 16 (Our)	**63.45%**	**61.51%**	**80.58%**	**27.22%**	34.71%	27.81%	66.52%	39.90%
TV-CycleGAN – RESNET 50 (Our)	63.03%	59.42%	80.35%	27.46%	**42.98%**	**37.78%**	**70.75%**	**36.65%**

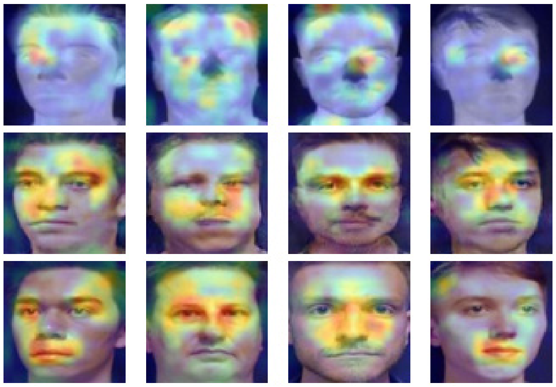


Fig. 12. Grad-CAM heatmaps from the TUFTS database using the pretrained VGG16 model. First row: Raw Thermal, second row: TV-CycleGAN transformation, third row: Ground truth.

losses (5 losses) in its objective function than our method that uses only one additional loss (Eq. 9).

Concerning PUJ database, the proposed TV-CycleGAN method shows its robustness and efficacy when the system and acquisition conditions change. Indeed, it reaches the accuracy rate of 42,98% and contributes to an accuracy enhancement of 15.71% compared to the LWIR to Visible face matching.

4. Thermal Facial Landmark Detection Results

We have also dealt with one of the hot topics in LWIR imagery, which is thermal face landmarks detection [68]–[70]. It is quite hard to directly perform the detection in thermal imagery due to its low contrast. To overcome this issue, our proposed TV-CycleGAN method proves its ability to allow interesting facial landmarks detection on the generated face images. These landmarks could be applied directly on the thermal ones. For that, we have applied the dlib Python library to detect 68 landmarks on the synthesized images, and then we have projected them on the LWIR face images, as shown in Fig. 13. As it can be observed, the obtained results are very satisfactory. There are very promising for many applications such as face tracking and automatic multispectral face alignment.

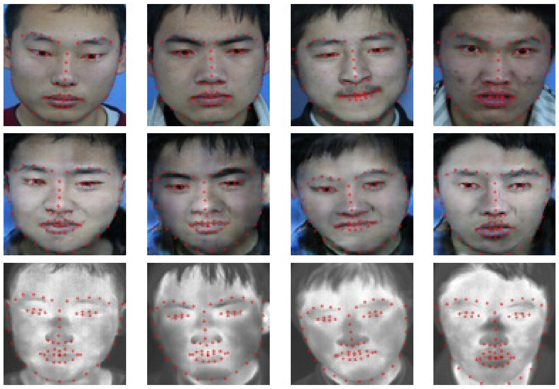


Fig. 13. Thermal facial landmark detection results based on TV- CycleGAN transformation. Top: detection on real visible images, Middle: detection on synthesized visible images from LWIR, Bottom: transferred facial landmarks coordinates on Thermal face images from those detected on synthesized face images (Middle).

5. Complexity and Time Computation Results

To show the computation performance of our proposed heterogeneous face recognition, two hardware configurations have been used. The first configuration consists of Intel i3 5010 as CPU without any dedicated GPU where the second one uses the Intel Xeon CPU with Tesla K80 GPU. We have computed the complexity of each model used in each stage based on the number of floated point operations (FLOPs) and the corresponding average time elapsed for computation. The obtained results are reported in Table VI.

From Table VI, we can figure out that our heterogeneous face recognition system can be adapted easily to video surveillance systems including its three stages, and can run at a frame rate of 5 FPS when a dedicated GPU is used; however, when the hardware configuration has only a CPU for the computation, it takes 2.78 seconds to detect, synthesize and recognize the subject with the VGG16 model that corresponds to 83.65 × 10^9 FLOPs, and 2.49 seconds with the RESNET 50 model, corresponding to 72.15 × 10^9 FLOPs, respectively. This is unsuitable for real-time applications.

TABLE VI. Time Computation Performance of Our Heterogeneous Face Recognition System Regarding Two Hardware Configuration With/Without GPU

Stages	Model	FLOPs	Configuration 1	Configuration 2
Face Detection	YOLO v3	65.86×10^9	1895.72 ± 147.51ms	153.08 ± 27.42ms
Face Synthesis	TV-CycleGAN	2.49×10^9	80.53 ± 5.7ms	7 ± 0.6ms
Face Recognition	VGG16	15.3×10^9	804.66 ± 151.01ms	32.03 ± 3ms
	RESNET50	3.8×10^9	508.94 ± 102.32ms	11.95 ± 0.8ms
Total	with VGG16	83.65×10^9	2780.91 ± 304.22ms	192.11 ± 31.02ms
	with RESNET50	72.15×10^9	2485.19 ± 255.53ms	172.03 ± 28.82ms

V. Conclusion

Cross spectral face recognition is a challenging task due to the large gap between modalities, especially between LWIR and visible spectra. Our contribution in this paper is fourfold. First, we have proposed a thermal-visible face detection method. This method is based on the YOLO v3 architecture and provides an advanced solution for face detection in both thermal and visible imagery, which makes it suitable for several applications, such as facial emotion recognition (FER) or liveness detection. Second, We have annotated a full thermal face database and have shared it with the scientific community in Github repository[4]. Third, we have proposed a modified CycleGAN, called, TVCycleGAN that allows to translate LWIR images to visible-like images. Finally, the synthesized-visible face images obtained by this network are very promising for thermal facial landmark detection. Summing up the results, it can be concluded that the proposed TV-CycleGAN deep learning network shows its robustness and efficacy in LWIR to visible faces synthesis for heterogeneous face recognition. Compared to some recent stateof- the-art methods, the proposed method gives more realistic faces and conserves better persons' identities. Thus, our results could be considered as a major asset for the use of our proposed system in daily real-life scenarios. For the future, we plan to improve the proposed method by guiding the GAN training. We also aim to extend our findings to Thermal-Visible face registration, using the detected face landmarks in synthesized visible images.

References

[1] N. K. Benamara, M. Keche, M. Wellington, Z. Munyaradzi, "Securing E-payment Systems by RFID and Deep Facial Biometry," in *2021 1st International Conference on Artificial Intelligence and Data Analytics (CAIDA)*, Riyadh, Saudi Arabia, Apr. 2021, pp. 151–157.

[2] S. Dargan, M. Kumar, "A comprehensive survey on the biometric recognition systems based on physiological and behavioral modalities," *Expert Systems with Applications*, vol. 143, p. 113114, Apr. 2020.

[3] M. Turk, A. Pentland, "Eigenfaces for Recognition," *Journal of Cognitive Neuroscience*, vol. 3, pp. 71–86, Jan. 1991.

[4] P. Belhumeur, J. Hespanha, D. Kriegman, "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 711–720, July 1997.

[5] L. Wiskott, J.-M. Fellous, N. Kuiger, C. von der Malsburg, "Face recognition by elastic bunch graph matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 775–779, July 1997.

[6] B. Hamdan, K. Mokhtar, "Face recognition using Angular Radial Transform," *Journal of King Saud University - Com- puter and Information Sciences*, vol. 30, pp. 141–151, Apr. 2018.

[7] R. Shoja Ghiass, O. Arandjelović, A. Bendada, X. Maldague, "Infrared face recognition: A comprehensive review of methodologies and databases," *Pattern Recognition*, vol. 47, pp. 2807–2824, Sept. 2014.

[8] Mamta, M. Hanmandlu, "Robust authentication using the unconstrained infrared face images," *Expert Systems with Applications*, vol. 41, pp. 6494–6511, Oct. 2014.

[9] M. Kanti Bhowmik, Kankan, S. Majumder, G. Majumder, A. Saha, A. Nath, D. Bhattacharjee, D. K. Basu, M. Nasipuri, "Thermal Infrared Face Recognition – A Biometric Identi- fication Technique for Robust Security system," in *Reviews, Refinements and New Ideas in Face Recognition*, P. Corcoran Ed., InTech, July 2011.

[10] M. Akhloufi, A. Bendada, J.-C. Batsale, "State of the art in infrared face recognition," *Quantitative InfraRed Thermography Journal*, vol. 5, pp. 3–26, June 2008.

[11] G. Pan, L. Sun, Z. Wu, S. Lao, "Eyeblink-based Anti-Spoofing in Face Recognition from a Generic Webcamera," in *2007 IEEE 11th International Conference on Computer Vision*, Rio de Janeiro, Brazil, 2007, pp. 1–8, IEEE.

[12] S. Jia, G. Guo, Z. Xu, "A survey on 3D mask presentation attack detection and countermeasures," *Pattern Recognition*, vol. 98, p. 107032, Feb. 2020.

[13] B. Hamdan, K. Mokhtar, "A self-immune to 3D masks attacks face recognition system," *Signal, Image and Video Processing*, vol. 12, pp. 1053–1060, Sept. 2018.

[14] S. Hu, N. Short, B. S. Riggan, M. Chasse, M. S. Sarfraz, "Het- erogeneous Face Recognition: Recent Advances in Infrared- to-Visible Matching," in *2017 12th IEEE International Con- ference on Automatic Face & Gesture Recognition (FG 2017)*, Washington, DC, DC, USA, May 2017, pp. 883–890, IEEE.

[15] R. Shoja Ghiass, H. Bendada, X. Maldague, "Université Laval Face Motion and Time-Lapse Video Database (UL-FMTV)," in *Proceedings of the 2018 International Conference on Quantitative InfraRed Thermography*, 2018, QIRT Council.

[16] T. Bourlai, A. Ross, C. Chen, L. Hornak, "A study on using mid-wave infrared images for face recognition," Baltimore, Maryland, May 2012, pp. 83711K–83711K–13.

[17] T. Bourlai Ed., *Face Recognition Across the Imaging Spectrum*. Cham: Springer International Publishing, 2016.

[18] S. G. Kong, J. Heo, F. Boughorbel, Y. Zheng, B. R. Abidi, Koschan, M. Yi, M. A. Abidi, "Multiscale Fusion of Visible and Thermal IR Images for Illumination-Invariant Face Recognition," *International Journal of Computer Vision*, vol. 71, pp. 215–233, Feb. 2007.

[19] D. Bhattacharjee, "Adaptive polar transform and fusion for human face image processing and evaluation," *Human- centric Computing and Information Sciences*, vol. 4, p. 4, Dec. 2014.

[20] A. R. Pal, A. Singha, "A comparative analysis of visual and thermal face image fusion based on different wavelet family," in *2017 International Conference on Innovations in Electronics, Signal Processing and Communication (IESC)*, Shillong, India, Apr. 2017, pp. 213–218, IEEE.

[21] G. Hermosilla, F. Gallardo, G. Farias, C. Martin, "Fusion of Visible and Thermal Descriptors Using Genetic Algo- rithms for Face Recognition Systems," *Sensors*, vol. 15, pp. 17944–17962, July 2015.

[22] N. K. Benamara, E. Zigh, T. Boudghene Stambouli, M. Keche, "Combined and Weighted Features for Robust Multispectral Face Recognition," in *Computational Intelligence and Its Applications*, vol. 522, 2018, pp. 549–560.

[23] N. K. Benamara, E. Zigh, T. B. Stambouli, M. Keche, "Efficient Multispectral Face Recognition using Random Feature Selection and PSO-SVM," in *Proceedings of the 2nd International Conference on Networking, Information Systems & Security - NISS19*, Rabat, Morocco, 2019, pp. 1–6.

[24] K. Guo, S. Wu, Y. Xu, "Face recognition using both visible light image and near-infrared image and a deep network," *CAAI Transactions on Intelligence Technology*, vol. 2, pp. 39–47, Mar. 2017.

[25] D. Lin, X. Tang, "Inter-modality Face Recognition," in *Computer Vision – ECCV 2006*, vol. 3954, A. Leonardis, H. Bischof, A. Pinz Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 13–26.

[4] https://github.com/nkbenamara/Terravic-Facial-IR-Database-Annotations-

[26] D. Yi, R. Liu, R. Chu, Z. Lei, S. Z. Li, “Face Matching Between Near Infrared and Visible Light Images,” in *Advances in Biometrics*, vol. 4642, S.-W. Lee, S. Z. Li Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 523–530.

[27] F. Juefei-Xu, D. K. Pal, M. Savvides, “NIR-VIS heterogeneous face recognition via cross-spectral joint dictionary learning and reconstruction,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Boston, MA, USA, June 2015, pp. 141–150, IEEE.

[28] S. Liu, D. Yi, Z. Lei, S. Z. Li, “Heterogeneous face image matching using multi-scale features,” in *2012 5th IAPR International Conference on Biometrics (ICB)*, New Delhi, India, Mar. 2012, pp. 79–84, IEEE.

[29] L. Huang, J. Lu, Y.-P. Tan, “Learning modality-invariant features for heterogeneous face recognition,” in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, Nov. 2012, pp. 1683–1686.

[30] J. Lezama, Q. Qiu, G. Sapiro, “Not Afraid of the Dark: NIR-VIS Face Recognition via Cross-Spectral Hallucination and Low-Rank Embedding,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, July 2017, pp. 6807–6816, IEEE.

[31] K. Mallat, N. Damer, F. Boutros, A. Kuijper, J.-L. Dugelay, “Cross-spectrum thermal to visible face recognition based on cascaded image synthesis,” in *2019 International Confer- ence on Biometrics (ICB)*, Crete, Greece, June 2019, pp. 1–8, IEEE.

[32] A. Kantarcı, H. K. Ekenel, “Thermal to Visible Face Recogni- tion Using Deep Autoencoders,” *arXiv:2002.04219 [cs, eess]*, Feb. 2020.

[33] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, “Gener- ative Adversarial Networks,” *arXiv:1406.2661 [cs, stat]*, June 2014.

[34] L. Song, M. Zhang, X. Wu, R. He, “Adversarial Discrimina- tive Heterogeneous Face Recognition,” *arXiv:1709.03675 [cs]*, Sept. 2017.

[35] C. Chen, A. Ross, “Matching Thermal to Visible Face Images Using a Semantic-Guided Generative Adversarial Network,” in *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*, Lille, France, May 2019, pp. 1–8, IEEE.

[36] T. Zhang, A. Wiliem, S. Yang, B. Lovell, “TV-GAN: Gener- ative Adversarial Network Based Thermal to Visible Face Recognition,” in *2018 International Conference on Biometrics (ICB)*, Gold Coast, QLD, Feb. 2018, pp. 174–181, IEEE.

[37] W.-T. Chu, P.-S. Huang, “Thermal Face Recognition Based on Multi-scale Image Synthesis,” in *MultiMedia Modeling*, vol. 12572, J. Lokoč, T. Skopal, K. Schoeffmann, V. Mezaris, X. Li, S. Vrochidis, I. Patras Eds., Cham: Springer Interna- tional Publishing, 2021, pp. 99–110.

[38] X. Di, B. S. Riggan, S. Hu, N. J. Short, V. M. Patel, “Multi-Scale Thermal to Visible Face Verification via Attribute Guided Synthesis,” *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 3, pp. 266–280, Apr. 2021.

[39] H. Zhang, V. M. Patel, B. S. Riggan, S. Hu, “Generative adversarial network-based synthesis of visible faces from polarimetrie thermal faces,” in *2017 IEEE International Joint Conference on Biometrics (IJCB)*, Denver, CO, Oct. 2017, pp. 100–107, IEEE.

[40] B. S. Riggan, N. J. Short, S. Hu, “Thermal to Visible Synthesis of Face Images Using Multiple Regions,” in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Lake Tahoe, NV, Mar. 2018, pp. 30–38, IEEE.

[41] Z.-Q. Zhao, P. Zheng, S.-T. Xu, X. Wu, “Object Detection With Deep Learning: A Review,” *IEEE Transactions on Neu- ral Networks and Learning Systems*, vol. 30, pp. 3212–3232, Nov. 2019.

[42] R. Girshick, J. Donahue, T. Darrell, J. Malik, “Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation,” in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, USA, June 2014, pp. 580–587, IEEE.

[43] R. Girshick, “Fast R-CNN,” in *2015 IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, Dec. 2015, pp. 1440–1448, IEEE.

[44] S. Ren, K. He, R. Girshick, J. Sun, “Faster R-CNN: To- wards Real-Time Object Detection with Region Proposal Networks,” *arXiv:1506.01497 [cs]*, Jan. 2016.

[45] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, June 2016, pp. 779–788, IEEE.

[46] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A. C. Berg, “SSD: Single Shot MultiBox Detector,” in *Computer Vision – ECCV 2016*, vol. 9905, B. Leibe, J. Matas, N. Sebe, M. Welling Eds., Cham: Springer International Publishing, 2016, pp. 21–37.

[47] J. Redmon, A. Farhadi, “YOLOv3: An Incremental Improve- ment,” *arXiv:1804.02767 [cs]*, Apr. 2018.

[48] A. Kumar, A. Kaur, M. Kumar, “Face detection techniques: a review,” *Artificial Intelligence Review*, vol. 52, pp. 927–948, Aug. 2019.

[49] H. Jiang, E. Learned-Miller, “Face Detection with the Faster R-CNN,” *arXiv:1606.03473 [cs]*, June 2016.

[50] R. Belaroussi, M. Milgram, “A comparative study on face detection and tracking algorithms,” *Expert Systems with Applications*, vol. 39, pp. 7158–7164, June 2012.

[51] Y. K. Cheong, V. V. Yap, H. Nisar, “A novel face detection algorithm using thermal imaging,” in *2014 IEEE Symposium on Computer Applications and Industrial Electronics (ISCAIE)*, Penang, Malaysia, Apr. 2014, pp. 208–213, IEEE.

[52] C. Ma, N. Trung, H. Uchiyama, H. Nagahara, A. Shimada, R.- Taniguchi, “Adapting Local Features for Face Detection in Thermal Image,” *Sensors*, vol. 17, p. 2741, Nov. 2017.

[53] S. Yang, P. Luo, C. C. Loy, X. Tang, “WIDER FACE: A Face Detection Benchmark,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, June 2016, pp. 5525–5533, IEEE.

[54] M. Mirza, S. Osindero, “Conditional Generative Adversarial Nets,” *arXiv:1411.1784 [cs, stat]*, Nov. 2014.

[55] P. Isola, J.-Y. Zhu, T. Zhou, A. A. Efros, “Image-to- Image Translation with Conditional Adversarial Networks,” *arXiv:1611.07004 [cs]*, Nov. 2018.

[56] J.-Y. Zhu, T. Park, P. Isola, A. A. Efros, “Unpaired Image-to- Image Translation Using Cycle-Consistent Adversarial Net- works,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, Oct. 2017, pp. 2242–2251, IEEE.

[57] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, S. P. Smolley, “Least Squares Generative Adversarial Networks,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, Oct. 2017, pp. 2813–2821, IEEE.

[58] K. Simonyan, A. Zisserman, “Very Deep Convolutional Net- works for Large-Scale Image Recognition,” *arXiv:1409.1556 [cs]*, Apr. 2015.

[59] O. M. Parkhi, A. Vedaldi, A. Zisserman, “Deep Face Recognition,” in *Procedings of the British Machine Vision Conference 2015*, Swansea, 2015, pp. 41.1–41.12, British Machine Vision Association.

[60] K. He, X. Zhang, S. Ren, J. Sun, “Deep Residual Learning for Image Recognition,” *arXiv:1512.03385 [cs]*, Dec. 2015.

[61] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, A. Zisserman, “VGGFace2: A Dataset for Recognising Faces across Pose and Age,” in *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, Xi’an, May 2018, pp. 67–74, IEEE.

[62] O. Ronneberger, P. Fischer, T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, vol. 9351, N. Navab, J. Hornegger, W. M. Wells, F. Frangi Eds., Cham: Springer International Publishing, 2015, pp. 234–241.

[63] K. Panetta, Q. Wan, S. Agaian, S. Rajeev, S. Kamath, R. Ra- jendran, S. P. Rao, A. Kaszowska, H. A. Taylor, A. Samani, X. Yuan, “A Comprehensive Database for Benchmarking Imaging Systems,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, pp. 509–520, Mar. 2020.

[64] S. Wang, Z. Liu, S. Lv, Y. Lv, G. Wu, P. Peng, F. Chen, X. Wang, “A Natural Visible and Infrared Facial Expression Database for Expression Recognition and Emotion Infer- ence,” *IEEE Transactions on Multimedia*, vol. 12, pp. 682–691, Nov. 2010.

[65] S. Wang, Z. Liu, Z. Wang, G. Wu, P. Shen, S. He, X. Wang, “Analyses of a Multimodal Spontaneous Facial Expression Database,” *IEEE Transactions on Affective Computing*, vol. 4, pp. 34–46, Jan. 2013.

[66] R. Pulecio, C. Gerardo, “Face recognition on distorted infrared images augmented by perceptual quiality-aware features,” *S. Z. Li and A. K. Jain, Handbook of Face Recognition. New York, NY: Springer, 2005.*, Dec. 2016.

[67] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, “Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization,” *International Journal of Computer Vision*, vol. 128, pp. 336–359, Feb. 2020.

[68] M. Kopaczka, K. Acar, D. Merhof, “Robust Facial Landmark Detection and Face Tracking in Thermal Infrared Images using Active Appearance

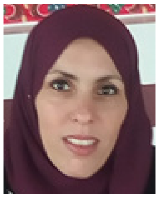
Models:," in *Proceedings of the 11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, Rome, Italy, 2016, pp. 150–158, SCITEPRESS - Science and Technology Publications.

[69] D. Poster, S. Hu, N. Nasrabadi, B. Riggan, "An Examination of Deep-Learning Based Landmark Detection Methods on Thermal Face Imagery," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Long Beach, CA, USA, June 2019, pp. 980–987, IEEE.

[70] W.-T. Chu, Y.-H. Liu, "Thermal Facial Landmark Detection by Deep Multi-Task Learning," in *2019 IEEE 21st Interna- tional Workshop on Multimedia Signal Processing (MMSP)*, Kuala Lumpur, Malaysia, Sept. 2019, pp. 1–6, IEEE.

Nadir Kamel Benamara

He received the "Ingenieur d'Etat" and the "Master" degrees in Advanced Telecom- munications Engineering from the Institut National des Télécommunications et des TIC d'Oran (INTTIC), Algeria in 2016. Currently, he is a PhD Candidate at the Université des Sciences et de la Technologie d'Oran - Mohamed Boudiaf (USTO-MB), Algeria. His research focuses on Biometrics, Computer Vision and Deep Learning.

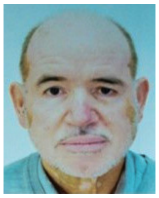


Ehlem Zigh

She is an associate Professor class 'A' at Institut National des Télécommunications et des TIC d'Oran, Algeria. She received her Habilitation degree in Electronic field from Djillali Liabes University, Sidi Bel Abbès, Algeria in 2017. Her Doctorate degree from the University of Sciences and Technologies Mohammed Boudiaf of Oran, Algeria in 2014. She is a Head of a research group at LaRATIC Laboratory at Institut National des Télécommunica- tions et des TIC d'Oran (INTTIC), Algeria. She is a focal point of the Artificial Intelligence Training proposed by AI commons in Algeria. Her research interests include image processing, deep learning, soft computing techniques and internet of things. She has around 20 national and international communications, eleven international publications and she has published a book chapter in a Handbook of Research on Artificial Intelligence Techniques and Algorithms (DOI: 10.4018/978-1-4666-7258-1.ch010), Malaysia. Dr. E. Zigh has been a chairman session at the international SCA 2020 Online Conference. She is member of a technical program committee of EAI innovation research conferences. She is a reviewer at The International Journal of Energy Optimization and Engineering, IGI Global, Algerian journal of research and technology, European journal of remote sensing and Interactive Learning Environment journals.

Tarik Boudghene Stambouli

Doctorat d'Etat in Electronics (Option Signal Processing) from Université des Sciences et de la Technologie d'Oran Mohamed Boudiaf – Department of Electronics, was graduated in Electronics. Lecturer in Department of Electrotech- nics, Member of Laboratoire Signaux et Images in same University, he currently focuses his research on biometrics.



Mokhtar Keche

Mokhtar Keche received the Ingenieur degree in Telecommunications from ENST Paris in 1978, and the Docteur Ingenieur degree and PhD from the University of Rennes in France and the University of Nottingham in U.K in 1982 and 1998, respectively. He is actually a Professor at the University of USTO in Algeria. His research interests are in the areas of Digital Communications, Array Processing, Multitarget Tracking, Road Traffic Estimation, and Biometry.

MDFRCNN: Malware Detection using Faster Region Proposals Convolution Neural Network

Mahendra Deore[1]*, Uday Kulkarni[2]

[1] Department of Computer Engineering, MKSSS's Cummins College of Engineering for Women, Pune-411052 (India)
[2] Department of Computer Science & Engineering, SGGS Institute of Engineering and Technology, Nanded-431606 (India)

Received 31 August 2020 | Accepted 14 June 2021 | Published 30 September 2021

LA UNIVERSIDAD
EN INTERNET

Abstract

Technological advancement of smart devices has opened up a new trend: Internet of Everything (IoE), where all devices are connected to the web. Large scale networking benefits the community by increasing connectivity and giving control of physical devices. On the other hand, there exists an increased '*Threat*' of an '*Attack*'. Attackers are targeting these devices, as it may provide an easier '*backdoor entry to the users*' *network*'. **MALicious softWARE** (MalWare) is a major threat to user security. Fast and accurate detection of malware attacks are the sine qua non of IoE, where large scale networking is involved. The paper proposes use of a visualization technique where the disassembled malware code is converted into gray images, as well as use of Image Similarity based Statistical Parameters (ISSP) such as Normalized Cross correlation (NCC), Average difference (AD), Maximum difference (MaxD), Singular Structural Similarity Index Module (SSIM), Laplacian Mean Square Error (LMSE), MSE and PSNR. A vector consisting of gray image with statistical parameters is trained using a Faster Region proposals Convolution Neural Network (F-RCNN) classifier. The experiment results are promising as the proposed method includes ISSP with F-RCNN training. Overall training time of learning the semantics of higher-level malicious behaviors is less. Identification of malware (testing phase) is also performed in less time. The fusion of image and statistical parameter enhances system performance with greater accuracy. The benchmark database from Microsoft Malware Classification challenge has been used to analyze system performance, which is available on the Kaggle website. An overall average classification accuracy of 98.12% is achieved by the proposed method.

Keywords

Classification, CNN, Dynamic Analysis, Faster RCNN (F-RCNN), Malware, Malware Static.

DOI: 10.9781/ijimai.2021.09.005

I. Introduction

MALWARE is a major menace to Internet security today. There are various distinctive sorts of cyber assaults in the current day digital world. A few of these are very renowned like, phishing sites, botnets, denial of service (DoS), malware assaults and so on. Lately malware attacks are being increasingly propagated because of the huge development of internet and web-based products like IoE. A report from Symantec in 2019 announced a new malware technique i.e., FormJacking (FJ). Cyber attackers inject malware code in web page forms (specifically, payment page forms handled by Payment Processors) to steal sensitive information about the payment cards, names, addresses, phone numbers, etc. These types of attacks are called 'Supply Chain Attacks' (SCA), and are written in JAVA. According to the Symantec report, an average of approximately 4800 sites have been compromised by the FJ code and there is a 78% hike in SCA. '***Jacking***' is popular amongst cyber attackers. There are varieties of jackings viz. cyber, crypto, form, page, Brand, I, Wi, page, thread, mouse, paste, Data, side, Bio, Juice, etc.

* Corresponding author.
E-mail address: mdeore83@gmail.com

Big data Analytics can be defined as the process of lookup, processing, storing enormous data so as to separate important information out of it. With growth in big data analytics, security and protection concerns are additionally amplified.

Big data servers are effectively open to a more extensive population base; consequently, they increase the possibility of malware attacks. To shield users from the hazards of malware, security companies offer a diverse set of antivirus tools. Usually, these tools follow signature-based methodologies. Signature based recognition is inclined to a few difficulties. For example, there has to be a database with patterns of known sets of threats. Also, frequent refresh is required for these signatures in the repositories which requires the intervention of experienced staff in the signature creation process. Thus, antivirus organizations are not able to define, analyze and develop effective signature patterns.

Due to the escalating growth of online transactions, the level and number of cyber-crimes are increasing. In the present situation, where malware assault is massively expanding, it is very troublesome for pattern matching scanners to recognize new variations of existing malicious programs. Therefore, there is a high demand to formulate other strategies to recognize malware.

This requirement asks for a Malware Detection System (MDS) which can detect malware accurately and act fast enough to quarantine the same. MDS is traditionally feature vector based in which crucial characteristics of the malware are extracted and used to identify the same in real time systems. Behavioral based malware detection can broadly be divided into three categories namely, static, dynamic and hybrid. According to Gandotra [1] in the Static Analysis (SA), malware software is analyzed without being executed. SA typically extracts features like operational code (OPCODE) frequency distribution, control flow graph, syntactic library call, byte-sequence n-grams, string signature etc., after unpacking executable in advance. SA protects the Operating System (OS) from malicious damage but it is vulnerable to code obfuscation techniques. In Dynamic Analysis (DA) malware is executed by making use of a controlled environment viz. sandbox, emulator, simulator, virtual machine and then it is analyzed by monitoring tools like Capture Bat, Process monitor (.pmon), poison IVY, etc. Sandbox generates a detailed and extensive report which requires human interpretation and analysis. The analysis process can be automated to a greater extent but with an addition of extensive computational complexities. Thus, it is time consuming [3]. However, both static and dynamic analyses have some limitations and it is difficult to use either static or dynamic analysis for malware detection. The next approach is the combination of both which is called the hybrid approach. It analyzes the signature of malware in the first phase and combines it with behavioral specifications for a complete analysis.

Malware database is huge. Such systems are largely dependent on Machine Learning (ML) algorithms. To figure out malware patterns in the code successfully, effective solution is - Visual Analytic Technique (VAT), where malware patterns are presented as an image. VAT provides summarized pictures of attacks. These images can be trained using ML for analyzing malware patterns. Malware Detection Developer (MDD) focuses on image patterns due to two reasons. The first reason is, even though the malware developers work in the direction of hiding the code, at the same time, while coming up with variants of the malware, they use the same old code. Therefore, the deviation δ between two images of a single malware family is very small. MDD can take advantage of this mind set and use the similarity mining machine learning method to identify the family of malware. The second reason is an image classification technique which is more mature and faster [4] [5].

This paper presents literature survey in Section III and the Key extract is malware can be packed using different packing methods and with different resolutions. Therefore, for VTA (image) based analysis, research options are still open for providing an improved solution to classify malware. Keeping this as a base, this paper proposes extracting ISSP features for all the malware families taking into consideration all the variants in the binary file. Finally, ISSP features and malware images are trained using fast and robust F-RCNN classifier.

The rest of the paper is organized as follows: related work is presented in Section II. The proposed work model is presented in Section IV. Feature vector formation using perceptual features and the mathematical model of the F-RCNN classifier is described in sections V and VI. Computation of the statistical parameters is described in Section VII. Description of the database is given in Section VIII. Experimental setup and related results are presented in Section IX. Section X discusses the performance analysis of the proposed method. Section XI concludes the paper.

II. Related Work

This section focuses on varieties of features and classification techniques explored by researchers. Related work can be bifurcated in two categories i.e. the work based on image representation and methods other than image representation. As paper proposes image representation of malware, work related to this method is explained first.

A. Image Based Methods

Image is a 2D representation. Key points of 2D representations are as follows

1. Data dimension does not affect processing once similarity space is formed.
2. Equally important clusters are formed.
3. Similar clusters are displayed adjacent for clear visualization [27][28][29].

Malware analysis using Visualization Technique (VT) was proposed by Yoo [16]. He classified images using Self-Organizing Map. VT is mostly used for document and image analysis where files are huge and data is massive. Therefore, it has wide applications in computer security, as malware attacks are in the thousands at a time. Shiravi [35] and N. Diakopoulos [34] identified Brute Force attack on Secure Shell (SSH) by representing details of Internet Protocol (IP) address, UserIDs and various anomalies using varieties of colors. VT was used to display large network packets and helped security analysts to find similarities by checking minute details using a zoom option. S.Foresti [39] and M. Wagner [40] demonstrated usage of VT to represent information like time ('when'), IP address ('Where'), Data ('what') and estimated distances to other hosts.

Quist [17] proposed the use of an Ether hypervisor framework to track and visually represent execution of malware programs. The dynamic analysis framework was named as VERA. Trinius [18] introduced a new concept i.e., Malware Instruction SeT (MIST) for monitoring malwares. They used a CW Sandbox to collect information regarding API calls and performed action. They visually represented distance matrices of features for five malwares.

To improve malware detection, different sections of the binary executable are now represented as gray scale images. These images provide detailed structure of malware, to the extent that they show even small changes in the code, without altering the remaining code structure. Gray scale texture helps in identifying similar patterns of the binary code [41]. L. Nataraj [42] proposed GIST descriptors to classify obfuscated malware.

The Function Length Frequency (FLF) algorithm proposed by Tian [9] was used to detect Trojan after surveying varieties of techniques. Zolkipli [10] suggested the use of Variable Length Instruction Sequences (VLIS) with machine learning algorithms. Shankarapani [11] proposed two models, namely, Malware Examiner using Disassembled Code (MEDiC) and Static Analyzer for Vicious Executables (SAVE). Results were promising as the model had better detection even if malware is obfuscated. Nataraj [19] presented malware binaries as gray scale images. Using a KNN classifier they achieved good average accuracy along with increased speed of malware detection. On a similar line Kancherla [20] used byte plot (image of executables) and achieved 95% accuracy using the SVM classifier.

Kong and Yan [12] used hex dump n-gram, disassembly code, PE header and selected features using the L1 regularized method. They applied different classifiers viz. NB, SVM, K means Neural Network (KNN), decision tree and analyzed the performance of all the features. They concluded that the PE header feature is more prominent in malware detection. Santos et al., 2013b [13] tried to figure out the relevance of each OPCODE and calculated the frequency of OPCODE sequences. They verified the performance using the same four classifiers used by previous researchers. They stated empirically that the model can detect unknown malware as well.

Similarity is calculated based on the distance between each and every pair of points. Minimum distance represents maximum similarity [30] [31]. Projection and semantic orientation are 2D VTs, normally used to check similarity patterns [26] [32] [33].Frequency domain-based feature extraction i.e., Wavelet transforms, was proposed by Gu [14]. Li and Li [15] proposed static features viz. API, classes, functions and packages detecting malware for android APKs. They used different layers of 'Characteristic Tree' containing information of API calls. The method can classify unknown APKs. Han [21] converted a Windows PE binary file into a gray scale image. After that, using an Entropy Graph Generator (EGG), they calculated the entropy of each and every line of an image. Malware detections were done based on similarities of the original binary file.

Arefkhaniet [22] introduced a Local Sensitive Hashing technique specific to image processing, to classify similar input (malware) with high probability. Wu [23] converted disassembled binary executable into opcode sequences into an image. They used PCA to reduce the dimensionality and KNN classifier. Rezaei [24] proposed a similarity measurement algorithm for malware detection which compares the opcode strings of malicious files to improve the detection rate and speed. Venkatraman [36], Zhang [37] and Wylie Shanks [38] explored usage of VT for analyzing malware attack chronology and demonstrated successful system connections with the help of colors.

B. Other Methods

Santos et al., 2013b [13] introduced an OPCODE Executable trace Malware (OPEM) framework. It is a hybrid of statistically obtained frequency of occurrence of OPCODE and dynamically obtained executable traces. Performance was evaluated using Baysine, Decision Tree, SVM and KNN classifiers. Kolosnjaji [25] proposed feature fusion of headers of PE files and convolution of n-grams of an instruction. They achieved a 93% recall rate and accuracy using SVM and ANN (Feed forward) classifier. Ripper Cohen [6] proposed the Repeated Incremental Pruning to Produce Error Reduction (RIPPERk) algorithm that supersedes the learning algorithm, Incremental Reduced Error Pruning (IREP). In RIPPERk, k represents the number of multiple optimization iterations. Schultz [7] made use of three static features (byte sequence, strings and Portable Executable (PE)) and for the first-time used the data mining concept. Kolter [8] proposed a combination of n-gram (instead of non-overlapping byte sequence) features with classifiers viz. Naive-Bayes (NB), Support Vector Machine (SVM), Decision Tree and their boosted versions. They found that the boosted decision tree provides better results.

The following section illustrates an extensive and organized literature summary of innovative techniques proposed by researchers and challenges from state of the art.

III. Literature SURVEY – FORMULATING Problem Statement

A research problem should represent the core subject matter and it should be a discovery of new knowledge. This objective not only asks for rigorous literature survey, but also demands interpretation of the surveyed information to achieve a proper research path. Graphical presentation is given by the author which makes it more suitable to extract the required information.

Fig. 1 provides information about three basic analysis techniques like SA, DA and hybrid, explored by researchers. The SA technique is still preferred by most of the researchers [50]. The hybrid approach is not still popular amongst researchers.

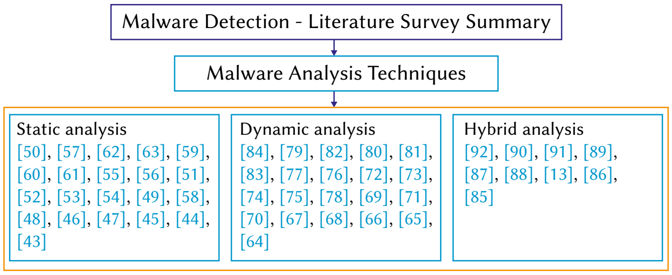


Fig. 1. Survey depicts the type of analysis technique explored by researchers.

Feature extractions and classification techniques are the two pillars of MDS. Fig. 2 is devoted to different feature vectors used by researchers. API calls, system calls, n-gram and OPCODE are still features that are mostly used in MDS. Researchers typically have two paths: the first one is to optimize the feature vector and get a significant limited feature set, and the second one is the selection of a prominent limited set of features manually. Work implemented by researchers in either way is unique itself and uses varieties of byte and hex related features.

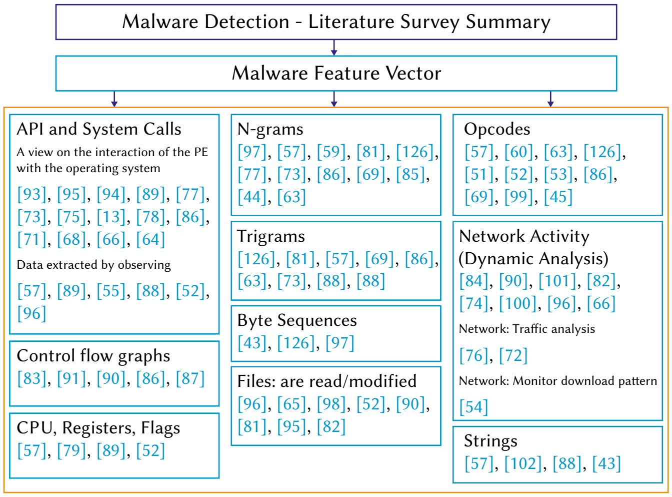


Fig. 2. Survey depicts varieties of feature used by researchers.

A feature vector extracted from malware code should be trained with the help of neural networks. After training, system performance is tested by applying real time malware data. Researchers developed different techniques to extract the feature set. A breakthrough from the survey reveals that researchers using available crawlers, filters etc., end up with a *plethora of* feature vector space which may be redundant. Many times, some features may worsen the accuracy of the system. Thus, feature selection to improve system performance is mandatory but the same should be done without compromising accuracy.

The role of a classifier is important as it defines accuracy and precision. Fig. 3 presents three basic learning process categories i.e., supervised, unsupervised and semi-supervised. Supervised learning is the first choice of researchers where malware annotations are given for training the network.

Fig. 4 presents the wide usage of deep learning techniques like CNN, Deep Neural Network (DNN), Recurrent Neural network (RNN), auto encoders etc. in malware detection. These techniques are well established and provide high performance. CNN and its extensions are preferred for image-based analysis therefore, it is mostly used by researchers. CNN variants viz. Region based CNN (R-CNN), Fast R-CNN, Faster R-CNN are mostly used in image analysis for detecting objects [182].

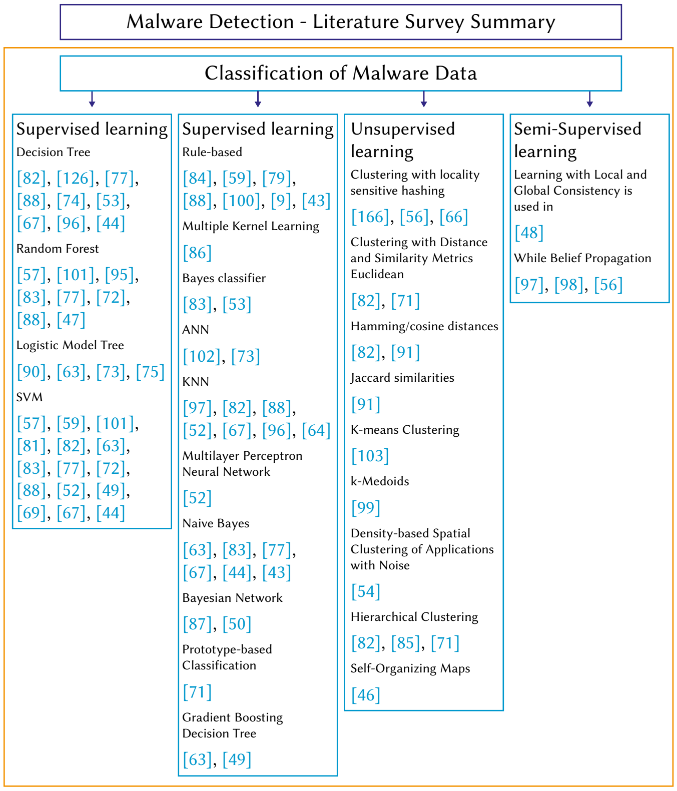


Fig.3. Survey depicts different classifiers explored by researchers.

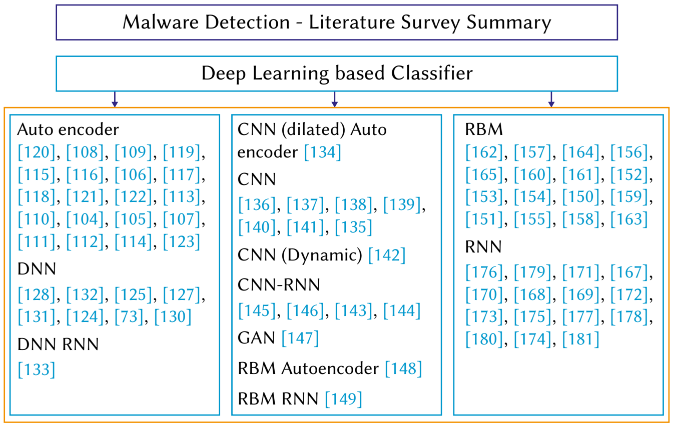


Fig. 4. Literature survey based on Deep Learning Classifier.

These techniques are specific to image analysis and provide less training and testing times, which is the need of a malware detection technique, as it has to run in real time and detect malware as fast as possible. This motivates the author to select FR-CNN. It is a technique with a significantly low computational cost and the same has not been investigated for malware detection. This technique achieves more precision and a faster response.

IV. Proposed Model of MDS

The paper proposes MDS architecture comprising of deep learning network to accurately detect and classify malware families using an image-based technique which is described in Fig. 1. The Benchmark database from Kaggle is used to evaluate the performance of a system. The features vector of malware families is presented as gray scale images. These images will be trained using deep learning and facilitate adaptive learning in real time environment to achieve high accuracy.

The Main contribution of this research work is as follows:

1. *Consideration of prominent static features e.g., string signature, byte-sequence, N-grams, OPCODE*
2. *Represent feature vector as a gray scale image reflecting the malware family behavior*
3. *Arrange feature vector sub parts as varieties of 'Regions' of an image*
4. *NOTE: 'Region' is a generic term. Rectangular regions are considered in this paper, as it is common.*
5. *Minimize the training set*
6. *Compute statistical parameters of the image generated in point-2*
7. *Apply the F-RCNN classifier to a matrix having statistical parameters as well as an image.*

To the best of our knowledge, the above combination i.e., a matrix of static features and an image with F-RCNN classifier has not been evaluated by researchers.

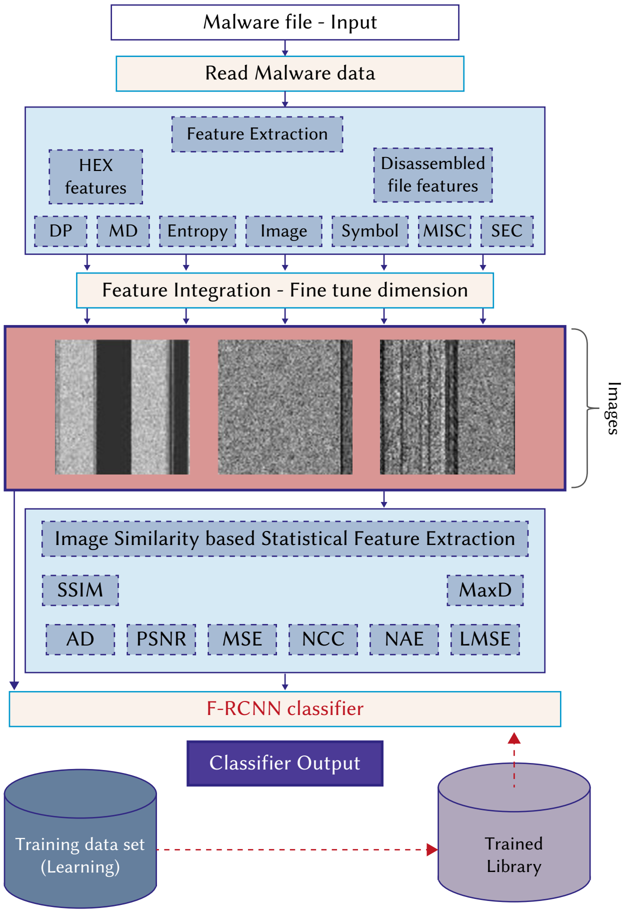


Fig. 5. System Architecture of the proposed MDF-RCNN.

Fig. 5 depicts the system architecture of the proposed MDS. Feature extraction and classification of the malware input file are two major modules of MDS. Each one is described as follows.

V. Feature Extraction Module

A. Features Based on HEX and Disassembled Files

There are basically two major types of features extracted for MDS. These are HEX dump-based features (n-gram, MD1, entropy, image 1(haralick) and Image 2 (lbpfeatures)) and disassembled file features (meta-data, SYM, OPC, MISC, DP and SEC.)

1. Hex Based Features

a) N-gram

An N-gram is a contiguous sequence of n items from a given sequence. The technique is used intensively for characterizing sequences. The sequences may be from a speech or text. Malware samples may be viewed as sequences of HEX value. An N-gram analysis of these HEX values may provide valuable information. The malware sample consists of special symbols viz.??, indicating that the data of that location is not initialized. It also contains byte sequence whose value may range from 0 to 255.

b) Entropy

Entropy (ENT) is basically a measure of randomness or uncertainty or maybe the O amount of disorder. Obfuscation presence can be detected by entropy [183] [184]. In MDS, entropy is calculated based on byte representation. It measures the disarray of the distribution of the bytes in malware code by setting *'order'* and *'randomness factor'*. A sliding window is applied on the malware code and entropy is calculated for each windowed segment. It is represented by:

$$E = e_i, Number\ of\ windowed\ segment\ i = 1, \dots \dots, N \tag{1}$$

The Shannon's formula,

$$e_i = -\sum_{k=1}^{m} f(k) * \log_2 f(k) \tag{2}$$

$f(k) =\ frequency\ \ of\ byte\ k\ with\ window\ segment.$
$m = number\ of\ distinct\ bytes\ in\ the\ window\ segment.$

2. Features Extracted From Disassembled Files

Malware executables must first be disassembled to extract features. The PE format is used by Windows OS (WOS). It's basically a data structure encapsulating important information which will be utilized by the WOS loader to manage the wrapped executable code shown in Fig. 6. The PE file contains *Headers* viz. DOS header, Section table, optional header, PE header, and *Sections* viz. code, imports, data. The Dynamic linker uses this information to map the file into memory. The PE information is important as its basic structure manages memory protection based on the code or data region.

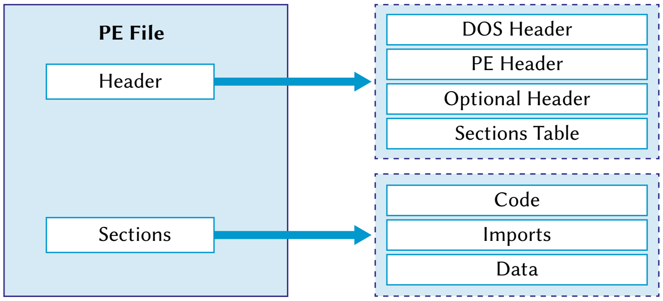


Fig. 6. PE File Structure.

a) Section (SEC)

The '*sections*' consist of code, data and import sections. Further classification includes .data, .idata, .edata, .rdata, .text, .bss, .rsrc, .relocand, .tls. A malware code uses packing techniques where it modifies default sections and may create new sections, to evade Metadata (MD) shown in Fig. 6. An MD program can generate a feature vector by detecting different characteristics of SEC.

b) Metadata

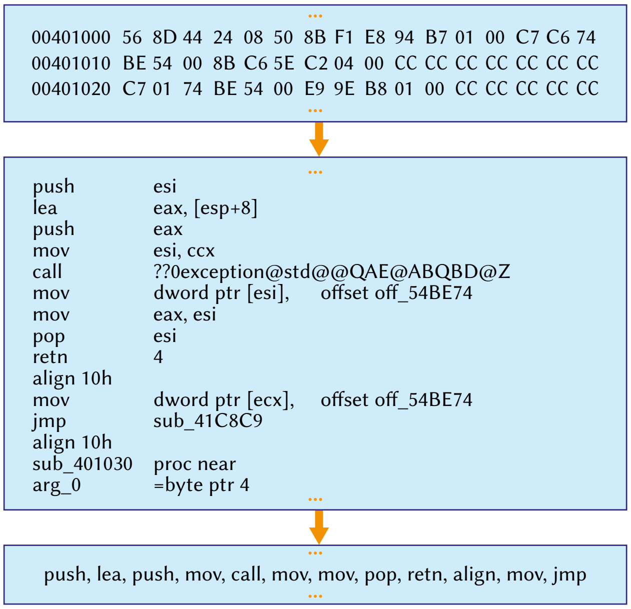
After disassembling, two features viz. the number of lines in the file and file size, are computed and included within the Metadata category (MD2).

c) Symbol:

In malware sample code a set of symbols like [, -, +,], @, ?etc. may be present. Actually, these may correspond to indirect calls or Dynamic Library Loading (DLL), in malware code. In an indirect call, the address of the subroutine is loaded from memory or register. According to [188], the function call depends upon the architecture as well as the optimal decision of a compiler; therefore, such indirect calls may reveal information about data obfuscation. DLL is loaded during runtime by the executable code and executes library functions based on their address. Static analyzers cannot capture such run time events. Therefore, these garbled characters are used specifically by malware developers to evade MD.

d) Operation Code (OPCODE)

OPCODE or mnemonic is a digit and denotes assembly code. The micro-processor executes the OPCODE; therefore, it plays a very important role as it describes the behavioral characteristics of malware as shown in Fig. 7. Machines are x86 based. The instruction set list is complex and large, therefore [189], selected 93 OPCODEs based on frequent use in the malicious application and its commonness. The OPCODE frequency is calculated from the malware code. According to [190], use of Instruction Replacement Technique (IRT) may evade detection. Santos [53] used only OPCODE based features to generate the feature vector and detected malware just by one single class with reasonable amount of accuracy. Thus, it proves that OPCODE based features can contribute more for MD. Researchers also suggested an OPCODE, n-gram based method for MD. The detection is based on the OPCODE frequency feature, calculated by Term Frequency–Inverse Document Frequency (TF-IDF) statistical technique. The OPCODE sequence given vector was used to train the SVM classifier.



```
...
00401000  56  8D 44  24  08  50  8B  F1  E8  94  B7  01  00  C7  C6  74
00401010  BE  54  00  8B  C6  5E  C2  04  00  CC  CC  CC  CC  CC  CC  CC
00401020  C7  01  74  BE  54  00  E9  9E  B8  01  00  CC  CC  CC  CC  CC
...
```

```
...
push           esi
lea            eax, [esp+8]
push           eax
mov            esi, ccx
call           ??0exception@std@@QAE@ABQBD@Z
mov            dword ptr [esi],     offset off_54BE74
mov            eax, esi
pop            esi
retn           4
align 10h
mov            dword ptr [ecx],     offset off_54BE74
jmp            sub_41C8C9
align 10h
sub_401030     proc near
arg_0          =byte ptr 4
...
```

```
...
push, lea, push, mov, call, mov, mov, pop, retn, align, mov, jmp
...
```

Fig. 7. Disassembly File Structure.

e) Register (REG)

The microprocessor has an internal register set which may be used for a specific task. According to [191], in some situations, registers are renamed to make the MD process more complicated and confusing. This asks for keeping track of REG used and frequency of usage of those registers. This feature is useful and helps in detecting a family of malware.

f) Data Define (DP)

Few malware programs use a packing technique and therefore do not use API calls. Instead they contain a few OPCODEs. Typically, they use data related assembler directives like Define Byte (db), Define Word (dw) and Define Double Word (dd). This feature is key for classifying varieties of malware families.

g) Miscellaneous (MISC)

This feature should be selected manually by identifying keywords from the disassembled code. The Interactive Disassembler (**IDA**) tool may be used for the same. Features extracted will be like the number of imported DLLs, identifying strings viz. **hkey_local_machine**(it specifies access to specific paths of the Windows registry), number of blocks in the PE etc. Thus, it depends upon the experience of the MD software developer engineer.

After extracting these features gray scale image is prepared with the help of feature coefficients. Refer Fig. 8 for the same. Typically, these images of a single family should show similarity as the changes in the malware code is not drastic. Consider this point as a base author motivated to compute image similarity based statistical parameters. The next section describes the same.

B. Features - Image Similarity Based Statistical Parameters

This feature set focuses on similarity between two images. A similarity parameter matrix will be computed based on malware images from the '*x*' family compared with itself as well as the remaining families. Reference image (R_I) and Input image (I_I) are two input images. Suppose R_I is from the '*x*' family then I_I will be the remaining images from the '*x*' family and images from the other families. R_I will be constant throughout the process of computing the similarity parameters. As the number of images per family are in the thousands their mean value will be calculated.

The NCC method is used for template matching which is a process used for finding incidences of a pattern or object within an image. Eq. (1), is used to calculate NCC.

$$NCC(R_I, I_I) = C_{R_I I_I}(\widehat{R_I}, \widehat{I_I}) = \sum_{[m,n]\in R} \widehat{R_I}(m,n)\widehat{I_I}(m,n) \quad (3)$$

$$Where,\ \widehat{R_I} = \frac{R_I - \overline{R_I}}{\sqrt{\sum(R_I - \overline{R_I})^2}},\ \widehat{I_I} = \frac{(I_I - \overline{I_I})}{\sqrt{\sum(I_I - \overline{I_I})^2}}$$

AD provides the average of change concerning the input image and the reference image. AD can be expressed as follows:

$$AD(R_I, I_I) = \frac{1}{mn}\sum_{m=1}^{M}\sum_{n=1}^{N}[R_I(m,n) - I_I(m.n)] \quad (4)$$

MaxD provides the maximum of the error signal (i.e., the difference between the processed and reference image). MD is defined as follows:

$$MaxD(R_I, I_I) = \max\{[R_I(m,n) - I_I(m.n)]\} \quad (5)$$

SSIM is based on three factors i.e., luminance, contrast, and structure to better suit the workings of the human visual system. It is a perceptual metric that quantifies image quality degradation. This parameter is selected as the malware developer makes changes in the old code and comes up with the modified code. The modified code can be thought of as the '*Noise*' element in an image. SSIM is defined as follows:

$$SSIM(R_I, I_I) = [l(R_I, I_I)^{\alpha} .\ c(R_I, I_I)^{\beta} .s(R_I, I_I)^{\gamma}] \quad (6)$$

where l = luminance, c = contrast, s = structure

The Laplacian error map shows spatial error distribution across an image. The overall image quality is given by LMSE as follows:

$$LMSE(R_I, I_I) = \frac{\sum_{m=1}^{M}\sum_{n=1}^{N}[L(R_I(m,n)) - L(I_I(m.n)\ )]^2}{\sum_{m=1}^{M}\sum_{n=1}^{N}[L(R_I(m,n))]^2} \quad (7)$$

where L((m,n)) is the Laplacian operator

NAE measures the numerical variance between the R_I and I_I. Moreover, the results that are near to zero means that the image has a high similarity to the original one and the results near the value one indicate that the image has a very poor quality. NAE is calculated as follows:

$$NAE(R_I, I_I) = \frac{\sum_{m=1}^{M}\sum_{n=1}^{N}|R_I(m,n) - I_I(m.n)|}{\sum_{m=1}^{M}\sum_{n=1}^{N}[R_I(m,n)]} \quad (8)$$

MSE and PSNR are used to compare the quality of the image compression. MSE represents the cumulative squared error between the R_I and I_I, whereas the PSNR represents a measure of the peak error. The lower the value of the MSE, the lower the error.

$$MSE(R_I, I_I) = \frac{1}{mn}\sum_{m=1}^{M}\sum_{n=1}^{N}[L(R_I(m,n)) - L(I_I(m.n)\ )]^2$$

$$PSNR = 10 * \log_{10}\frac{255^2}{MSE} \quad (9)$$

After computing ISSP one more feature vector is produced which will be used to train classifier.

VI. Classifier – F-RCNN

This section describes the reason for selecting the classifier i.e., F-RCNN. In computer vision (CV) object or region detection is a major task. Ross [182] proposed a selective search method to extract *N* number of limited regions from an image. These regions are referred to as *Region Proposals*. Associated training network is referred as *Region Proposal Network* (RPN). The algorithm overrides the problem of selecting a huge number of regions. Region based CNN (R-CNN) is thus a fusion of the *Region Proposals* algorithm with CNN. As the first step, this algorithm selects some proposed regions from the image, puts Bounding Boxes (BB) and labels their categories. The Deep learning algorithm (CNN) extracts varieties of features using forward computation from these proposed regions and then trains the network to classify the categories and BB. Following section describes methodology to represent feature coefficients as an image.

A. Image Representation

Malware samples are represented as an image, where each byte of the malware code corresponds to one pixel of gray scale image [185]. The author proposes use of all the features extracted viz. n-gram, MD1, MD2 etc., to construct an image as shown in Fig. 8. The image is formed in such a way that there are ***'regions'*** of feature vectors. ***Each feature vector (N-gram, MD1, entropy, MD2 etc.) may be viewed as a region of an image. Thus, fundamentally the R-CNN algorithm is more suitable for MDS. R-CNN*** will be more effective and efficient for classifying malware.

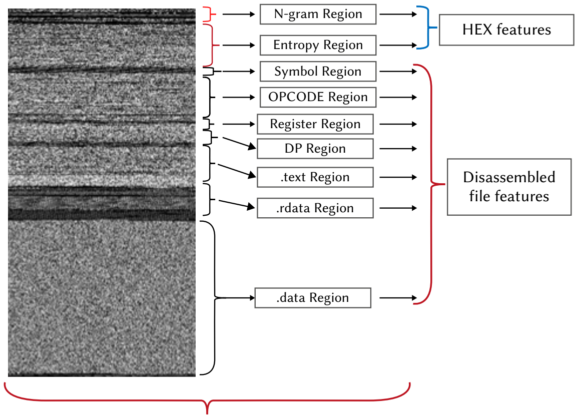


Fig. 8. Gray scale image of malicious code with feature coefficient Regions.

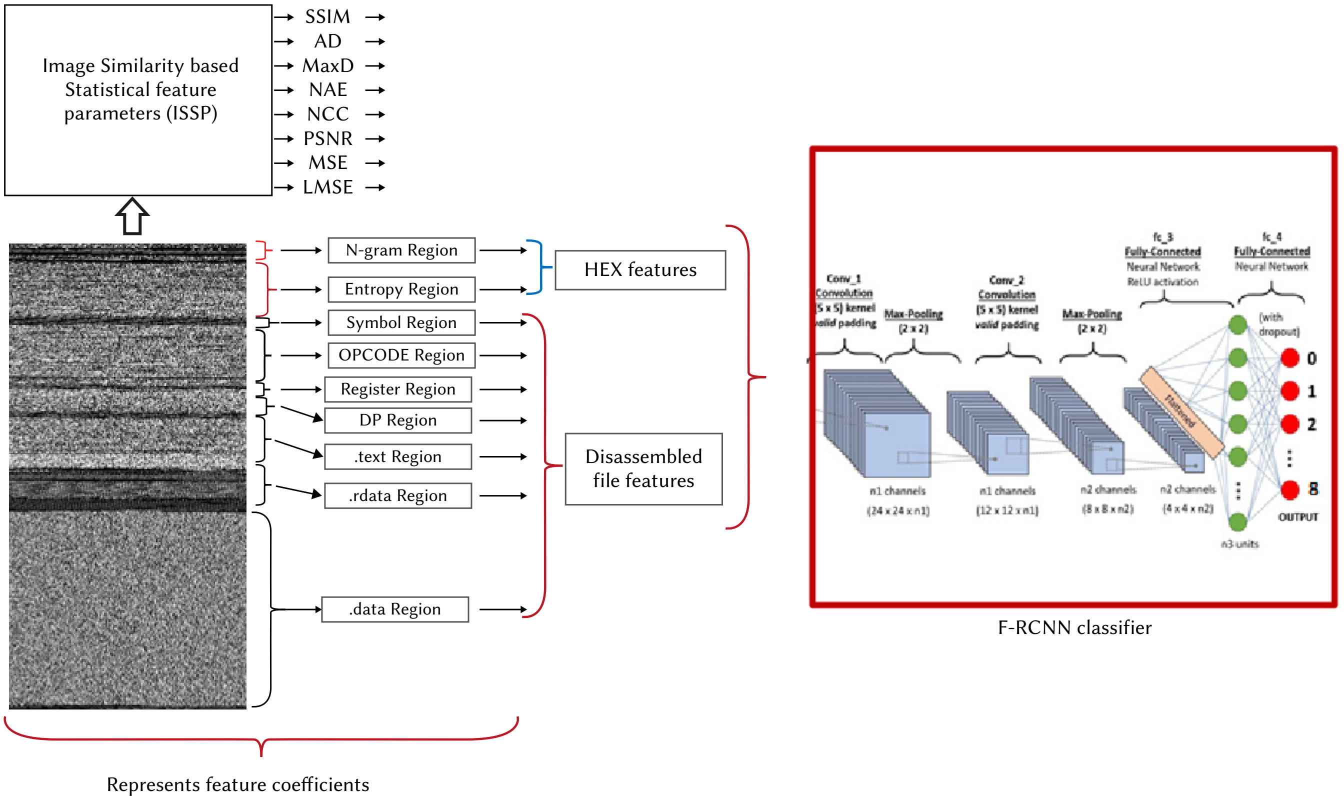


Image Similarity based Statistical feature parameters (ISSP)
SSIM
AD
MaxD
NAE
NCC
PSNR
MSE
LMSE
N-gram Region
Entropy Region
Symbol Region
OPCODE Region
Register Region
DP Region
.text Region
.rdata Region
.data Region
HEX features
Disassembled file features
Represents feature coefficients
Conv_1 Convolution (5 x 5) kernel valid padding
Max-Pooling (2 x 2)
Conv_2 Convolution (5 x 5) kernel valid padding
Max-Pooling (2 x 2)
fc_3 Fully-Connected Neural Network ReLU activation
fc_4 Fully-Connected Neural Network
(with dropout)
n1 channels (24 x 24 x n1)
n1 channels (12 x 12 x n1)
n2 channels (8 x 8 x n2)
n2 channels (4 x 4 x n2)
n3 units
0
1
2
8
OUTPUT
F-RCNN classifier

Fig. 9. F-RCNN Architecture for MDS.

Fig. 11 represents a sample of nine malware categories. The image has very fine and typical texture patterns and the same may be used to visualize the malware family. It has been observed that malware of the same family has similar signatures or fingerprints in some area of an image [187]. This information may be used as knowledge to identify the malware family. Zhang [186] represented information gains and the probability of the OPCODE to construct an image. M. Ahmadi [57] used Local Binary Pattern features and Haralick features to construct an image.

As R-CNN extracts features from each block, features from the same block will be repeatedly extracted, leading to a greater number of repetitive computations as shown in Fig. 9. **Therefore, the author proposes F-R-CNN, which is an improved version of R-CNN where CNN performs forward computation on the whole image**.

The entire image with a set of proposed regions (*K* object) with similarity based statistical parameters is input for an F-RCNN network. The network performs several convolution operations *(conv)* and maximum pooling layers on the entire input feature vector, and produces *conv* feature map. Next step it performs is to generate a feature map, by extracting a fixed length feature vector generated from the Region of Interest (RoI) pooling layer operation on the proposed objects. Each and every generated feature vector is input to a sequence of fully connected (f_c) layers. f_c layer provides two outputs. The first output is Softmax Probability Estimation (SPE) over *K* object classes plus a catch-all "background" class. The second output is 4 real valued numbers for every *K* object class. One of the *K* object classes has a set of four values which provides updated bounding box positions, calculated to reduce overlaps shown in Fig. 10.

Faster R-CNN consists of two major modules. The first module i.e., Region Proposal Network (RPN) is a deep fully convolutional network which proposes regions. The second module uses F-RCNN detector which uses the regions proposed by first module and learns about region positions from the same module due to attention-based CNN. The first module is presented in Table I. Table II presents the algorithm for generating the unified network for the overall system.

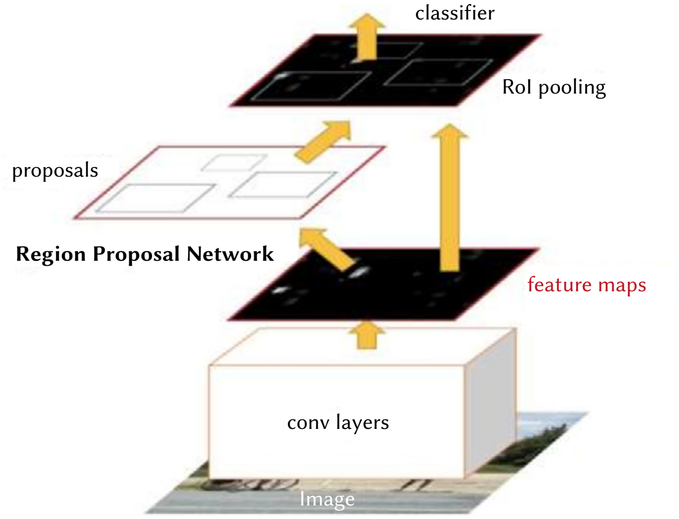


Fig. 10. F-RCNN Architecture.

VII. Statistical Parameters

To compute different statistical parameters, initially all the malware families are segregated in different folders. Nine folders are created as there are nine malware families. Malware files are processed and gray images are produced. These images are used to compute parameters. Table III presents the algorithmic steps.

TABLE I. Region Proposal Generation Algorithm

Region Proposal Network

Input: Image of any size

Output: 1. Set of rectangular object proposals
2. Each object with abjectness score

Convolutional model used: Zeiler and Fergus(ZF) (five sharable convolutional layers)

1 Generate convolutional feature map.

2 $n * n$ Sliding spatial window generated for small network. Sliding window is mapped to feature map of low dimension (256$d\,f\,or\,ZF$).

3 **Anchors**

a. Each sliding window predicts multiple region proposals, simultaneously.
b. k = Maximum region proposals for each location (k anchor boxes)
c. Feature map size = $W * H$
d. In the sliding window the question and an anchor, both are centered. Aspect ratio and scale is associated. Aspect ratio = 3 and scale = 3, provides $k = 9$ anchors [204].
e. 'cls' layer is a two class Softmax layer. It estimates the probability of the object or non-object for each region proposal. Logistic regression will be used to produce of 2 * k scores.
f. Regression ***(reg)*** layer has coordinates of k boxes. It will generate 4 * k outputs.
g. The total number of anchors will be $W * H * k$
h. Anchors are ***translation-invariant which reduces the model size***. As per the case of Fully Convolutional Network (FCN) [204]

4 **Multi-Scale Anchors**

a. Compute multi-scale anchors by selecting multiple scale sliding window.
b. Apply multi-scale sliding window to image and feature map of single scale.
c. Generated structure may be viewed as 'pyramid of filters'

5 **Loss Function**

a. Apply binary label to each and every bounding box.
b. ***If*** $(Intersection\ over\ Union\ (IoU) \cap Ground\ Truth(GT)\ \ box) \geq 0.7)$
then assign positive label
c. If $(Intersection\ over\ Union\ (IoU) \cap GTbox) \leq 0.3\ )$
then assign negative label
else discard anchor
d. Multiple anchors may be marked as positive by a single GT box.
e. Compute Loss function

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda * \frac{1}{N_{reg}} \sum_i P_i^* * L_{reg}(t_i, t_i^*) \tag{10}$$

where, i = anchor index in mini batch
P_i = *Predicted probability of anchor being an object*
P_i^* = *Binary GTlabel*
t_i = *four parameteroised co-ordinates of Bounding Box (BB)*
t_i^* = *four parameteroised co-ordinates of GT Box of positive anchor*
L_{cls} = log *loss over object versus non-object classes*
L_{reg} = *regression Loss calculated only when* $P_i^* = 1$ *cls layer has* $\{P_i\}$
and reg layer has $\{t_i\}$

f. Compute Robust Loss

$$L_{reg}(t_i, t_i^*) = R(t_i - t_i^*) \tag{11}$$

g. Normalize L_{cls} and L_{reg}
N_{cls} = *Normalisation of* L_{res} *by mini batchsize*
N_{reg} = *Normalisation of* L_{reg} *by the number of anchor locations*
h. *Apply weight factor* λ
i. BB regression from an anchor box to GT box

$$t_x = \frac{(x - x_a)}{w_a},\ t_y = \frac{(y - y_a)}{h_a},\ t_w = \log\frac{(w)}{w_a},\ t_h = \log\frac{(h)}{h_a},$$
$$t_x^* = \frac{(x^* - x_a)}{w_a},\ t_y^* = \frac{(y^* - y_a)}{h_a},\ t_w^* = \log\frac{(w^*)}{w_a},\ t_h^* = \log\frac{(h^*)}{h_a} \tag{12}$$

where,
x, y = centre co-ordinates of box
w and h = width and height of box
x, y, w and hare variables for predicted box
x_a*,* y_a*,* w_a *and* h_a *are variablesfor anchor box*
x^**,* y^**,*w^* *and* h^* *are variables for GTbox*

6 **RoI pooling layer algorithm**

1. Input RoI window of size $h * w$ with $r\,(row)\ and\ c\,(column)$ information.
2. Divide ROI window into sub windows of size $H * W$

$$window\ size_{approximate} = \frac{h}{H} * \frac{w}{W}$$

3. Generate output grid cell by maximum pooling values from each sub window
4. Apply independent pooling to each feature map channel, on a similar line to the standard maximum pooling.

7 **Optimization of Loss function**

1. From a single image mini-batch of many +/- anchors are identified.
2. Sample size of 256 anchors from an image is randomly selected to compute loss function of a mini batch.
3. Equal number of positive and negative anchors are selected i.e., 128 each, if sample size is 256.
4. In case of a low number of either of the anchors (≤ 128), then the mini batch will be padded in such a way to get equal numbers of both the anchors.
5. All new layers are randomly initialized by computing weights from zero mean Gaussian distribution with 0.01 standard deviation.
6. Shared convolutional layers are initialized by standard practice i.e. pre-training model a model for ImageNet classification.

8 **Training RPN**

RPN is trained by back propagation (BP) and stochastic gradient descent (SGD) using 'image centric' sampling strategy.

TABLE II. Approach to Unified Network of RPN and F-RCNN

Problem: Independently trained F-RCNN and RPN networks will modify their convolutional layer differently. The algorithm is required so that both can '*share*' the convolutional layer

Generating Unified system Network

Input: RPN generated regions

Output: Trained model of overall system

1 Train RPN network as per algorithm described in Table I

2 a. Initialized the ImageNet-pre-trained model
b. Input region proposals generated by step -1
c. Train detection network by Fast R-CNN using.
Note: convolutional layer is yet not shared by both the training model

3 *Shared* convolutional layers are finalized.

4 Layers unique to RPN will be fine-tuned.

5 RPN training network is initialized by detector network and the system is trained.

5 Fine tune layers from ***Shared*** convolutional layers unique to F-RCNN.

6 F-RCNN network is trained.

VIII. Database Description

This section describes the publicly available datasets for malware. System performance is analyzed on the benchmark database from Kaggle (https://www.kaggle.com/c/malware-classification/data).It was a Microsoft malware classification challenge. The database contained known malware files representing a mix of 9 different families viz. Gatak, Obfuscator.ACY, Kelihos_ver1, Tracur, Simda, Vundo, Kelihos_ver3, Lollipop and Ramnit. The file structure of the database is shown in Table IV and Table V.

TABLE III. Statistical Parameter Computation

```
Input   Folder structure is as follows
        Main folder – contains sub folders equal to number of
        malware families ($i = 9$ for this case)
            - Sub-folders (9 malware families)
            - Each sub-folder has different number of images $j$
                        $R_I = Reference\ image$
                        $I_I = Input\ image$
        $i = Number\ of\ malware\ families$ in main directory (folder)
        $j = Number\ of\ malware\ variants\ (images)\ of\ specific\ malware\ family\ in\ a\ subfolder$
        // Initialize empty array
        Parameter Array = {∅}
        for ($\beta = 0 ; \beta < i ; \beta++$)
        // Load reference image – first image of malware family
        $R_I = \beta[0]$
        for// select malware families one by one
            ($k = 0 ; k < i ; k++$)
        // Get number of images present of a specific malware family
            $j = size\ (k_{sub\text{-}folder})$
        for ($local_{cnt} = 0; local_{cnt} < j ; local_{cnt}++$)
        // Load Input image from malware family
            $I_i = (k)[local_{cnt}]$
        // calculate SSIM
        $SSIM(R_i, I_i) = [l(R_i, I_i)^{\alpha} . c(R_i, I_i)^{\beta} . s(R_i, I_i)^{\gamma}]$
        $where\ l = luminance,\ c = contrast,\ s = structure$
        // Calculate MSE
            $MSE(R_I, I_I) = \frac{1}{mn}\sum_{m=1}^{M}\sum_{n=1}^{N}[L(R_I(m,n)) - L(I_I(m.n))]^2$
        // calculate PSNR
            $PSNR = 10 * \log_{10}\frac{255^2}{MSE}$
        // calculate Normalized Cross-Correlation (NK)
            $NCC(R_I, I_I) = C_{R_I I_I}(\widehat{R_I}, \widehat{I_I}) = \sum_{[m,n] \in R} \widehat{R_I}(m,n)\widehat{I_I}(m,n)$
            $\widehat{R_I} = \frac{R_I - \overline{R_I}}{\sqrt{\sum(R_I - \overline{R_I})^2}}$ , $\widehat{I_I} = \frac{(I_I - \overline{I_I})}{\sqrt{\sum(I_I - \overline{I_I})^2}}$
        // calculate Normalized Absolute-error (NAE)
            $NAE(R_I, I_I) = \frac{\sum_{m=1}^{M}\sum_{n=1}^{N}|R_I(m,n) - I_I(m.n)|}{\sum_{m=1}^{M}\sum_{n=1}^{N}[R_I(m,n)]}$
        // calculate Maximum difference
            $MD(R_I, I_I) = \max\{[R_I(m,n) - I_I(m,n)]\}$
        // calculate Laplacian Mean Square Error (LMSE)
        $LMSE(R_I, I_I) = \frac{\sum_{m=1}^{M}\sum_{n=1}^{N}[L(R_I(m,n)) - L(I_I(m.n))]^2}{\sum_{m=1}^{M}\sum_{n=1}^{N}[L(R_I(m,n))]^2}$
        $whereL(m,n)\ is\ Laplacian\ operator$
        // Store all the values in an array
        end
        // Take average of an array an obtain single value
        Parameter_array(k)=[ mean(SSIM); mean(MSE); mean(PSNR);
        mean(NCC); mean(NAE); mean(MaxD); mean(LMSE)]
        end
        end
```

TABLE IV. Kaggle Dataset Basic Information

Header	Description
ID	Twenty-character hash value for unique identification of file
Class	Integer representing family of malware
RAW data	HEX representation of the file's binary content
Metadata manifest	Log of various metadata information e.g. Function calls, Strings etc. extracted from the binary using IDA disassembler tool.
Size	0.5 Tera byte uncompressed

TABLE V. DATASET Description

Malware Family	Malware category	Sample Size
Gatak	Backdoor	1013
Obfuscator. ACY	obfuscated malware	1228
Kelihos_ver1	Backdoor	398
Tracur	Trojan Downloader	751
Simda	Backdoor	42
Vundo	Trojan	475
Kelihos_ver3	Backdoor	2942
Lollipop	Adware	2478
RAmnit	Worm	1541

IX. Experimental Results and Discussion

The gray scale images of the feature vector for the malware family listed in Table V are shown in Fig. 11. It can be clearly observed that the image for each family is unique in itself. Identification becomes simpler. Feature vector 'Regions' are also clearly visible.

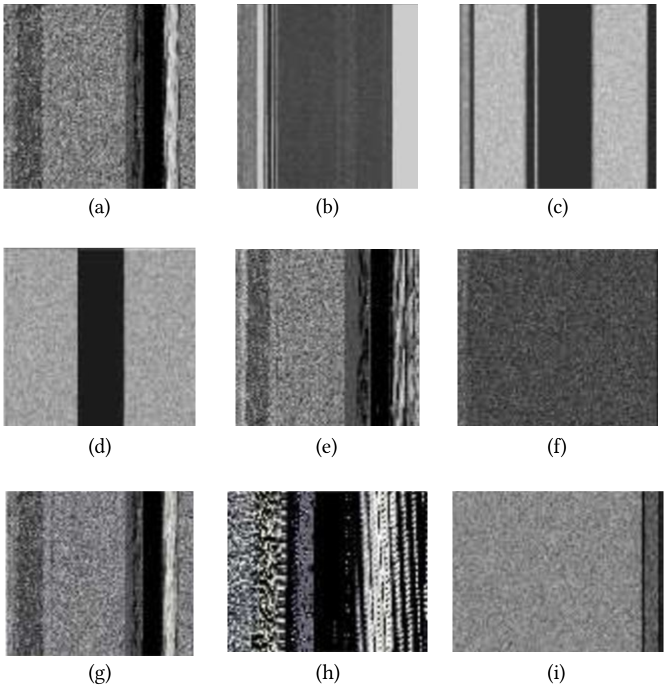


Fig. 11. Malware images of different malware families.
(a) Ramnit (b) Lollipop (c) Kelihos_ver3 (d) Vundo (e) Simda (f) Tracur (g) Kelihos_ver1 (h) Obfuscator. ACY (i) Gatak

As seen, initially one image from the first malware family is taken as a '*reference image*'. The remaining images from the first malware family and all the images of all eight malware families are used as '*Input image*'. Reference image and Input images are used to compute all the parameters. All the parameters per '*Input image*' are stored in the respective arrays (e.g.SSIM_array, MSE_array and so on). After iterating through all the images of one family, the mean value of an array is calculated. Thus, per family there is a single mean value. The mean value matrix is plotted. The same is depicted in Fig. 12.

Fig. 12(a) shows the SSIM value. It is 0.56 for the Ramint malware. For the remaining families the value ranges between 0 to 0.02. Thus, there is high structural similarity with self-family, but with other families less SSIM value reflects very little similarity. On a similar line, the NAE parameter for the same family is 0.4 and for other families it is more than 0.7. Refer Fig. 12 (b) – (h). It depicts plots for PSNR, MD, MSE, LMSE, NK. It has been observed that, there is clear bifurcation in

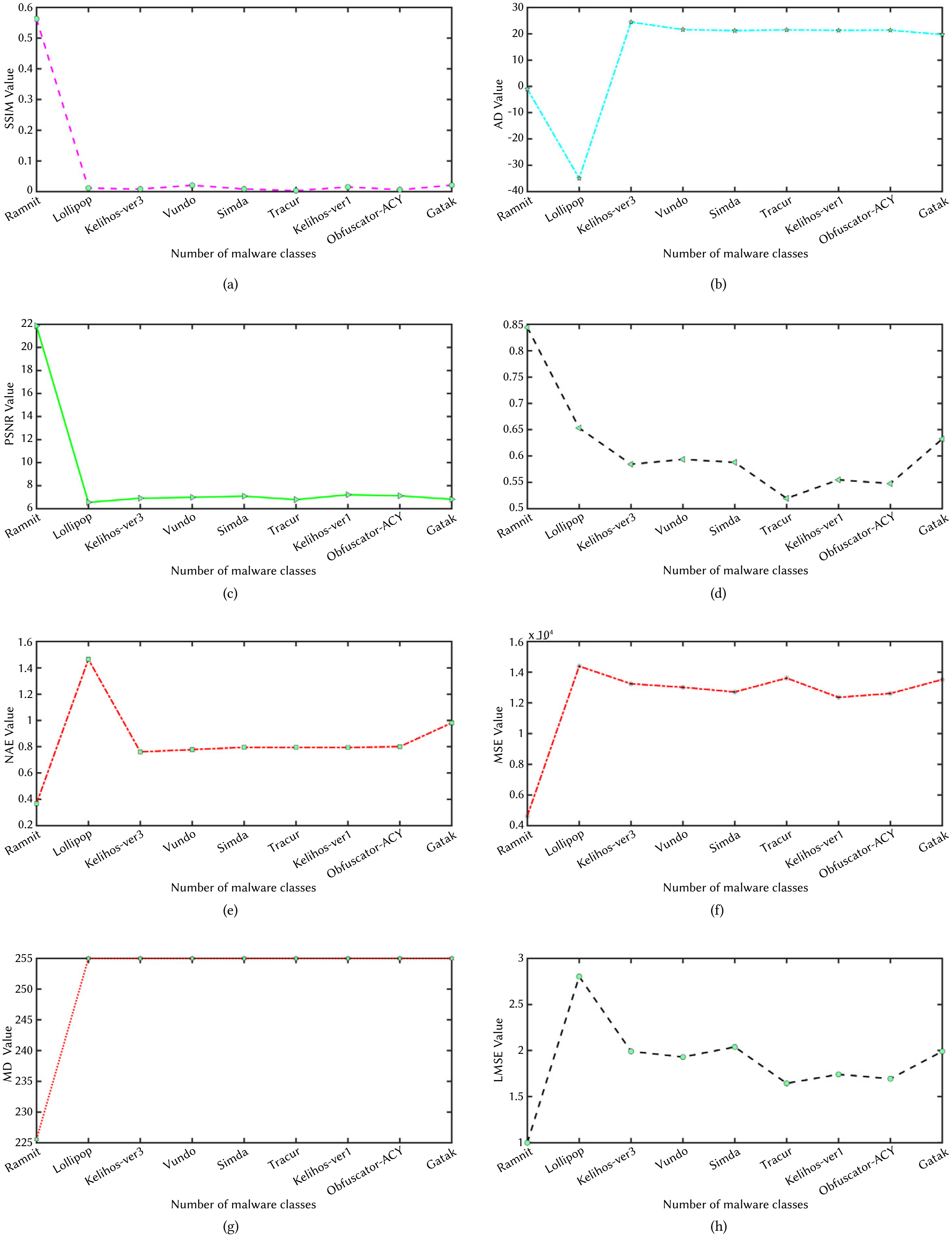


Fig. 12. Parameter plots - (a)SSIM (b) AD (c) PSNR (d) NK (NCC) (e) NAE (f) MSE (g) MD (h) LMSE.

the statistical parameter values to the same family class and a different family class. But in case of AD for the same class of family the value is approximately -1, but for the remaining families the value is either positive or negative with appropriate value difference. One can define a threshold range (0.95 to 1.1) for the AD. All these parameters can be used to train the F-RCNN classifier.

In Fig. 13 When the SC values were plotted for the same scenario then the results were not so promising. Range or proper threshold was difficult; therefore, this parameter was not taken into consideration by the author.

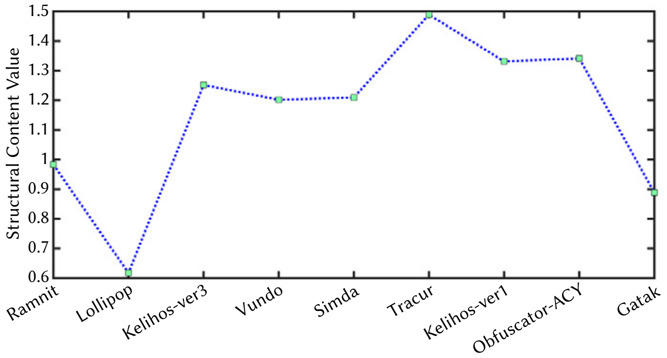


Fig. 13. SC plot.

Generated images are used to train the F-RCNN network. Annotations about regions are marked. In the experiment, the malware image set is randomly divided into a training set (60%) and a testing set (40%). This ratio of 60:40 has been selected to check the robustness of the model. The *training* data is less as compared to standard training data, i.e., 70:30. A trained library or network is created after the training process. After the training model is completed, the error and loss function of the model is used for judgment and evaluation. While training the system 30,000 epochs are selected. But while plotting the graph it is represented in percentage of the total value. Fig. 14 and Fig. 15 represent the error and loss plots in the training process. As the training iteration increases, the total error as well as loss value decreases and gradually stabilizes.

When the iterations reach 100%, the total loss value becomes flat and achieves the possible minimum value. The result shows that the training model based on F-RCNN with ISSP fusion with gray scale image is successful.

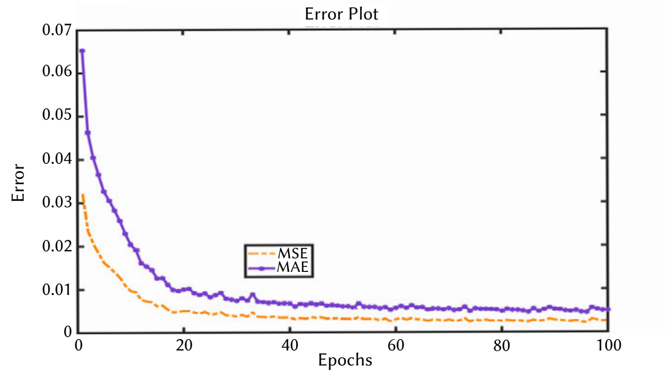


Fig. 14. MSE and MAE plot.

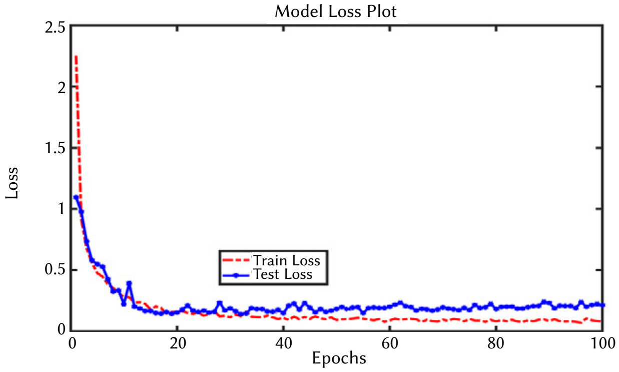


Fig. 15. Loss plot.

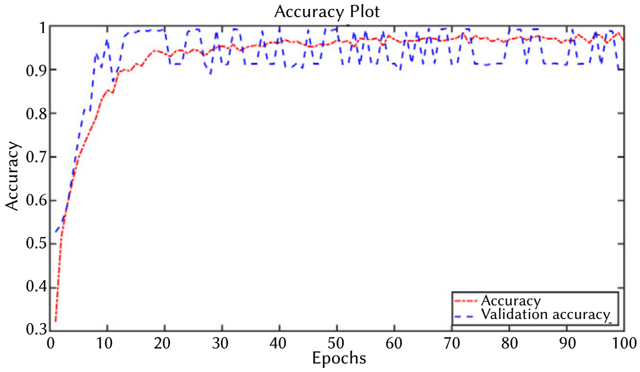


Fig. 16. Training model accuracy plot.

The network keeps a history of the trained data with error, loss and accuracy achieved while training the model. Fig. 16 represents the overall accuracy of the model with validation accuracy which approximates to 98.12%. In the testing phase, the remaining 30% of the malware files will be used. For each file a feature vector will be presented as a gray scale image will be generated and ISSP will be computed. The total matrix will be input to the trained network. The output generated from the trained network will be analyzed with the help of the statistical method where different parameters like True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) will be computed.

X. Performance Analysis

A. Performance Metrics

This section compares results obtained from the proposed work and state-of-the art methods. The proposed method opted the Kaggle benchmark dataset therefore results are compared with those research techniques that opted for the same dataset. Similarly, a comparison of the proposed algorithm for various performance metrics is stated in Table VI and Table VII respectively. Graphical plots for all comparisons are illustrated in Fig. 17.

Accuracy is the major performance parameter for the MD system, which specifies how accurately malwares are classified. Accuracy is calculated based on the following equation:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} X\ 100$$

Table VI depicts the confusion matrix of the proposed scheme for the MDS using Kaggle database.

TABLE VI. Confusion Matrix

Malware	Malware Detection %								
	Ramnit	Lollipop	Kelihos_ver3	Vundo	Simda	Tracur	Kelihos_ver1	Obfuscator.ACY	Gatak
RAmnit	98.57	0.19	0.19	0	0	0.06	0.45	0.45	0.06
Lollipop	0.36	98.82	0	0.40	0	0	0	0.40	0
Kelihos_ver3	0.03	0	99.66	0	0	0	0	0.30	0
Vundo	0	0	0	96.42	0.21	0.84	0	2.52	0
Simda	0	0	0	0	95.12	2.43	2.43	2.43	0
Tracur	0	0	0	0	0	99.73	0.13	0.13	0
Kelihos_ver1	0	0	0	0	0	0	100	0	0
Obfuscator.ACY	0.97	0.40	0.40	0.40	0.65	1.30	3.50	91.36	0.97
Gatak	0	0	0	0	0	0.09	0	0.19	99.70

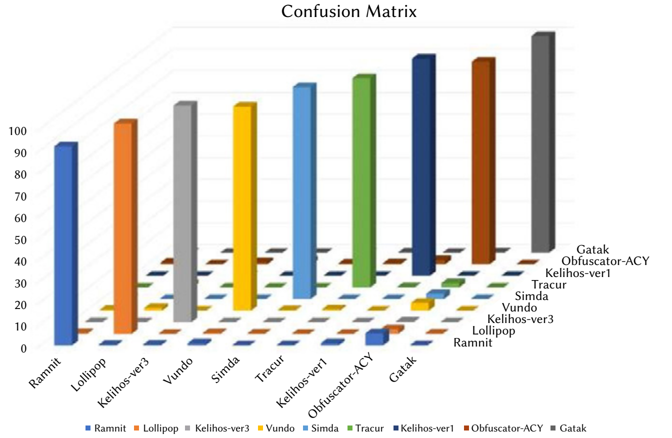


Fig. 17. Confusion Matrix Plot.

As the FRCNN classifier is not used by other researchers, the author compared the results with learning algorithms presented by researchers, as shown in Table VII.

TABLE VII. Comparative Performance of the Proposed System

Author/ Year	Dataset Used	Classifier	Accuracy (%)
Rao et al., 2017 [192]	NSL-KDD	IPDS-KNN	99.6
Shapoorifard et al., 2017 [193]	NSL-KDD	KFN-KNN	99
Vishwakarma et al., 2017 [194]	KDD cup 99	ACO-KNN	94.7
Dada et al., 2017 [195]	KDD cup 99	MIX-KNN	98.55
Ingre et al., 2017 [196]	NSL-KDD	CFS-DT	90.3
Malik et al., 2017 [197]	KDD cup 99	MULTI-DTs	91.94
Moon et al., 2017 [198]	Netflow	DT	84.7
Zhao et al., 2017 [199]	KDD cup 99	DBN-PNN	99.14
Tan et al., 2017 [200]	NETFLOW	DBN	97.6
Le et al., 2017 [201]	KDD cup 99	LSTM	97.54
Agarap et al., 2017 [202]	NETFLOW	GRU	84.15
Saxe et al., 2017 [203]	NETFLOW	CNN	92
Ding et al., 2016 [165]	Netflow	DBN	96.1
Nadeem et al., 2016 [120]	KDD cup 99	DBN	99.18
Alom et al., 2016 [159]	NSL-KDD	DBN	97.5
Krishnan et al., 2016 [180]	KDD cup 99	RNN	77.55
Kim et al., 2016 [173]	KDD cup 99	LSTM	96.93
MDFRCNN	**Kaggle dataset**	**F-RCNN**	**98.12**

XI. Conclusion

The paper proposes a state-of-the-art technique at feature extraction as well at classification level. The paper analyses different features viz. n-gram, MD1, MD2, entropy, OPCODE, Register, symbols, data define and sections of malware file for generating the feature vector. The feature vector is converted to a gray level image for visual analysis, where typical behavioral patterns can be observed for a particular malware family. Gray-scale image conversion widely opens up the scope for using state of the art image processing techniques, which have been more mature and proven.

Feature vectors have been presented in an image as different '*Regions*' which allows the use of Region Proposed Network (RPN). Exhaustive work done in the region-based analysis in an image, motivated the author to opt for the proposed methodology.

Malware codes are normally 75% to 80% identical. The image constructed from this code after extracting features should show similarity. Considering this point the author is motivated to introduce different image similarity based statistical parameters (ISSP) such as NCC, AD, MD, SSIM, LMSE, MMSE and PSNR as a feature set to improve system performance. The feature plot shown in Fig. 14, concludes that the features are distinctive. Thus, fusion of gray scale image with similarity parameters is used to train the classifier.

The development of region-based analysis with CNN as a base classifier offers R-CNN. The next modified versions of the basic R-CNN are Fast RCNN and Faster R-CNN (F-RCNN) techniques which have been proven for less training and testing time as shown in Fig. 5. This type of deep learning technique is more suitable for MDS where not only real time learning can be implemented with less time, but testing or producing output in the form of malware detection is desideratum. The system performance is analyzed using the benchmark database from Kaggle. This dataset is publicly available and results can be compared with the baseline. The database consists of nine malware families listed in Table V with details of malware families, malware categories and the number of sample files.

F-RCNN classifier with image-based visualization of the feature vector and ISSP as an additional feature resulted in better performance for classifying nine classes of malware. The proposed model offered an overall accuracy of 98.12% with improved rate of MD.

References

[1] E. Gandotra, D. Bansal, and S. Sofat, "Malware analysis and classification: A survey," Journal of Information Security, vol. 5, no. 02, pp. 56, 2014.

[2] Sahs, Justin & Khan, Latifur, "A Machine Learning Approach to Android Malware Detection" Proceedings - European Intelligence and Security Informatics Conference, EISIC 2012, pp.141-147, 2012.

[3] M. P. Deore and U.V. Kulkarni, "Malware Detection Techniques and its Classification: A Survey", International Journal of Research in Electronics

AND Computer Engineering (IJRECE), vol.6, no 4, pp.63-71, 2018.

[4] E. Bou-Harb, M. Debbabi and C. Assi, "Cyber Scanning: A Comprehensive Survey," in *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, pp. 1496-1519, 2014.

[5] M. P. Deore, U.V. Kulkarni and B.M. Patre, "Malware Classification Using Machine Learning: A Survey", Journal of Advanced Research in Dynamical and Control Systems (JARDCS), vol.10, Issue no.10, pp.181-190, 2018.

[6] Cohen, W. W, "Learning to classify English text with ILP methods", In Advances in Inductive Logic Programming, L. De Raedt, ed. IOS Press, Amsterdam, The Netherlands, pp.124-143, 2002.

[7] M. G. Schultz, E. Eskin, F. Zadok, S. J. Stolfo, "Data mining methods for detection of new malicious executable", Security and Privacy, Proceedings. 2001 IEEE Symposium, pp. 38-49, 2001.

[8] J. Z. Kolter, M. A. Maloof, "Learning to detect and classify malicious executables in the wild", Journal Machine Learning Research. 7, pp. 21-44, 2006.

[9] R. Tian, L. M. Batten, S. C. Versteeg, "Function length as a tool for malware classification", in: Malicious and Unwanted Software, MALWARE 2008. 3rd International Conference on, pp. 69-76, 2008.

[10] Zolkipli Mohamad Fadli, Aman Jantan, "An approach for malware behavior identification and classification", *Computer Research and Development (ICCRD) 2011 3rd International Conference on*, vol. 1, 2011.

[11] Shankarapani, M., Ramamoorthy, S., Movva, R., Mukkamala, S., "Malware detection using assembly and api call sequences". J. Comput. Virol. pp. 1–13, 2010.

[12] D. Kong, G. Yan, "Discriminant malware distance learning on structural information for automated malware classification", in: ACM SIGKDD '13,nKDD '13, ACM, New York, NY, USA, pp. 1357-1365,2013.

[13] Santos I., Devesa J., Brezo F., Nieves J., Bringas P.G, "OPEM: A Static-Dynamic Approach for Machine-Learning-Based Malware Detection", International Joint Conference CISIS'12-ICEUTE´12-SOCO´12 Special Sesstelligent Systems and Computing. Springer, Berlin, Heidelberg, vol 189 2013b.

[14] B. Gu, Y. Fang, P. Jia, L. Liu, L. Zhang and M. Wang, "A New Static Detection Method of Malicious Document Based on Wavelet Package Analysis," *2015 International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP)*, 2015, pp. 333-336,2015.

[15] Q. Li and X. Li, "Android Malware Detection Based on Static Analysis of Characteristic Tree," *2015 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*, Xi'an, pp. 84-91, 2015.

[16] Yoo, In Seon, "Visualizing windows executable viruses using self-organizing maps", pp. 82-89, 2004.

[17] D. A. Quist and L. M. Liebrock, "Visualizing compiled executables for malware analysis", *6th International Workshop on Visualization for Cyber Security*, Atlantic City, NJ, pp. 27-32, 2009.

[18] P. Trinius, T. Holz, J. Göbel and F. C. Freiling, "Visual analysis of malware behavior using tree maps and thread graphs", *6th International Workshop on Visualization for Cyber Security*, Atlantic City, NJ, pp. 33-38, 2009.

[19] Nataraj, L., Karthikeyan, S., Jacob, G. and Manjunath B, "Malware Images: Visualization and Automatic Classification", Proceedings of the 8th International Symposium on Visualization for Cyber Security, Article No. 4, 2011.

[20] K. Kancherla and S. Mukkamala, "Image visualization based malware detection", *IEEE Symposium on Computational Intelligence in Cyber Security (CICS)*, Singapore, pp. 40-44, *2013.*

[21] S. Han, H. Mao, and W. J. Dally, "Deep compression: Com-pressing deep neural network with pruning, trained quantization and Huffman coding", CoRR, abs/1510.00149, 2, 2015.

[22] M. Arefkhani and M. Soryani, "Malware clustering using image processing hashes", *9th Iranian Conference on Machine Vision and Image Processing (MVIP)*, Tehran, pp. 214-218,*2015.*

[23] Wu Q., Qin Z., Zhang J., Yin H., Yang G., Hu K, "Android Malware Detection Using Local Binary Pattern and Principal Component Analysis", In: Zou B., Li M., Wang H., Song X., Xie W., Lu Z. (eds) Data Science. ICPCSEE 2017. Communications in Computer and Information Science, vol 727. Springer, Singapore, 2017.

[24] S. Rezaei, A. Afraz, F. Rezaei, M. R. Shamani, "Malware detection using opcodes statistical features", in: 2016 8th International Symposium on Telecommunications (IST), pp. 151–155, 2016.

[25] B. Kolosnjaji, G. Eraisha, G. Webster, A. Zarras and C. Eckert, "Empowering convolutional networks for malware classification and analysis," *2017 International Joint Conference on Neural Networks (IJCNN)*, Anchorage, AK, pp. 3838-3845, 2017.

[26] S. Dübel, M. Röhlig, H. Schumann and M. Trapp, "2D and 3D presentation of spatial data: A systematic review," 2014 IEEE VIS International Workshop on 3DVis (3DVis), 2014, pp. 11-18, doi: 10.1109/3DVis.2014.7160094.

[27] N. Cao and W. Cui, "Introduction to Text Visualization", Atlantis Press, Paris, 2016.

[28] D. Keim, "Information visualization and visual data mining", IEEE Transactions on Visualization and Computer Graphics, vol. 8, no. 1, pp. 1–8, 2002.

[29] S. Few, "Information Dashboard Design - The Effective Visual Communication of Data", Sebastopol, CA: O'Reilly, 2006.

[30] J. Jacobs and B. Rudis, "Data-driven security analysis, visualization, and dashboards", in Indianapolis, John Wiley & Sons, 2014.

[31] N. Cao, L. Lu, Y.-R. Lin, F. Wang, and Z. Wen, "Social Helix: visual analysis of sentiment divergence in social media", Journal of Visualization, vol.18, no. 2, pp. 221–235, 2015.

[32] T. Songqing, "Imbalanced Malware Images Classification: a CNN based Approach", arXiv:1708.08042, 2017.

[33] W. B. Balakrishnan, "Security Data Visualisation", SANS Institute Inc, 2014.

[34] N. Diakopoulos, D. Elgesem, A. Salway, A. Zhang, and K. Hofland, "Compare clouds: visualizing text corpora to compare media frames", in Proceedings of IUI Workshop on Visual Text Analytics, 2015.

[35] H. Shiravi, A. Shiravi, and A. A. Ghorbani, "A survey of visualization systems for network security", IEEE Transactions on Visualization and Computer Graphics, vol.18, no.8, pp.1313–1329, 2012.

[36] Venkatraman, Sitalakshmi and Mamoun Alazab, "Use of Data Visualisation for Zero-Day Malware Detection", *Security and Communication Networks*, 1728303:1-1728303:13.,2018.

[37] T.Y.Zhang, X.M.WangLi, Z.Z.Li, F.Guo,Y.Ma, and W.Chen, "Survey of network anomaly visualization", Scince China Information Sciences, vol. 60, no. 12, 2017.

[38] W. Shanks, "Enhancing Intrusion Analysis through Data Visualization", SANS Institute, Inc, 2015.

[39] S.Foresti, J.Agutter, Y.Livnat, S.Moon, and R.Erbacher, "Visual correlation of network alerts", IEEE Computer Graphics and Applications, vol.26, no.2, pp.48–59,2006.

[40] M. Wagner, D. Sacha, A. Rind et al.,"Visual Analytics: Foundations and Experiences in Malware Analysis," in book: Empirical Research for Software Security: Foundations and Experience, L.benOthmane, M. GiljeJaatun, and E. Weippl, Eds., CRC/Taylor and Francis, pp. 139–171, 2017.

[41] K. Han, J. H. Lim, and E. G. Im, "Malware analysis method using visualization of binary files," in Proceedings of the the2013 Research in Adaptive and Convergent Systems, Montreal, Quebec, Canada, pp. 317–321, 2013.

[42] L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath, "Malware images: Visualization and automatic classification", in Proceedings of the 8th International Symposium on Visualization for Cyber Security, (VizSec '11), USA, 2011.

[43] N. Nissim, R. Moskovitch, L. Rokach, Y. Elovici, "Novel active learning methods for enhanced pc malware detection in windows OS", Expert Systems with Applications, vol. 41, no. 13, pp. 5843 – 5857, 2014.

[44] S. M. Tabish, M. Z. Shafiq, M. Farooq, Malware detection using statistical analysis of byte-level file content, in: Proceedings of the ACM SIGKDD Workshop on Cyber Security and Intelligence Informatics, CSI-KDD '09, pp. 23–31, 2009.

[45] W. Wong, M. Stamp, "Hunting for metamorphic engines", Journal in Computer Virology, vol. 2, no. 3, pp. 211-229, 2006.

[46] S. Attaluri, S. McGhee, M. Stamp, "Profile hidden Markova models and metamorphic virus detection", Journal in Computer Virology, pp.151-169, 2009.

[47] M. Siddiqui, M. C. Wang, J. Lee, "Detecting internet worms using data mining techniques", Journal of Systemic, Cybernetics and Informaticsm, pp.48-53, 2009.

[48] I. Santos, J. Nieves, P. G. Bringas, "Semi-supervised Learning for Unknown Malware Detection", International, Symposium on Distributed

Computing and Artificial Intelligence, Springer Berlin Heidelberg Berlin, Heidelberg, pp. 415-422, 2011.

[49] Z. Chen, M. Roussopoulos, Z. Liang, Y. Zhang, Z. Chen, A. Delis, "Malware characteristics and threats on the internet ecosystem", Journal of Systems and Software, pp.1650-1672, 2012.

[50] J. Yonts, "Attributes of malicious files", Tech. rep., The SANS Institute, 2012.

[51] X. Hu, K. G. Shin, S. Bhatkar, K. Gri_n, Mutantx-s, " Scalable malware clustering based on static features", in: USENIX Annual Technical Conference, pp. 187-198, 2013.

[52] D. Kong, G. Yan, "Discriminant malware distance learning on structural information for automated malware classification", in: ACM SIGKDD '13, nKDD '13, ACM, New York, NY, USA, pp. 1357-1365, 2013.

[53] I. Santos, F. Brezo, X. Ugarte-Pedrero, P. G. Bringas, "Opcode sequences as representation of executables for data-mining-based unknown malware detection", Information Sciences 231 pp.64-82, 2013.

[54] P. Vadrevu, B. Rahbarinia, R. Perdisci, K. Li, M. Antonakakis, "Measuring and detecting malware downloads in live network traffic", in: Computer Security ESORICS 2013: 18th European Symposium on Research in Computer Security, Egham, UK, September 9-13, 2013. Proceedings, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 556-573, 2013.

[55] J. Bai, J. Wang, G. Zou, "A malware detection scheme based on mining format information", The Scientific World Journal, 2014.

[56] A. Tamersoy, K. Roundy, D. H. Chau, "Guilt by association: large scale malware detection by mining file-relation graphs", in: Proceedings of the 20th ACM SIGKDD, ACM, pp. 1524-1533, 2014.

[57] M. Ahmadi, G. Giacinto, D. Ulyanov, S. Semenov, M. Tromov, "Novel feature extraction, selection and fusion for effective malware family classification", arXiv:1511.04317, 2016.

[58] M. Egele, T. Scholte, E. Kirda, C. Kruegel, "A survey on automated dynamic malware-analysis techniques and tools", ACM computing surveys (CSUR), 2012.

[59] Z. Feng, S. Xiong, D. Cao, X. Deng, X. Wang, Y. Yang, X. Zhou,Y. Huang, G. Wu, "Hrs: A hybrid framework for malware detection", In Proceedings of the 2015 ACM International Workshop on Security and Privacy Analytics, ACM, pp. 19-26, 2015.

[60] M. Gharacheh, V. Derhami, S. Hashemi, S. M. H. Fard, "Proposing an hmm-based approach to detect metamorphic malware", Fuzzy and Intelligent Systems (CFIS), pp. 1-5, 2015.

[61] P. Khodamoradi, M. Fazlali, F. Mardukhi, M. Nosrati, "Heuristic metamorphic malware detection based on statistics of assembly instructions using classification algorithms", in: Computer Architecture and Digital Systems (CADS), 2015 18th CSI International Symposium on, IEEE, pp.1-6, 2015.

[62] Pai, S., Troia, F.D., Visaggio, C.A.. "Clustering for malware classification", *Journal Computer Virology, Hack Tech*13, pp. 95–107, 2017.

[63] J. Sexton, C. Storlie, B. Anderson, "Subroutine based detection of APT malware", Journal of Computer Virology and Hacking Techniques, pp. 1-9, 2015.

[64] T. Lee, J. J. Mody, "Behavioral classification", In EICAR Conference, pp. 1-17, 2006.

[65] M. Bailey, J. Oberheide, J. Andersen, Z. M. Mao, F. Jahanian, J. Nazario, "Automated classification and analysis of internet malware", In Recent advances in intrusion detection, Springer, ,pp. 178-197, 2007.

[66] U. Bayer, P. M. Comparetti, C. Hlauschek, C. Kruegel, E. Kirda, Scalable, "Behavior-based malware clustering", In NDSS, vol. 9, pp. 8-11, 2009.

[67] I. Firdausi, C. Lim, A. Erwin, A. S. Nugroho, "Analysis of machine learning techniques used in behavior-based malware detection", in: ACT '10, IEEE, pp. 201-203, 2010.

[68] Y. Park, D. Reeves, V. Mulukutla, B. Sundaravel, "Fast malware classification by automated behavioral graph matching", in: Workshop on Cyber Security and Information Intelligence Research, ACM, pp. 45, 2010.

[69] B. Anderson, D. Quist, J. Neil, C. Storlie, T. Lane, "Graph-based malware detection using dynamic analysis", Journal in Computer Virology, vol. 7, no. 4, pp. 247-258, 2011.

[70] M. Lindorfer, C. Kolbitsch, P. M. Comparetti, "Detecting environment sensitive malware", in: Recent Advances in Intrusion Detection, Springer, pp. 338-357, 2011.

[71] K. Rieck, P. Trinius, C. Willems, T. Holz, "Automatic analysis of malware behavior using machine learning", Journal of Computer Security, vol. 19, no. 4, pp. 639-668, 2011.

[72] P. M. Comar, L. Liu, S. Saha, P. N. Tan, A. Nucci, "Combining supervised and unsupervised learning for zero-day malware detection", in: INFOCOM, 2013 Proceedings IEEE, pp. 2022-2030, 2013.

[73] G. E. Dahl, J. W. Stokes, L. Deng, D. Yu, " Large-scale malware classification using random projections and neural networks", in: Acoustics, Speech and Signal Processing (ICASSP), IEEE, pp. 3422-3426, 2013.

[74] S. Nari, A. A. Ghorbani, "Automated malware classification based on network behavior", in: Computing, Networking and Communications (ICNC), 2013 International Conference on, IEEE, pp. 642-647, 2013.

[75] S. Palahan, D. Babi_c, S. Chaudhuri, D. Kifer, "Extraction of statistically significant malware behaviors", in: Computer Security Applications Conference, ACM, pp. 69-78, 2013.

[76] M. Kruczkowski, E. N. Szynkiewicz, "Support vector machine for malware analysis and classification", in: Web Intelligence (WI) and Intelligent Agent Technologies (IAT), IEEE Computer Society, pp. 415-420, 2014.

[77] D. Uppal, R. Sinha, V. Mehra, V. Jain, "Malware detection and classification based on extraction of api sequences", in: ICACCI, IEEE, pp. 2337-2342, 2014.

[78] A. Elhadi, M.A. Maarof, B. Barry, "Improving the Detection of Malware Behaviour Using Simplified Data Dependent API Call Graph", International Journal of Security and Its Applications, vol. 7, no. 5, pp. 29-42, 2013.

[79] M. Ghiasi, A. Sami, Z. Salehi, "Dynamic VSA: a framework for malware detection based on register contents", Engineering Applications of Artificial Intelligence, pp.111- 122, 2015.

[80] N. Kawaguchi, K. Omote, "Malware function classification using apis in initial behavior", in: Information Security (AsiaJCIS), 2015 10th Asia Joint Conference on, IEEE, pp. 138-144, 2015.

[81] C.-T. Lin, N.-J. Wang, H. Xiao, C. Eckert, "Feature selection and extraction for malware classification", Journal of Information Science and Engineering, vol. 31, no. 3, pp. 965-992, 2015.

[82] A. Mohaisen, O. Alrawi, M. Mohaisen, "Amal: High-fidelity, behavior based automated malware analysis and classification", Computers & security, vol. 52, pp. 251-266, 2015.

[83] T. Wuchner, M. Ochoa, A. Pretschner, "Robust and effective malware detection through quantitative data flow graph metrics," in: Detection of Intrusions and Malware, and Vulnerability Assessment, Springer, pp. 98-118, 2015.

[84] G. Liang, J. Pang, C. Dai, "A behavior-based malware variant classification technique", International Journal of Information and Education Technology, vol. 6, pp.291, 2016.

[85] J. Jang, D. Brumley, S. Venkataraman, "Bitshred: feature hashing malware for scalable triage and semantic analysis", in: Computer and communications security, ACM, pp. 309-320, 2011.

[86] B. Anderson, C. Storlie, T. Lane, "Improving malware classification: bridging the static/dynamic gap", in: Proceedings of the 5th ACM workshop on Security and artificial intelligence, ACM, pp. 3-14, 2012.

[87] M. Eskandari, Z. Khorshidpour, S. Hashemi, "Hdm-analyser: a hybrid analysis approach based on data mining techniques for malware detection", Journal of Computer Virology and Hacking Techniques, vol. 9, pp. 77-93, 2013.

[88] R. Islam, R. Tian, L. M. Batten, S. Versteeg, "Classification of malware based on integrated static and dynamic features", Journal of Network and Computer Applications, pp.646-656, 2013.

[89] M. Egele, M.Woo, P. Chapman, D. Brumley, "Blanket execution: Dynamic similarity testing for program binaries and components", in: USENIX Security 14, USENIX Association, San Diego, CA, pp. 303-317, 2014.

[90] M. Graziano, D. Canali, L. Bilge, A. Lanzi, D. Balzarotti, "Needles in a haystack: Mining information from public dynamic analysis sandboxes for malware intelligence", in: USENIX Security '15, pp. 1057-1072, 2015.

[91] M. Polino, A. Scorti, F. Maggi, S. Zanero, "Jackdaw: Towards Automatic Reverse Engineering of Large Datasets of Binaries, in: Detection of Intrusions and Malware", and Vulnerability Assessment, Lecture Notes in Computer Science, Springer International Publishing, pp. 121-143, 2015.

[92] P. Vadrevu, R. Perdisci, "MAXS: Scaling Malware Execution with Sequential Multi-Hypothesis Testing", in: ASIA CCS '16, ASIA CCS '16, ACM, New York, NY, USA, pp. 771-782, 2016.

[93] M. P. Deore and U.V. Kulkarni, "Static Way of Effective Feature Extraction

and Malware Classification", Online International Interdisciplinary Research Journal, International Conference on Recent Multidisciplinary Research (ICRMR-2018), Organized and Hosted by Foundation of Innovative Research at conference center, AIT,Thailand ,vol.8 , no 2, pp.81-93, 2018.

[94] M. Asquith, "Extremely scalable storage and clustering of malware metadata", Journal of Computer Virology and Hacking Techniques, pp. 1-10, 2015.

[95] W. Mao, Z. Cai, D. Towsley, X. Guan, "Probabilistic inference on integrity for access behavior based malware detection", in: International Workshop on Recent Advances in Intrusion Detection, Springer, pp. 155-176, 2015.

[96] F. Ahmed, H. Hameed, M. Z. Shafiq, M. Farooq, "Using spatio-temporal information in api calls with machine learning algorithms for malware detection", in: Proceedings of the 2nd ACM workshop on Security and artificial intelligence, ACM, pp. 55-62, 2009.

[97] E. Raff, C. Nicholas, "An alternative to ncd for large sequences, lempel-zivjaccard distance", in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, pp. 1007-1015, 2017.

[98] D. H. Chau, C. Nachenberg, J. Wilhelm, A. Wright, C. Faloutsos, "Polonium: Tera-scale graph mining for malware detection", in: ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 131-142, 2010.

[99] Y. Ye, T. Li, Y. Chen, Q. Jiang, "Automatic malware categorization using cluster ensemble", in: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, pp. 95- 104, 2010.

[100] M. Lindorfer, C. Kolbitsch, P. M. Comparetti, "Detecting environment sensitive malware", in: Recent Advances in Intrusion Detection, Springer, pp. 338-357, 2011.

[101] B. J. Kwon, J. Mondal, J. Jang, L. Bilge, T. Dumitras, "The dropper effect: Insights into malware distribution with downloader graph analytics", in: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, ACM, pp. 1118-1129, 2015.

[102] J. Saxe, K. Berlin, "Deep neural network based malware detection using two dimensional binary program features, in: Malicious and Unwanted 47Software (MALWARE)", 2015 10th International Conference on, IEEE, pp. 11-20, 2015.

[103] K. Huang, Y. Ye, Q. Jiang, Ismcs: an intelligent instruction sequence based malware categorization system, in: Anti-counterfeiting, Security, and Identification in Communication, 2009, IEEE, pp. 509-512, 2009.

[104] Hardy, W. Chen, L.Hou, S. Ye, Y. Li X, "A deep learning framework for intelligent malware detection", In Proceedings of the International Conference Data Mining (ICDM), Barcelona, Spain, pp. 61, 2016.

[105] Wang X., Yiu S.M. "A multi-task learning model for malware classification with useful file access pattern from API call sequence", arXiv, arXiv:1610.05945, 2016.

[106] Javaid, Salman., "Analysis and Detection of Heap-based Malwares Using Introspection in a Virtualized Environment.", University of New Orleans Theses and Dissertations. 1875. 2014.

[107] Ma, T.; Wang, F.; Cheng, J.; Yu, Y.; Chen, X., "A Hybrid Spectral Clustering and Deep Neural Network Ensemble Algorithm for Intrusion Detection in Sensor Networks.", Sensors, 1701.,2016.

[108] Aminanto, M.E., Kim, K., "Deep Learning-Based Feature Selection for Intrusion Detection System in Transport Layer", Available online:https://pdfs.semanticscholar.org/bf07/e753401b36662eee7b8cd6c65cb8cfe31562.pdf (accessed on 23 February 2019).

[109] Diro, A.A. Chilamkurti, N., "Deep learning: The frontier for distributed attack detection in Fog-to-Things computing", IEEE Communication, pp.169–175, 2018.

[110] Chawla, S., "eep Learning Based Intrusion Detection System for Internet of Things", University of Washington: Seattle, WA, USA, 2017.

[111] Cox, J.A. James, C.D. Aimone, J.B., "A signal processing approach for cyber data classification with deep neural networks", Procedia Comput. Sci., pp.61, 349–354, 2015.

[112] Wang, Z. "The Applications of Deep Learning on Traffic Identification", Black Hat: Washington, DC, USA, 2015.

[113] Lotfollahi, M.; Shirali, R.; Siavoshani, M.J.; Saberian, M. ,"Deep Packet: A Novel Approach for Encrypted Traffic Classification Using Deep Learning", arXiv:1709.02656, 2017.

[114] Mi, G.; Gao, Y.; Tan, Y., "Apply stacked auto-encoder to spam detection", In Proceedings of the International Conference in Swarm Intelligence, Beijing, China, 26–29 ,pp. 3–15,2015.

[115] Loukas, G., Vuong, T., Heartfield, R., Sakellari, G., Yoon, Y., Gan, D., "Cloud-based cyber-physical intrusion detection for vehicles using Deep Learning", pp. 3491–3508, 2018.

[116] Diro, A.A.; Chilamkurti, N. Leveraging LSTM Networks for Attack Detection in Fog-to-Things Communications", IEEE Commun. Mag. 56, pp. 124–130, 2018.

[117] Shi, C.; Liu, J.; Liu, H.; Chen, Y., "Smart user authentication through actuation of daily activities leveraging WiFi-enabled IoT " , In Proceedings of the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing, Chennai, India, ACM: New York, NY, USA, pp. 10–14, 2017.

[118] Yousefi-Azar, M.; Varadharajan, V.; Hamey, L.; Tupakula, U. Auto encoder-based feature learning for cyber security applications. In Proceedings of the 2017 International Joint Conference Neural Networks (IJCNN) Anchorage, AK, USA, 14–19, pp. 3854–3861,2017.

[119] Abdulhammed, R., Faezipour, M., Abuzneid, A., AbuMallouh, A., "Deep and machine learning approaches for anomaly-based intrusion detection of imbalanced network traffic", IEEE Sens. Lett., 2018.

[120] Nadeem M., Marshall O., Singh, S., Fang, X., Yuan X., Semi-Supervised Deep Neural Network for Network Intrusion Detection", Available online: https: //digitalcommons .kennesaw . edu/ccerp/2016/Practice/2/ (accessed on 23 February 2019).

[121] Alom, M.Z. Taha, T.M., "Network intrusion detection for cyber security using unsupervised deep learning approaches", In Proceedings of the 2017 IEEE National Aerospace and Electronics Conference (NAECON), Dayton, OH, USA, 27–30 ,pp. 63–69, June 2017.

[122] Mirsky, Y.; Doitshman, T.; Elovici, Y.; Shabtai, A. Kitsune: "An ensemble of auto encoders for online network intrusion detection", arXiv:1802.09089., 2018.

[123] David, O.E.; Netanyahu, N.S. "Deep sign: Deep learning for automatic malware signature generation and classification", In Proceedings of the 2015 International Joint Conference Neural Networks (IJCNN), Killarney, Ireland, pp. 1–8., 2015.

[124] Saxe, J.; Berlin, K. "Deep neural network based malware detection using two dimensional binary program features", In Proceedings of the 10th International Conference Malicious and Unwanted Software (MALWARE),Washington, DC, USA, pp. 11–20, 2015.

[125] Mizuno, S.; Hatada, M.; Mori, T.; Goto, S. "Bot Detector: A robust and scalable approach toward detecting malware-infected devices", In Proceedings of the 2017 IEEE International Conference Communications (ICC), Paris, France; pp. 1–7, 2017.

[126] S. Srakaew, W. Piyanuntcharatsr, S. Adulkasem, On the comparison of malware detection methods using data mining with two feature sets", Journal of Security and Its Applications , pp. 293-318, 2015.

[127] Grosse, K.; Papernot, N.; Manoharan, P.; Backes, M.; McDaniel, P., "Adversarial perturbations against deep neural networks for malware classification", arXiv:1606.04435, 2016.

[128] Cordonsky, I.; Rosenberg, I.; Sicard, G.; David, E.O., "Deep Origin: End-to-end deep learning for detection of new malware families", In Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil; pp. 1–7,2018.

[129] Huang, W.; Stokes, J.W., "MtNet: A multi-task neural network for dynamic malware classification. In Proceedings of the International Conference Detection of Intrusions and Malware, and Vulnerability Assessment", Donostia-San Sebastián, Spain, pp. 399–418, 2016.

[130] Roy, S.S.; Mallik, A.; Gulati, R.; Obaidat, M.S.; Krishna, P.V. "A Deep Learning Based Artificial Neural Network Approach for Intrusion Detection", In Proceedings of the International Conference Mathematics and Computing, Haldia, India, pp. 44–53, 2017.

[131] Tang, T.A. Mhamdi, L. McLernon, D. Zaidi, S.A.R. Ghogho, M., "Deep learning approach for network intrusion detection in software defined networking", In Proceedings of the 2016 International Conference Wireless Networks and Mobile Communication (WINCOM), Fez, Morocco, pp. 258–263, 2016.

[132] Diro, A.A. Chilamkurti, N. Distributed attack detection scheme using deep learning approach for internet of things", Future Gener. Comput. Syst., 82, 761–768, 2018.

[133] Mi, G.; Gao, Y.; Tan, Y. "Apply stacked auto-encoder to spam detection", In Proceedings of the International Conference in Swarm Intelligence, Beijing, China; pp. 3–15, 2015.

[134] Yu, Y.; Long, J.; Cai, Z., "Network intrusion detection through stacking dilated convolutional auto encoders", Secur. Commun. Netw. , 2017.

[135] Gibert, D., "Convolutional Neural Networks for Malware Classification", Universit at Politècnica de Catalunya: Barcelona, Spain, 2016.

[136] Zeng, F.; Chang, S.;Wan, X., "Classification for DGA-Based Malicious Domain Names with Deep Learning Architectures", Int. J. Intell. Inf. Syst., 6, pp. 67–71, 2017.

[137] Yamanishi, K., "Detecting Drive-By Download Attacks from Proxy Log Information Using Convolutional Neural Network", Osaka University: Osaka, Japan, 2017.

[138] McLaughlin, N. del Rincon, J.M. Kang, B. Yerima, S. Miller, P. Sezer, S. Safaei, Y. Trickel, E. Zhao, Z. Doupe, A., " Deep android malware detection", In Proceedings of the 7th ACM on Conference on Data and Application Security and Privacy, Scottsdale, AZ, USA, pp. 301–308, 2017.

[139] Wang, W. Zhu, M. Zeng, X. Ye, X. Sheng, Y., "Malware traffic classification using convolutional neural network for representation learning. In Proceedings of the IEEE 2017 International Conference on Information Networking (ICOIN), Da Nang, Vietnam pp. 712–717, 2017.

[140] Wang, W. Zhu, M. Wang, J. Zeng, X. Yang, Z., "End-to-end encrypted traffic classification with one-dimensional convolution neural networks", In Proceedings of the 2017 IEEE International Conference Intelligence and Security Informatics (ISI), Beijing, China, pp. 43–48., 2017.

[141] Shibahara, T. Yamanishi, K. Takata, Y. Chiba, D.Akiyama, M. Yagi, T. Ohsita, Y. Murata, M., "Malicious URL sequence detection using event de-noising convolutional neural network", In Proceedings of the 2017 IEEE International Conference Communications (ICC), Paris, France, pp. 1–7,2017.

[142] Hill, G.D. Bellekens, X.J.A., "Deep learning based cryptographic primitive classification", arXiv2017, arXiv: 1709.08385.

[143] Kolosnjaji, B. Zarras, A.Webster, G. Eckert, C., "Deep learning for classification of malware system call sequences", In Proceedings of the Australasian Joint Conf. on Artificial Intelligence, Hobart, Australia, pp. 137–149,2016.

[144] Tobiyama, S., Yamaguchi, Y. Shimada, H. Ikuse, T., Yagi, T., "Malware detection with deep neural network using process behavior", In Proceedings of the IEEE 40th Annual Computer Software and Applications Conference (COMPSAC), Atlanta, GA, USA, Volume 2, pp. 577–5822016.

[145] Mac, H. Tran, D. Tong, V. Nguyen, L.G. Tran, H.A., "DGA Botnet Detection Using Supervised Learning Methods", In Proceedings of the 8th International Symposium on Information and Communication Technology, Nhatrang, Vietnam, pp. 211–218, 2017.

[146] Yu, B. Gray, D.L. Pan, J. de Cock, M. Nascimento, "A.C.A. Inline DGA detection with deep networks", In Proceedings of the 2017 IEEE International Conference Data Mining Workshops (ICDMW), New Orleans, LA, USA, pp. 683–692.

[147] Anderson, H.S. Woodbridge, J. Filar, B., "DeepDGA: Adversarially-tuned domain generation and detection. In Proceedings of the 2016 ACM Workshop on Artificial Intelligence and Security, Vienna, Austria, pp. 13–21, 2016.

[148] Li, Y. Ma, R. Jiao, R., "A hybrid malicious code detection method based on deep learning. Methods 2015, 9, 205–216.

[149] Maimó, L.F. Gómez, A.L.P. Clemente, F.J.G. Pérez, M.G., "A self-adaptive deep learning-based system for anomaly detection in 5G networks", IEEE Access, 6, pp. 7700–7712, 2018.

[150] Alrawashdeh, K. Purdy C., "Toward an online anomaly intrusion detection system based on deep learning", In Proceedings of the 15th IEEE International Conference Machine Learning and Applications (ICMLA), Miami, FL, USA, pp. 195–200, 2015.

[151] Yuan, Z. Lu, Y. Wang, Z. Xue, Y., "Droid-sec: Deep learning in android malware detection", ACM SIGCOMM Comput. Commun. Rev. pp. 44, 371–372, 2014.

[152] Weber M., Schmid M., Schatz M., Geyer D., "A toolkit for detecting and analyzing malicious software", In Proceedings of the 18th Annual Computer Security Applications Conference, Las Vegas, NV, USA, pp. 423–431, 2002.

[153] Hou, S. Saas, A. Ye, Y. Chen, L., "Droiddelver: An android malware detection system using deep belief network based on API call blocks", In Proceedings of the International Conference Web-Age Information Manage, Nanchang, China, pp. 54–66, 2016.

[154] Xu, L. Zhang, D. Jayasena, N. Cavazos, J. "HADM: Hybrid analysis for detection of malware", In Proceedings of the SAI Intelligent Systems Conference, London, UK, pp. 702–724, 2016.

[155] Benchea, R. Gavrilu ¸ t, "D.T. Combining restricted Boltzmann machine and one side perceptron for malware detection", In Proceedings of the International Conference on Conceptual Structures, Iasi, Romania, pp. 93–103,2014.

[156] Zhu, D. Jin, H. Yang, Y. Wu, D. Chen, W., "Deep Flow: Deep learning-based malware detection by mining Android application for abnormal usage of sensitive data", In Proceedings of the 2017 IEEE Symposium Computers and Communications (ISCC), Herakli on, Greece, pp. 438–443, 2017.

[157] Ye, Y. Chen, L. Hou, S. Hardy, W. Li, X., "DeepAM: A heterogeneous deep learning framework for intelligent malware detection", Knowl. Inf. Syst., pp.265–285, 2018.

[158] Gao, N.; Gao, L.; Gao, Q.; Wang, H., "An intrusion detection model based on deep belief networks", In Proceedings of the 2014 2nd International Conference Advanced Cloud and Big Data (CBD), Huangshan, China, pp. 247–252,2014.

[159] Alom, M.Z. Bontupalli, V. Taha, T.M. Intrusion detection using deep belief networks", In Proceedings of the 2015 National Aerospace and Electronics Conference (NAECON), Dayton, OH, USA, pp. 339–344, 2015.

[160] Dong, B.; Wang, X. "Comparison deep learning method to traditional methods using for network intrusion detection", In Proceedings of the 8th IEEE International Conference Communication Software and Networks (ICCSN), Beijing, China, pp. 581–585, 2016.

[161] Kang, M.J. Kang, J.W., "Intrusion detection system using deep neural network for in-vehicle network security", PLoS ONE, e0155781, 2016.

[162] Nguyen, K.K. Hoang, D.T. Niyato, D., "Wang, P.; Nguyen, P.; Dutkiewicz, E. Cyberattack detection in mobile cloud computing", A deep learning approach. In Proceedings of the 2018 IEEE Wireless Communications and Networking Conference (WCNC), Barcelona, Spain, pp. 1–6, 2018.

[163] Tzortzis, G.; Likas, A., "Deep Belief Networks for Spam Filtering. in Tools with Artificial Intelligence", In Proceedings of the 2007 19th IEEE International Conference on ICTAI, Patras, Greece, Volume 2, pp. 306–309, 2007.

[164] He, Y.; Mendis, G.J.; Wei, J., "Real-time detection of false data injection attacks in smart grid A deep learning-based intelligent mechanism", IEEE Trans. Smart Grid, pp. 2505–2516, 2017.

[165] Ding, Y.; Chen, S.; Xu, J., "Application of Deep Belief Networks for opcode based malware detection", In Proceedings of the 2016 International Joint Conference on Neural Networks (IJCNN), Vancouver, BC, Canada, pp. 3901–3908, 2016.

[166] J. Upchurch, X. Zhou, "Variant: a malware similarity testing framework", in: 2015 10th International Conference on Malicious and Unwanted Software (MALWARE), IEEE, pp. 31-39, 2015.

[167] Pascanu, R.; Stokes, J.W.; Sanossian, H.; Marinescu, M.; Thomas, "A. Malware classification with recurrent networks", In Proceedings of the 2015 IEEE International Conference Acoustics, Speech and Signal Process, (ICASSP), Brisbane, Australia, pp. 1916–1920,2015.

[168] Shibahara, T.; Yagi, T.; Akiyama, M.; Chiba, D.; Yada, T. "Efficient dynamic malware analysis based on network behavior using deep learning", In Proceedings of the 2016 IEEE Global Communications Conference (GLOBECOM),Washington, DC, USA, pp. 1–7, 2016.

[169] Woodbridge, J.; Anderson, H.S.; Ahuja, A.; Grant, D. ,"Predicting domain generation algorithms with long short-term memory networks", arXiv2016, arXiv:1611.00791, 2016.

[170] Lison, P.; Mavroeidis, V., "Automatic Detection of Malware-Generated Domains with Recurrent Neural Models", arXiv2017, arXiv:1709.07102, 2017.

[171] Tran, D.; Mac, H.; Tong, V.; Tran, H.A.; Nguyen, L.G., "A LSTM based framework for handling multiclass imbalance in DGA botnet detection", Neurocomputing, pp 2401–2413, 2018.

[172] Torres, P.; Catania, C.; Garcia, S.; Garino, C.G., "An Analysis of Recurrent Neural Networks for Botnet Detection Behavior", In Proceedings of the 2016 IEEE Biennial Congress of Argentina (ARGENCON), Buenos Aires,

Argentina, pp. 1–6, 2016.

[173] Kim, J. Kim, J.; Thu, H.L.T.; Kim, H. "Long Short Term Memory Recurrent Neural Network Classifier for Intrusion Detection", In Proceedings of the 2016 International Conference Platform Technology and Service (PlatCon), Jeju, Korea, pp. 1–5, 2016.

[174] Kim, J.; Kim, H. "Applying recurrent neural network to intrusion detection with hessian free optimization", In Proceedings of the International Conference on Information Security Applications, Jeju Island, Korea, pp. 357–369,2015.

[175] Kim, G.; Yi, H.; Lee, J.; Paek, Y.; Yoon, S., "LSTM-Based System-Call Language Modeling and Robust Ensemble Method for Designing Host-Based Intrusion Detection Systems", arXiv2016, arXiv:1611.01726, 2016.

[176] Loukas, G. Vuong, T. Heartfield, R.; Sakellari, G. Yoon, Y. Gan, D. "Cloud-based cyber-physical intrusion detection for vehicles using Deep Learning", IEEE Access, vol. 6, pp. 3491–3508, 2018.

[177] Cheng, M. Xu, Q. Lv, J. Liu,W. Li, Q.Wang, J., "MS-LSTM: A multi-scale LSTM model for BGP anomaly detection", In Proceedings of the IEEE 24th International Conference Network Protocols (ICNP), Singapore, pp. 1–6,2016.

[178] Kobojek, P.; Saeed, K., "Application of recurrent neural networks for user verification based on keystroke dynamic", J. Telecommun. Inf. Technol., pp.80–90., 2016.

[179] McDermott, C.D. Majdani, F. Petrovski, A., "Botnet detection in the internet of things using deep learning approaches", In Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, pp. 1–8,2018.

[180] Krishnan, R.B.; Raajan, N.R., "An intellectual intrusion detection system model for attacks classification using RNN" Int. J. Pharm. Technol. Pp. 23157–23164, 2016.

[181] Staudemeyer, R.C., "Applying long short-term memory recurrent neural networks to intrusion detection", S. Afr. Comput. J., pp. 136–154, 2015.

[182] R. Girshick., "Fast R-CNN", arXiv:1504.08083, 2015.

[183] D. Baysa, R. Low, and M. Stamp., "Structural entropy and metamorphic malware", Journal of Computer Virology and Hacking Techniques, vol. 9, no. 4, pp. 179–192, 2013.

[184] R. Lyda and J. Hamrock., "Using entropy analysis to find encrypted and packed malware", IEEE Security and Privacy, vol. 5, no. 2, pp. 40–45,2007.

[185] L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath. "Malware images: Visualization and automatic classification", In Proceedings of the 8th International Symposium on Visualization for Cyber Security, VizSec '11, pages 4:1–4:7, New York, NY,USA, ACM, 2011.

[186] Zhang, Jixin & Qin, Zheng & Yin, Hui & Ou, Lu & Hu, Yupeng., "IRMD: Malware Variant Detection Using Opcode ImageRecognition",pp.1175-1180, 2016.

[187] Mahotasfeatures,http://mahotas.readthedocs.org/en/latest/features.html, 2015.

[188] A. Moser, C. Kruegel, and E. Kirda., "Limits of static analysis for malware detection", In Computer Security Applications Conference, 2007. ACSAC 2007.Twenty-Third Annual, pp. 421–430, 2007.

[189] D. Bilar., "Statistical structures: Finger printing malware for classification and analysis", InBlackhat, 2006.

[190] B. Biggio, I. Corona, D. Maiorca, B. Nelson,N. A˘arndi¨A˘G, P. Laskov, G. Giacinto, and F. Roli., "Evasion attacks against machine learning at test time", In H. Blockeel, K. Kersting, S. Nijssen, andF. A¡elezn A¡, editors, Machine Learning and Knowledge Discovery in Databases, volume 8190 of Lecture Notes in Computer Science, pages 387–402, Springer Berlin Heidelberg, 2013.

[191] M. Christodorescu, S. Jha, S. Seshia, D. Song, and R. Bryant. "Semantics-aware malware detection", In Security and Privacy, 2005 IEEE Symposium on, pp. 32–46, 2005.

[192] B. B. Rao and K. Swathi, "Fast kNN classifiers for network intrusion detection system", Indian J. Sci. Technol., vol. 10, no. 14, pp. 1-10, 2017.

[193] H. Shapoorifard and P. Shamsinejad, "Intrusion detection using a novel hybrid method incorporating an improved KNN", Int. J. Comput. Appl., vol. 173, no. 1, pp. 5-9, 2017.

[194] S.Vishwakarma,V. Sharma, and A. Tiwari, "An intrusion detection system using KNN-ACO algorithm", Int. J. Comput. Appl., vol. 171, no. 10, pp. 18-23, 2017.

[195] E. G. Dada, "A hybridized SVM-kNN-pdAPSO approach to intrusion detection system", in Proc. Fac. Seminar Ser., pp. 14-21,2017.

[196] B. Ingre, A. Yadav, and A. K. Soni, ``Decision tree based intrusion detection system for NSL-KDD dataset,'' in Proc. Int. Conf. Inf. Commun. Technol. Intell. Syst., pp. 207-218,2017.

[197] A. J. Malik and F. A. Khan, "A hybrid technique using binary particle swarm optimization and decision tree pruning for network intrusion detection", Clust. Comput., vol. 2, no. 3, pp. 1-14, Jul. 2017.

[198] D. Moon, H. Im, I. Kim, and J. H. Park, "DTB-IDS: An intrusion detection system based on decision tree using behavior analysis for preventing APT attacks", J. Supercomput., vol. 73, no. 7, pp. 2881-2895, 2017.

[199] G. Zhao, C. Zhang, and L. Zheng, "Intrusion detection using deep belief network and probabilistic neural network", in Proc. IEEE Int. Conf. Comput. Sci. Eng., vol. 1, pp. 639-642,2017.

[200] Q. Tan, W. Huang, and Q. Li, "An intrusion detection method based on DBN in ad hoc networks", in Proc. Int. Conf. Wireless Commun. SensorNetw., pp.477-485, 2016.

[201] T.T.H. Le, J. Kim, and H. Kim, "An effective intrusion detection classifier using long short-term memory with gradient descent optimization", pp. 1-6,2017.

[202] A. F. Agarap., "A neural network architecture combining gated recurrent unit (GRU) and support vector machine (SVM) for intrusion detection in network traffic data.'' [Online], Available: https://arxiv.org/abs/1709.03082, 2017.

[203] J. Saxe and K. Berlin., "eXpose: A character-level convolutional neural network with embeddings for detecting malicious urls, file paths and registry keys., [Online]. Available: https://arxiv.org/abs/1702.08568, 2017.

[204] Ren, Shaoqing, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, pp.1137-114, 2015.

Mahendra Deore

M. Deore is working as an Asst. Professor in Computer Engineering Department at MKSSS's Cummins College of Engineering for Women, Pune 411051, India. He was awarded his Master of Technology Degree from Bharati Vidyapeeth Deemed University College of Engineering, Dhankawadi, Pune. He is currently pursuing Ph.D. Degree in Computer Science & Engineering from SGGS Institute of Engineering and Technology, Nanded under Swami Ramanand Teertha Marathwada University, Nanded, India. His areas of interest are big data, Security, Computer Networks and Machine learning. He has Fourteen years' experience in teaching.

Uday Kulkarni

U. Kulkarni is working as a Professor, Head in the Department of Computer Science and Engineering at SGGS Institute of Engineering and Technology, Nanded, India. He received doctoral degree from Swami Ramanand Teertha Marathwada University, Nanded, India in 2002. He is a recipient of a national level gold medal in the Computer Engineering Division for his research paper "Fuzzy Hyper sphere Neural Network Classifier" published in the journal of Institution of Engineers in 2004. He has published more than forty research papers in the field of Neural Networks, Fuzzy Logic and hybrid computing systems in the reputed journals and conferences.

Obtaining Anti-Missile Decoy Launch Solution From a Ship Using Machine Learning Techniques

Ramón Touza[1]*, Javier Martínez[2], María Álvarez[3], Javier Roca[2]

[1] Spanish Naval Academy, Pontevedra (Spain)
[2] Vigo University, Pontevedra (Spain)
[3] Defense University Center, Pontevedra (Spain)

Received 11 November 2020 | Accepted 2 March 2021 | Published 3 November 2021

Abstract

One of the most dangerous situations a warship may face is a missile attack launched from other ships, aircrafts, submarines or land. In addition, given the current scenario, it is not ruled out that a terrorist group may acquire missiles and use them against ships operating close to the coast, which increases their vulnerability due to the limited reaction time. One of the means the ship has for its defense are decoys, designed to deceive the enemy missile. However, for their use to be effective it is necessary to obtain, in a quick way, a valid launching solution. The purpose of this article is to design a methodology to solve the problem of decoy launching and to provide the ship immediately with the necessary data to make the firing decision. To solve the problem machine learning models (neural networks and support vector machines) and a set of training data obtained in simulations will be used. The performance measures obtained with the implementation of multilayer perceptron models allow the replacement of the current procedures based on tables and launching rules with machine learning algorithms that are more flexible and adaptable to a larger number of scenarios.

Keywords

Decoys, Machine Learning, Missile, Multilayer Perceptron, Support Vector Machine.

DOI: 10.9781/ijimai.2021.11.001

I. Introduction

The missile is one of the most dangerous threats a warship can face (see Fig. 1). Currently, about eighty countries have anti-ship missiles in their arsenals, which can be launched from aircraft, ships, submarines or from a coastal battery.

Fig. 1. USS Stark after the impact of two missiles launched from an Iranian aircraft in the Persian Gulf (Source: https://www.history.navy.mil).

Furthermore, given the current international scenario with an increase in global terrorism, it cannot be ruled out that anti-ship missiles may fall into the hands of some terrorist group and given that naval operations are more often conducted in coastal waters near the coast, this threat takes on special significance, as ships are exposed to attacks from land and with reduced reaction times.

In parallel with the change in the global threat, governments have increased their control over military forces deployed in areas of operation by dictating increasingly restrictive rules of engagement (ROE's) [1] and although every ship commander has an inherent right of self-defense, if an escalation of tension in the conflict zone is not desired, the use of force must always be proportional. Moreover, the need to avoid friendly confrontations makes it even more difficult to use the ship's weapons in combating a possible missile attack.

In addition to the foregoing, the Anti Surface Missile Defense (ASMD) is, from the tactical point of view, a complex action in which decisions must be taken in seconds, with no margin for error, and in which all means of defense may be employed: hardkill (long, medium and short-range missiles, naval guns and small arms) and softkill (passive and active electronic countermeasures and decoys).

The decoys are passive elements, which can be considered the last layer of defence of the ship. As shown in Fig. 2, the intention is to deceive the missile and focus its guidance system on a false target, this process, in naval tactical terminology, is known as seduction.

Another advantage of decoys is that their use does not involve the lethal use of force, which is always permitted by the rules of engagement.

* Corresponding author.

E-mail address: rtougil@fn.mde.es

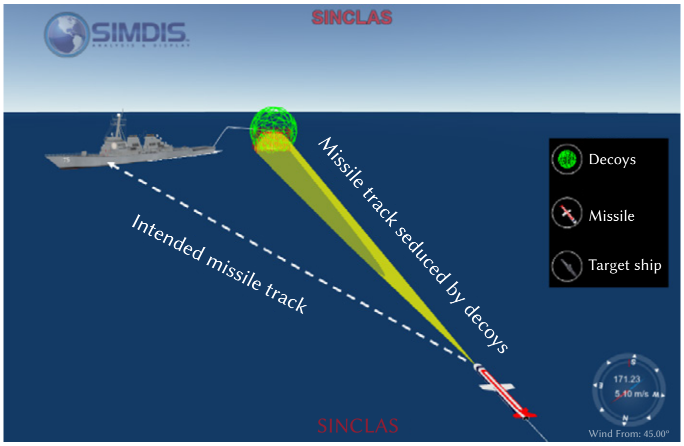


Fig. 2. Seducing a missile with decoys.

However, its use is not simple, to be effective and as shown in Fig. 3, the ship must react immediately, choose one of the 4 fixed launchers that are located in different positions and also must consider altering course to port or starboard to improve efficiency.

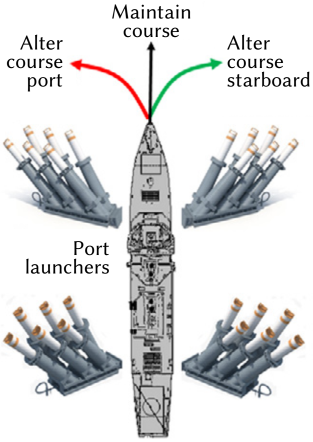


Fig. 3. Typical ship launchers configuration and possible maneuvers.

The launcher/alter course combination is what is known as the launch solution and therefore, for a scenario there are 12 possible solutions (4 launchers x 3 possible ship manoeuvres).

Currently, ships have implemented a series of pre-planned reactions, which they will use in the event of a missile attack. However, these reactions can only cover a limited number of scenarios.

Applying current technology of artificial intelligence, in particular machine learning techniques, it is possible to improve the accuracy of the reactions obtaining solutions for any scenario.

Moreover, for the same scenario there can be more than one launch solution, so within this last set it will be necessary to determine which is the best reaction to use. As an example, for two valid reactions, one may present a much greater miss distance of the missile from the ship than the other. In short, the feasible solutions for the same scenario can be ordered according to a series of criteria, allowing the ship to make a better decision.

Finally, as regards current anti-ship missiles, we can classify them, according to their guidance system, as: electro-optical (EO), infrared (IR), radio frequency (RF) or dual (combination of the above) and therefore, to defend against them, different types of decoys are used: flares for EO and IR guidance missiles, chaff for RF guidance missiles or a combination of the above for dual guidance missiles.

In this paper we will focus on the chaff launch solution for seducing radio frequency guidance missiles, although the methodology outlined here is applicable to any type of missile.

II. Objectives

The aim of this work is to develop a machine learning (ML) model which, trained from data from a number of simulations, obtains for any scenario what reactions of the ship allow to seduce the missile.

In addition, the ML model must have other data outputs: the expected missile miss distance, the time the ship is being tracked by the missile before it is focused on the decoy, and the probability of success of each possible launch solution. All these data can feed into a multi-criteria decision layer to obtain the best solution to the problem.

The basic architecture of the process would be as shown in Fig. 4, where a large number of scenarios would be simulated in the laboratory, to obtain a dataset with which to train a machine learning model, to be subsequently implemented on the ship and to obtain the best launch solution.

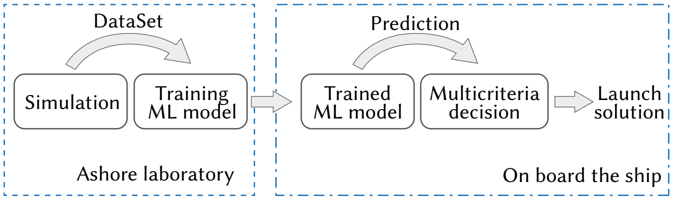


Fig. 4. Laboratory-on board process.

III. Simulation

In a real action against an attacking missile, the ship available time may not reach one minute, so the decoy launching solution must be pre-planned and there must also be a solution for all or most of the possible scenarios in which the ship may find itself.

The possible launch solutions are obtained by numerical simulation techniques before going to sea and are implemented on the ship by solution tables. This simulation is based on the interaction of three models: ship, decoy and missile, which interact in a scenario given by a wind, a ship speed and a distance and bearing of the attacking missile.

Each of the models involved in the simulation are characterized by a series of parameters or characteristics that will affect the launch solution.

The ship model is mainly characterized by its dimensions, turning circle maneuver, position of the decoy launchers and the radar cross section (RCS). The RCS is a random measurement that fluctuates rapidly over time and whose value determines the size of the echo that the ship presents on the radar of the attacking missile. Therefore a low RCS will make the vessel less visible to the missile seeker radar and will make it easier for the decoys to attract it.

The missile model is characterized by its speed and height of flight and certain parameters of its seeker radar.

On the other hand, the decoy model will be characterized by its deployment data: distance, height, deployment time, drop speed, cloud diameter and by its radar cross section. The decoy RCS is also a random variable and is an extremely important data, since it will have to have a sufficiently high value, in relation to the RCS of the ship to deceive the missile.

In addition to the three models mentioned, each of the possible simulation scenarios is given by the ship speed, the bearing (direction) and distance of the attacking missile and the wind (direction and intensity). The wind parameter is especially relevant since it will be responsible for dragging the cloud formed by the decoy away from the ship, which should foil the missile from its intended target.

The three models interact in the simulation in order to calculate, in each scenario, the launch solution that allows the seduction of the enemy missile. The seduction mechanism consists of deploying a cloud or clouds of decoys with sufficient RCS so that, due to the effect of the wind and the speed of the ship, the seeker radar resolution cell of the attacking missile will be centered on the cloud of decoys and away from the ship. This effect is called the centroid effect[2], which can be summarized in the phases shown in Fig. 5 [3]. Initially, in Phase A, the missile has the ship inside its seeker radar resolution cell and the ship reacts with a decoys launch. In Phase B, due to the effect of the wind, the decoys separate from the ship and the missile centers its resolution cell in the center of all the RCS (ship and decoys) contained in the cell. Finally, in Phase C the decoy continues to move away from the ship and if it has enough RCS, the transfer of the tracking from ship to decoy takes place.

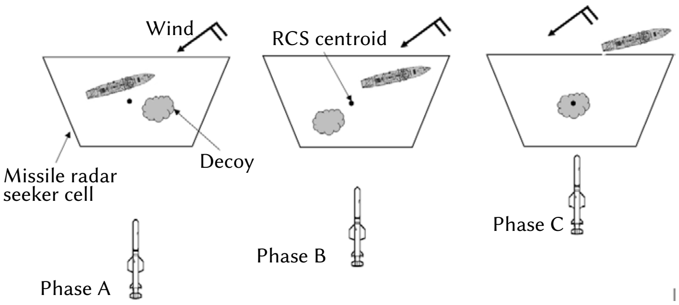


Fig. 5. Centroid effect.

The missile, ship and decoy models interact in the simulator as described for the centroid effect and it allows us to determine for each scenario which of the twelve ship possible reactions (combination of launcher and change of course) are effective in seducing the missile, i.e. success/failure. In addition, other data are obtained from the simulation, such as the minimum miss distance of the missile from the ship and the percentage of time that the missile has centered the ship in the seeker radar resolution cell before transferring the tracking to the decoy.

It should be noted that, as it is a stochastic simulator, a certain number of runs must be made in order to draw conclusions and therefore, for each possible solution, a probability of success can also be obtained.

In this work, a series of scenarios have been configured in the simulator with the data shown in Table I, giving a total of 186,624 scenarios. As there are 12 possible solutions per scenario, a total of 2,239,488 instances are obtained, of which 30 runs have been made. This provides a sufficient volume of data for training, validation and testing of any machine learning model.

TABLE I. Simulation Scenario Data

Scenario parameter	Values
Ship speed (S_{ship})	10, 15, 20 knots
Wind speed (S_{wind})	0, 10, 20 ,30 knots
Wind bearing (B_{wind})	[0°-355°] step 5°
Missile distance (D_{misil})	5000, 10000,15000 yards
Missile bearing (B_{misil})	[0°-355°] step 5°

Fig. 6 shows how the wind and missile bearing is measured.

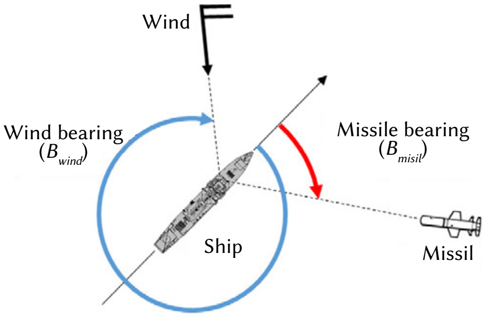


Fig. 6. Wind and missile bearing.

Before focusing on the types of ML models that can solve the problem, it is necessary to clarify the available dataset obtained from the simulations. The ML model should have as inputs the parameters of the scenario in which the vessel is, the available launchers (4 launchers in our case) and the possible alter course. Thus, the input data will be as shown in Table II.

TABLE II. Input Data

Data	Units
Ship speed (S_{ship})	knots
Wind speed (S_{wind})	knots
Wind relative bearing (B_{wind})	degrees
Missile distance (D_{misil})	yards
Missile relative bearing (B_{misil})	degrees
Launcher (L)	1, 2, 3, 4 (binary coded)
Alter course (A)	Port, 0°, starboard (binary coded)

From the input data, for each scenario and for each possible launch solution, we will have the following Target data (see Table III).

TABLE III. Target Data

DATA	UNITS
Solution	1 – success (no impact on ship); 0-fail (impact on ship)
Miss distance	Yards
Time in cell	Total engagement percentage
Probability of success (Note)	1-Probability of success >0.8 0- Probability of success ≤0.8

Note: 0.8 is taken as the probability of success since it corresponds to the turning point of the probability/frequency histogram and encompasses 87% of the total solutions.

IV. Machine Learning Models

As mentioned above, the aim of this work is to build a machine learning model that, trained from the data obtained from the simulation, can be implemented in a ship's combat system so that it will provide the necessary outputs to determine the best decoy launching solution.

Since we have four types of target data, it was decided to build four different ML models based on supervised learning: two binary classification models for solution and probability of success, and two regression models for miss distance and time in cell.

Initially, two types of machine learning techniques were chosen as possible candidates to solve the problem: Support Vector Machine (SVM) and Multi Layer Perceptron (MLP). Thus, this study tries to determine which is the optimal technique or which combination of both techniques is the optimal one to solve the problem.

A. Multilayer Perceptron (MLP) Model.

The term neural network has its origin in attempts to find mathematical representations of information processing in biological systems. It is a powerful structure that allows the creation of non-linear predictive models and is used in supervised and unsupervised learning[4]. In the case of supervised learning it allows to build binary and multiclass regression and classification networks [5]–[7].

The basic structure of a neural network is the neuron (see Fig. 7), the input variables (x_i) are connected to the neuron through weighted connections (w_i) that emulate dendrites, while the sum (Σ), the bias (b) and the activation function (h) play the role of the cell body and the propagation of the output is analogous to the axon in a biological neuron. The behavior of the neural network is defined by the shape of the connections of its neurons or nodes and by the values of the weights of these connections. These weights are automatically adjusted during training according to a learning algorithm, until the network carries out the desired task correctly [8].

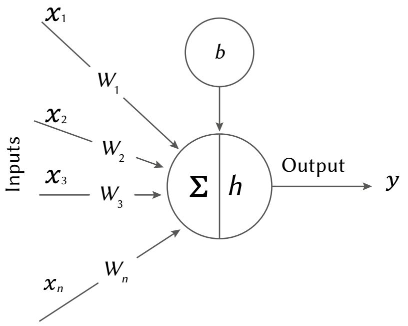


Fig. 7. Neuron model.

Multilayer perception (MLP) or multilayer network with feed-forward architecture, without feedback, is a kind of neural network, which has an architecture with a finite number of layers and neurons. In this structure three different types of layers are distinguished: the input layer, the hidden layers and the output layer, the latter having as many outputs as targets to be predicted. Generally, all the neurons in one layer are connected to all the neurons in the next layer, which is known as total connectivity or fully connected network.

The MLP characterizes the relationship between input and output layers, which is parameterized by the weights. This relationship is obtained by propagating the values of the input variables forward. To do this, each neuron in the network processes the information received by its inputs and produces a response or activation that propagates, through the corresponding connections, to the neurons in the next layer.

If we have an MLP with D inputs, and there are L layers with n_l neurons each one, the first $L-1$ layers will be the hidden layers and then there is the output layer, with so many outputs K as dimensions have the target to predict. The output of the neuron i of the layer l we will denote it by z_i^l. From one layer to the next, the output of the neuron i is weighted by a weight w_{ij}^l where l is the layer and j is the neuron of the layer l (see Fig. 8).

The weighted sum of the weights of a neuron input is called activation. Thus, for the neuron j of the layer l, the activation a_j^l is given by:

$$a_j^l = \sum_{i=0}^{n_{l-1}} w_{ji}^l \, z_i^{l-1}$$

This input is transformed using an activation function $h^{(l)}(\cdot)$ to generate the output z_j^l to the next layer:

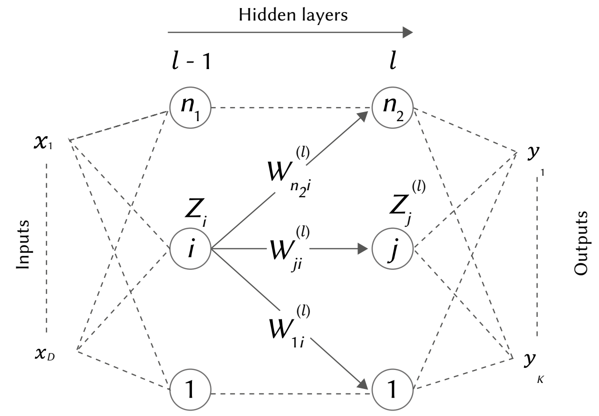


Fig. 8. Nomenclature used in the mathematical definition of neural networks.

$$z_j^l = h^{(l)}\left(a_j^{(l)}\right) = h^{(l)}\left(\sum_{i=0}^{n_{l-1}} w_{ji}^l \, z_i^{l-1}\right)$$

The activation functions for the output layer in our problem have been chosen according to the type of prediction. Thus, the softmax function was used for binary classification and the identity or purelin function for regression. For the hidden layers, the tansig function was used in all cases.

Now, from the training dataset given by some entries $\{\boldsymbol{x_n}\}$ and a target $\{\boldsymbol{t_n}\}$, the training objective is to find a set of weights $\boldsymbol{w}$ that minimizes the error function that we will denote $J_n(\boldsymbol{w})$ which measures the error associated with the training sample. The sum of squares function, below is a common choice:

$$J_n(\boldsymbol{w}) = \sum_{n=1}^{D} \|\boldsymbol{y}(\boldsymbol{x_n}, \boldsymbol{w}) - \boldsymbol{t_n}\|^2$$

To solve this problem, heuristic methods based on the descent of the gradient called backpropagation are used. The steps to carry out this algorithm are the following:

Step#1. Apply the inputs $\boldsymbol{x_n}$ to the MLP, propagate forward, calculate the outputs $\boldsymbol{y_k}$ and errors in the output layer $\boldsymbol{\delta_k = y_k - t_k}$

Step#2. Errors are propagated backwards, calculating for all hidden layers. So that, the error of a certain layer, is calculated on the basis of the error of the following layer:

$$\delta_j = h'(a_j) \sum_k w_{kj} \, \delta_k$$

Where the term δ_j is called error and must be calculated for each network node.

Step#3. Evaluation of error-function. Derivatives are calculated as:

$$\frac{\partial J_n}{\partial w_{ji}} = \delta_j z_i$$

Step#4. From the derivatives, bayesian regularization backpropagation algorithm is applied that progressively improves the weights of the network [9]–[12].

An important aspect to consider when designing a multilayer perceptron is the number of layers and neurons per layer. More than one hidden layer speeds up the training process, especially in very heavy problems. On the other hand, the more neurons per layer, the better the adaptability of the model to the problem. However, an increase in the number of layers and neurons can lead to overfitting and poor generalization [13], [14].

For this study, we have chosen to start with a single hidden layer network, and increase the number of layers while decreasing the error. In the same way, the number of nodes or neurons in the hidden layers has been increased until the error is stabilized. In the case of configurations with more than one hidden layer, the pyramid technique has been used, which consists of decreasing the number of nodes from the input layer to the output layer [15].

Therefore, for the problem of this study, first of all, it is necessary to determine which the optimal configuration of layers and neurons per layer is for each of the four target data (solution, miss distance, time in cell and probability of success), which will be given by the one that presents the best measure of performance with the least number of layers and neurons.

In order to determine the optimal configuration, it was decided to analyze three network models, each with one, two and three hidden layers. In addition, different numbers of neurons were configured in each layer in each model.

It was found that with four hidden layers the performance measures deteriorated significantly and the MLP models suffered early stops due to overfitting.

The different configurations analyzed are presented in Table IV.

TABLE IV. MLP Configurations Analyzed

	Number of neurons per layer		
Hidden layers	Layer 1	Layer 2	Layer 3
1	From 10 to 150 (step 10)	-	-
2	From 10 to 50 (step 5)	5 fewer neurons than in layer 1	-
3	From 10 to 50 (step 5)	5 fewer neurons than in layer 1	5 fewer neurons than in layer 2

B. Support Vector Machine (SVM) Model

Support Vector Machines (SVM) are part of the supervised learning techniques and are used for both classification and regression. SVMs belong to the family of linear classifications and they find a linear separator or hyperplane [16].

In the classification SVM, the optimal separator hyperplane is defined as the maximum margin separating hyperplane that maximizes its distance from the classes (see Fig. 9).

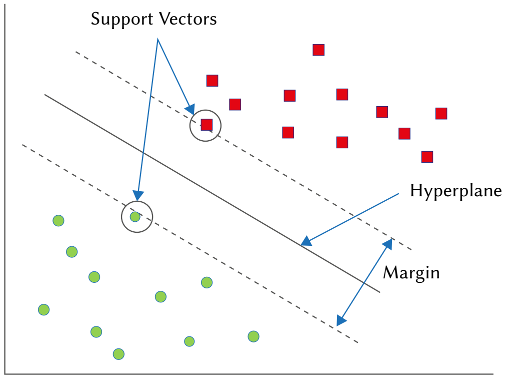


Fig. 9. Maximum margin separating hyperplane for a classification problem with two linearly separable classes.

The concept with which the maximum margin SVM works, is to find the hyperplane separator that is the same distance from the closest examples of each class. Equally, it is the hyperplane that maximizes the minimum distance between the examples of the data set and the hyperplane. Furthermore, it only considers the points that are on the borders of the decision region, these data are the so-called support vectors.

If we have a training dataset $S = \{(x_1, y_1), \dots (x_n, y_n)\}$ where $\boldsymbol{x_i} \in \mathbb{R}^D$ and $y_i \in \{-1, 1\}$, In the case of a linear function, the separating hyperplane will be a linear function:

$$y(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x} + b$$

Where $\boldsymbol{w} \in \mathbb{R}^D$ is the weight vector orthogonal to the hyperplane and $b \in \mathbb{R}$.

However, the most common formulation of the linear SVM and maximum margin is its primal formulation which is as follows, and corresponds to the quadratic optimization problem whose objective function and the corresponding constrains are:

$$\min_{\boldsymbol{w},b} \frac{1}{2} \|\boldsymbol{w}\|^2$$

s.t.

$$y_i(\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b) \geq 1, i = 1,2, \dots n$$

Separating hyperplanes have two major weaknesses: the requirement for linear separability of the sample and its linear nature. Therefore, in order to extend the SVM concept to non-linear classifiers, a transformation of the input space is performed, using kernel functions $(\Phi(\boldsymbol{x}))$, to another high dimensional feature space in which the data are linearly separable, this procedure is known as the kernel trick [17], [18]. The separation hyperplane then takes the following form:

$$y(\boldsymbol{x}) = \boldsymbol{w}^T \Phi(\boldsymbol{x}) + b$$

Likewise, for problems whose training data set is not linearly separable, the Soft-Margin algorithm can be used, which introduces slack variables $(\xi_i \geq 0)$ to relax the condition of the margin, allowing poorly classified observations and making the model more robust [16].

We thus obtain the following optimization problem with objective function and constrains:

$$\min_{\boldsymbol{w},\xi} \left(\frac{1}{2} \|\boldsymbol{w}\|^2 + C \sum_i \xi_i \right)$$

s.t.

$$y_i(\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b) \geq 1 - \xi_i, \qquad \xi_i \geq 0, i = 1,2, \dots n$$

Where the regularization parameter (C) indicates the importance or cost of misclassified instances.

SVM can be applied to regression problems by introducing an alternative loss function. In this work the loss function $\varepsilon -$ *insentitive* proposed by Vapnik has been used [19], [20], which ignores errors at a certain distance from the real value. In this regression SVM algorithm, some slack variables are also introduced (ξ, ξ^*) which measure the cost of errors in training points, the values of these variables being zero for all points within the band ±ε (see Fig. 10).

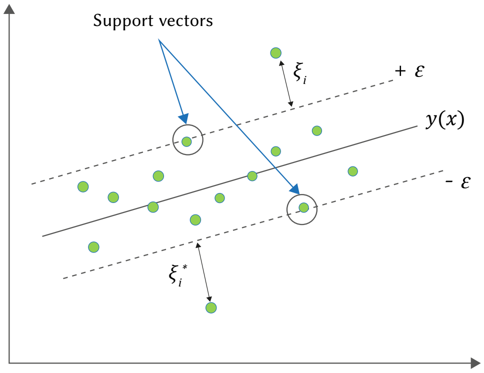


Fig. 10. Regression hyperplane, support vectors, slack variables and $\varepsilon -$ *insentitive* loss function.

In the case of regression SVM, the quality of the estimate will be given by the loss function (L):

$$L\big(\boldsymbol{y}, f(\boldsymbol{x},\boldsymbol{\omega})\big) = \begin{cases} 0 & if\ \ |y_i - f(\boldsymbol{x},\boldsymbol{w})| \le \varepsilon \\ |y_i - f(\boldsymbol{x},\boldsymbol{w})| - \varepsilon & otherwise \end{cases}$$

Thus, the optimal regression SVM hyperplane will be given by the following minimization problem:

$$\min_{w,\xi,\xi^*} \left(\frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_i (\xi_i^* + \xi_i)\right)$$

s.t.

$$y_i - f\big(x_i, \omega\big) \le \varepsilon + \xi_i^*$$

$$f\big(x_i, \omega\big) - y_i \le \varepsilon + \xi_i$$

$$\xi_i, \xi_i^* \ge 0, i = 1, \dots n$$

In the same way, in the case of non-linear training data, we will have to perform a non-linear transformation in the input space, to a high dimensional feature space by means of a kernel function.

In our problem, we have used classification SVM with a Soft-Margin algorithm, for the regression SVM models we have used a $\varepsilon - insentitive$ loss function and for both we have implemented a radial base or Gaussian kernel with gamma parameter ($\gamma > 0$).

Therefore, in each case, we must obtain the optimal values for: kernel scale (γ), boxconstrain (C) and $\varepsilon - insentitive$. For this purpose, the grip method or grid search has been used.

C. Finding the Optimal Model

To determine the optimal model for each of the four targets, the following steps were followed:

Step#1. Dataset normalization.

Step#2. Setting the MPL parameters and SVM hyper parameter ranges.

Step#3. Training, validation and testing of all MLP models contained in Table IV and SVM.

The same training and test data were used for this step, for all MLP and SVM configurations. In addition, a total of 10 runs were made, and a statistical study was subsequently carried out to compare the results. In this step, and to save computing time, from the total number of available instances and for each replica, 56 000 instances were sampled.

Step#4. Collection of performance measurements.

For the classification models (solution and probability of success): accuracy (matrix confusion).

For regression models (miss distance and time on cell): mean square error (mse) and R^2.

Step#5. Graphic and statistical analysis of performance measures.

Step#6. Identification of the optimal model for each target.

Step#7. Train the optimal models with a greater number of instances to improve performance measures. A total of 500,000 instances of total available dataset were used.

Step#8. Build the model set to obtain the predictions of the launch solutions.

By applying the Steps# 1-5, the following results have been obtained for each of the four target data:

a) Target: Solution (classification model). In this case and after analyzing the confusion matrices of the different models, the means of the accuracy values obtained are presented in Fig. 11. It can be observed that for the case of MLP with 2 and 3 hidden layers, precision values higher than 0.93 have been obtained.

Furthermore, with 95% confidence, no statistically significant difference was found between MLP with 2 and 3 hidden layers. Therefore, a 2-hidden-layer MLP with 40 and 35 neurons in each layer was taken as the optimal model for this target.

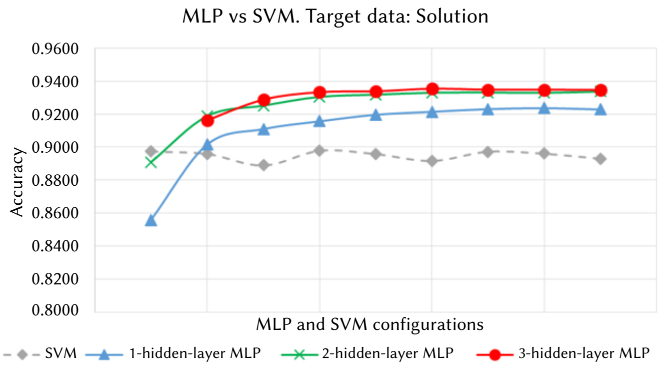


Fig. 11. Solution. Comparative of the accuracy means for the different models.

b) Target: Miss distance (regression model). In this case the value of mean square error and R^2 were used as measures of performance. In Fig. 12, it can be seen that mse minimum value has been reached for a 3-hidden-layer MLP. Furthermore, it was found that statistically and with 95% confidence, the mse values for 3-hidden-layer MLP are statistically lower than the 2-hidden-layer models. Therefore, we will take the 3-hidden-layer MLP with 35-30-25 neurons per layer, as the optimal model for the target data miss distance.

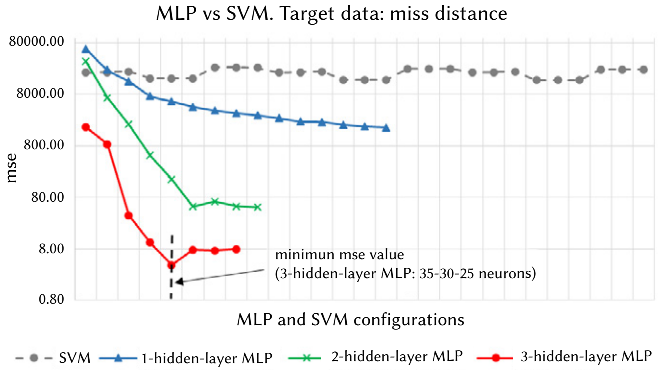


Fig. 12. Miss distance. Comparative of mse means for the different models.

c) Target: % Time in cell (regression model). In this case the value of mean square error and R^2 also were used as measures of performance.

After training the models very high values of mse (higher than 0.001) and poor values of R^2 (lower than 0.80) were obtained.

To improve the results the miss distance was introduced as an input and all models were retrained. With this, the results improved significantly, decreasing the mse by approximately 40% and now obtaining values of R^2 higher than 0.88.

On the other hand, once the graph in Fig. 13 has been created, it can be seen that the best mse results were obtained for 3-hidden layer MLP, specifically for a configuration of 35-30-25 neurons per layer.

d) Target: Probability of success >0.8 (classification model). In this case the same problem was found as for the previous target, the calculated accuracy values in the confusion matrices for the different configurations were lower than 80%, so they cannot be considered as valid for our problem.

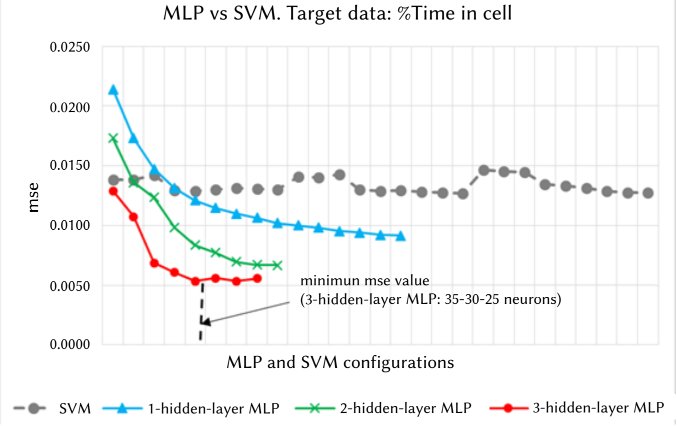


Fig. 13. % Time in cell. Comparative of mse means for the different models.

To improve the results it was introduced as miss distance as a new model input, improving the results reaching precision values close to 90%.

As can be seen in the graph in Fig. 14, all models achieve similar accuracy values, proving that statistically and with 95% confidence, there are no differences between them. Therefore, a 1-hidden-layer MLP of 60 neurons was taken as the optimal model.

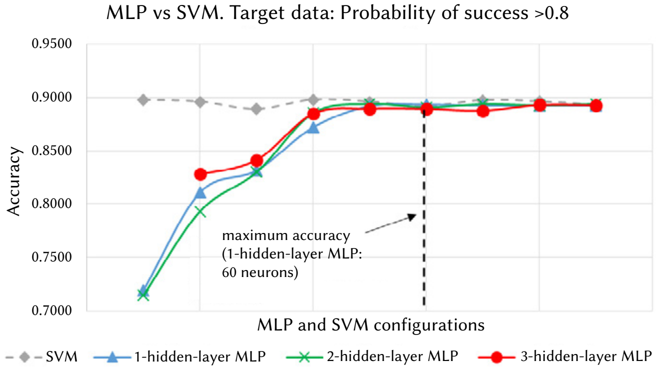


Fig. 14. Probability of success >0.8. Comparative of the accuracy means for the different models.

D. Training, Building the Models Set and Test

Once the optimal models had been found, the following steps were taken as shown in Fig. 15: the optimal models were trained, the final set of models was built and the performance measures of this set were calculated.

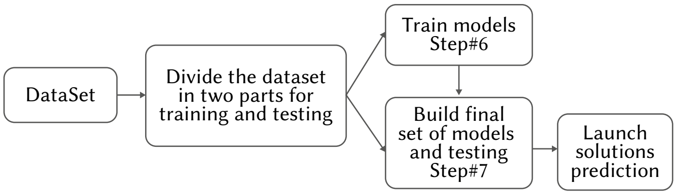

DataSet
Divide the dataset in two parts for training and testing
Train models Step#6
Build final set of models and testing Step#7
Launch solutions prediction

Fig. 15. Training, building the models set and test.

As mentioned in the description of step 3, to save computer time to find the optimal models, small samples of the dataset were used. Therefore, the next step was to train the optimal models found in the previous step with a greater number of instances to improve the performance measurements. For this purpose, we reserved a total of 500,000 instances for models training and the remaining part of the dataset for test the final set of models.

As mentioned above, the time in cell model and the probability of success model were trained by adding the miss target as an input. However, this data is not available in a real scenario, so it was necessary to build a set of MLPs linked between them, so that the output of the model corresponding to miss distance will be part of the input of the time in cell and probability of success models (see Fig. 16).

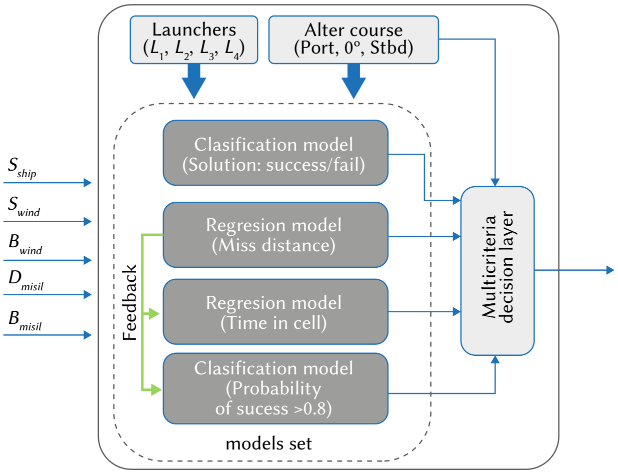


Fig. 16. Final solution. MLP models set.

As there is feedback from the output of the miss distance model to the inputs of others, there is inevitably a propagation of errors, so it is necessary to determine what the values of the performance measures are once the complete set of models have been built. For this purpose, the dataset sample reserved for the MLP set test was used, the test values are contained in Table V.

TABLE V. MLP Models Set Final Performance Measures

Target data	Performance measure
Solution	Accuracy: 96.70%
Miss distance	$mse = 1.42\ yards^2$; $R^2 > 0.99$
% Time in cell	$mse = 0.0043$; $R^2 = 0.91$
Probability of success >0.8	Accuracy: 93.82%

V. Conclusions and Future Work

The set of machine learning models based on neural networks developed in this study can be implemented on any ship and improve the decoy launching solution for any scenario and offer the operator a more evaluated launch response. This only requires the training of the models with the specific data of each ship. In other words, the decoy launch simulator can be substituted by a set of trained MLP and provide real time response to the launch problem.

By having in real time the values of miss distance, % time in cell, probability of success and the need or not for a change of course , they can be used as criteria in a multi-criteria decision algorithm (e.g. Analytic Hierarchy Process -AHP-) to obtain the best possible solution. In other words, following the set of MLP models, a solution manager could be implemented that would automatically evaluate all possible solutions for a scenario without operator intervention.

On the other hand, the model developed is scalable, it is possible to implement on board as many sets of models as there are threat missiles in the operations area and thus have a solution for each of them immediately.

It should also be noted that the work has focused on defense against radio frequency guided missiles, although the methodology proposed here may be applicable to infrared or dual guided threat missiles.

VI. Discussion

The implementation of machine learning models, to solve the problem of decoy launching, opens the way for the introduction of different artificial intelligence techniques in other ship systems, even in those situations where it is necessary to make tactical decisions.

Combined use of naval simulation and AI/machine learning techniques allows us to build models that improve and automatize the decision process on board, which can result in an improvement in the ship survivability and be one of the levers that help to achieve superiority in combat. On the other hand, it is also feasible to automate on-board operational processes, which would make it possible to reduce the required number of crew members and save cost.

References

[1] R. Lord, "Advances in Anti Ship Missile Protection - Naval Countermeasures," Chemring Naval Countermeasures. Salisbure, England, 2006.

[2] W. Sun, "Maneuvering Calculation of Ship Centroid Jamming," in *9th International Conference on Information and Social Science*, 2019, pp. 386–390.

[3] L. F. Galle, "Royal Netherlands Navy The Survivable Frigate," *1o Eur. Surviv. Work.*, 2002.

[4] N. Manju, B. S. Harish, and N. Nagadarshan, "Multilayer Feedforward Neural Network for Internet Traffic Classification," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 6, no. 1, p. 117, 2020.

[5] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The Bulletin of Mathematical Biophysics*, vol. 5, pp. 115–133, 1943.

[6] B. Widrow and M. E. Hoff, "Adaptive switching circuits," *In IRE WESCON Convention Record, Volume 4, pp. 96–104.* 1960.

[7] F. Rosenblatt, *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms.* Washington DC: Spartan Books, 1962.

[8] M. Award and R. Khanna, *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers.* Apress Media, LLC, 2015.

[9] D. J. C. Mackay, "A Practical Bayesian Framework for Backprop Networks," *Neural Comput.*, vol. 4, no. 3, pp. 448–472, 1992.

[10] F. D. Foresee and M. T. Hagan, "Gauss-Newton approximation to bayesian learning," *Proc. Int. Conf. Neural Networks (ICNN'97), Houston, TX, USA*, vol. 3, pp. 1930–1935, 1997.

[11] A. Suliman and B. Omarov, "Applying Bayesian Regularization for Acceleration of Levenberg Marquardt based Neural Network Training," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 5, no. 1, p. 68-72, 2018.

[12] K. K. Aggarwal, Y. Singh, P. Chandra, and M. Puri, "Bayesian Regularization in a Neural Network Model to Estimate Lines of Code Using Function Points," *Journal of Computer Science*, vol. 1, no. 4, pp. 505–509, 2005.

[13] R.Reed and R. J. Marksll, *Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks.* Cambridge, Massachusetts, 1999.

[14] J. Heaton, *Introduction to Neural Networks for Java*, 2nd Editio. Heaton Research, Inc., 2008.

[15] D. Stathakis, "How many hidden layers and nodes?," *International Journal of Remote Sensing*, vol. 30, no. 8, pp. 2133–2147, 2009.

[16] C. Cortes and V. Vapnik, "Support-Vector Networks," Kluwer Academic Publishers, 1995.

[17] C. M. Bishop, *Pattern recognition and machine learning.* Springer, 2006.

[18] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the fifth annual workshop on computatinal learning theory*, 1992, pp. 144–152.

[19] H. Drucker, C. J. C. Surges, L. Kaufman, A. Smola, and V. Vapnik, "Support vector regression machines," *Advances in Neural Information Processing Systems*, no. June 2013, pp. 155–161, 1997.

[20] V. Cherkassky and Y. Ma, "Practical selection of SVM parameters and noise estimation for SVM regression," *Neural Networks*, vol. 17, no. 1, pp. 113–126, 2004.

Ramón Touza Gil

Cmdr. R. Touza is an Officer in the Spanish Navy and a professor at the Spanish Naval Academy. He has received his Master degree in Advanced Data Analysis and Model Building from Complutense University of Madrid and Master in Decision Engineering from Juan Carlos University of Madrid. He is currently a PhD student at Polytechnic University of Cartagena (Spain).

Javier Martínez Torres

Dr. Javier Martínez is a Mathematician and Engineering PhD from the University of Vigo. He is currently an Assistant Professor at the University of Vigo and has participated in more than 20 research projects as principal investigator. He has published more than 50 papers in JCR indexed journals and participate in more than 25 international conferences.

María Álvarez Hernández

She is an associate professor in the Defense University Center – ENM (attached center of the University of Vigo), with a degree in Mathematics from University of Salamanca and PhD. in Statistics and Operations Research for the University of Granada. She has been part of several national projects in Spain, related to the development and research in the statistical field and belongs to the National Network of Biostatistics. The results of her research have been disseminated through a score of papers through indexed scientific journals (within the area of statistics and applied scientific journals) and presentation and participation in dozens national and international scientific conferences.

Javier Roca Pardiñas

Dr. Javier Roca is Associated Professor of the Department of Statistics and Operational Research of University of Vigo. He is specialized in nonparametric regression, and bootstrap inferences techniques with special emphasis on generalized additive models (GAM). Moreover, he is an expert in computational statistics, with important contributions to semiparametric prediction models. His research work generated more than forty publications in international journals of impact in different areas of knowledge such as statistics, computer science, environment, biomedicine, and engineering amongst others. He has also successfully performed as a member of four research projects funded by Spanish Ministry of Science and Innovation devoted to the development of basic and applied mathematical knowledge. He has been principal investigator on several projects with numerous companies and institutions.

Writing Order Recovery in Complex and Long Static Handwriting

Moises Diaz[1,2]*, Gioele Crispo[3], Antonio Parziale[3], Angelo Marcelli[3], Miguel A. Ferrer[1]

[1] Instituto Universitario para el Desarrollo Tecnológico y la Innovación en Comunicaciones, Universidad de Las Palmas de Gran Canaria, Las Palmas 35017 (Spain)
[2] Universidad del Atlántico Medio, Las Palmas de Gran Canaria, Las Palmas 35017 (Spain)
[3] DIEM, University of Salerno, 84084 Fisciano (SA) (Italy)

Received 22 October 2020 | Accepted 4 March 2021 | Published 19 April 2021

unir
LA UNIVERSIDAD
EN INTERNET

Abstract

The order in which the trajectory is executed is a powerful source of information for recognizers. However, there is still no general approach for recovering the trajectory of complex and long handwriting from static images. Complex specimens can result in multiple pen-downs and in a high number of trajectory crossings yielding agglomerations of pixels (also known as clusters). While the scientific literature describes a wide range of approaches for recovering the writing order in handwriting, these approaches nevertheless lack a common evaluation metric. In this paper, we introduce a new system to estimate the order recovery of thinned static trajectories, which allows to effectively resolve the clusters and select the order of the executed pen-downs. We evaluate how knowing the starting points of the pen-downs affects the quality of the recovered writing. Once the stability and sensitivity of the system is analyzed, we describe a series of experiments with three publicly available databases, showing competitive results in all cases. We expect the proposed system, whose code is made publicly available to the research community, to reduce potential confusion when the order of complex trajectories are recovered, and this will in turn make the trajectories recovered to be viable for further applications, such as velocity estimation.

Keywords

Cluster Resolution, Complex and Long Handwriting, Good Continuity Criteria, Writing Order Recovery.

DOI: 10.9781/ijimai.2021.04.003

I. Introduction

Over the last 40 years, handwriting analysis and recognition have been widely studied, and many theoretical and experimental results have been obtained both on handwriting acquired with tablets (on-line samples) and on that obtained with scanners (off-line samples) [1], [2]. These studies have contributed to many useful applications, such as mail sorting, form processing and handwriting recognition. More recently, the widespread use of devices such as smartphones, tablets, and electronic pen pads has given rise to a personal digital bodyguard concept [3]. This feature can supplement data protection, which enhances human-machine interactions through handwriting recognition.

In addition to such technological advances, the automatic processing of off-line handwriting is still of great interest in many fields of application, such as enterprise management [4],[5], education [6]–[8], and healthcare [9], [10]. Both public offices and private companies need to archive and retrieve digital versions of documents that contain handwritten samples. Recent years have witnessed the rise of many digital libraries that require systems that allow automatic searches in transcripts and ancient manuscripts [11].

* Corresponding author.
E-mail address: moises.diaz@ulpgc.es

It has long been understood that on-line handwriting systems perform much better than off-line ones because the former have access to dynamic information. This information consists mainly of the order in which the trajectory was written and its velocity profile. This limitation has motivated the development of systems for writing order recovery in static images [12]–[14]. These systems include two steps: 1) thinning of the handwriting, and 2) writing order recovering of the trajectory. Computer-based systems for recovering static trajectories have been proposed for several handwritten applications, including the reading of cursive handwriting [15], Latin and Arabic handwriting word recognition [16]–[18], Indian and Chinese character recognition [19], [20], digit recognition [21], historical document transcription [22], mathematical symbol recognition [23], signature verification [14], [24], [25], writer verification and identification [26], [27], handwriting style modeling [28], [29] and handwriting analysis and synthesis [30].

Recovering the trajectory in static handwriting may lead to multiple possible solutions, with only one representing the real trajectory sought. Solving this inverse problem becomes even more complicated when the handwritten pattern contained in the image comprises multiple components;[1] in this case, recovering a new component may depend on the order of the previous one and the in-air trajectory between two consecutive components.

[1] By "component", we refer to a piece of writing with two end-points, meaning that it is performed between two pen-ups. It can be denoted as a pen-down as well.

A. Our Contribution

This paper proposes a novel system for estimating the writing order recovering in complex and long static handwriting consisting of many components, separated by pen-ups, also named in-air trajectories. Inspired by human movements during the production of handwriting and by motor control perspectives [31], our system chooses the smoothest ballistic trajectories based on good continuity criteria when writing or drawing. To this end, we apply some multiscale strategies to recover the trajectory order in the agglomeration of pixels, which we call clusters. Additionally, we study the proposed system's performance, both when the end-points of the components are known and when they are unknown. We provide a quantitative measure of how heuristics on the choice of starting and ending points affect the writing order recovery. The effectiveness of the system is demonstrated in three databases, achieving competitive performance.

We focus our work on static specimens for two reasons: the first one is that developing a system that recovers complex, long, and discontinuous trajectories represent the most challenging case, and the second is that estimating the writing order from static images is of great interest for the early prescreening of neurodegenerative disorders [10], [32].

The code is developed in Matlab, and it is freely provided for research[2] .

The outline of the paper is as follows: Section II presents a brief overview of related works on writing order recovery of trajectories in static handwritten specimens. Section III describes the proposed solution for estimating handwriting recovering. A sensitivity and stability study of parameters is given in Section IV, whereas experimental results are presented in Section V. Finally, conclusions are drawn in Section VI.

II. Literature Review

Over the last thirty years, many systems have been proposed for recovering static trajectories, with two surveys assessing the related state of the art [12], [13]. Moreover, a competition has been considered in order to establish a common benchmark for writing order recovery of Arabic signatures [14], [33].

Systems used to recover trajectories from static handwriting employ three main approaches: contour-based, skeleton-based, and learning-based. The first two differ in terms of the "object" used to represent the handwriting, respectively its contour and its skeleton. The skeleton is the result of a thinning process that produces a 1-pixel-wide line, which follows the centerline of the original image, and that ideally corresponds to the original pen-tip trajectory.

Regarding contour-based approaches, [34] describes a method that segments cursive handwriting by detecting the points where the trace contour has the maximum curvature. The segments are ordered based on the contour curvature smoothness. Another handwriting segmentation is given in [16], and proceeds through an analysis of the handwritten contour. Here, the writing order trajectory is estimated by adopting graph-based representations at both the segment and stroke levels. The list of candidate paths is obtained by choosing curvature and width stroke preservation, which are local continuity criteria, as cost functions. Following up on [16], the system was employed in [17] to develop an off-line recognition system. In that work, a Hidden Markov Model (HMM) was trained for selecting the most likely written trajectory from the list of candidate paths. To avoid introducing artifacts in the handwriting, preprocessing steps, such as binarization, were omitted in [35]. Instead, the authors extracted control points from the grayscale images, which were used to recover the trajectory according to certain heuristics rules.

The skeleton-based methods can be categorized as local line order recovering and global graph searching methods. The former reconstructs the trajectory by choosing the most plausible direction at each branch point of the skeleton. These methods are simple and have a low computational cost, but their performances are typically limited by the difficulty of designing heuristic rules for different handwriting styles. Instead, graph searching methods recover the trajectory by representing the topological structure of the skeleton with a graph and traversing it. They have a greater computational cost and their performances depend on the definition of criteria for selecting the best trajectory among many alternatives.

An example of local line recovering is given in [36], where the trajectories are recovered according to good continuity criteria, which take into account the direction, length, and width of the strokes making up characters. Off-line signatures are recovered in [37] by following heuristic rules inspired by the way human beings write a given shape. The rules are applied to deal with low-level pixel processing and high-level stroke processing. A similar approach is applied to handwritten digits in [38]. In [39], a taxonomy of local, regional, and global features that can be used for recovering temporal properties from the image is proposed. On the other hand, a likelihood measure is developed in [40]. It selects the most likely writing order recovery from the analysis of skeletons. In an on-line automatic signature verification system, the method proposed in [40] shows a greater false acceptance rate in skilled forgeries than in random forgeries. A genetic algorithm is also used for recovering segments extracted from the skeleton [41]. The fitness function adopted by the genetic algorithm for selecting the best individual of the population takes the writing direction, the repetition of segments, and the angular deviation on the crossing of the occlusion stroke into account. Additionally, a complete framework to recover the dynamic properties (i.e., velocity and pressure) from an image-based signature is presented in [24] by using classical approaches to recover the static trajectories in the signatures.

As for global graph methods, they represent the topological structure of the skeleton with a graph whose vertices represent the end-points, the junctions and the contact points of the skeleton, and whose arcs represent lines and curves. These methods determine the writing order recovery by finding the most appropriate path along the graph. In [42], one of these methods is proposed for recovering the trajectory of words through the traveling salesman problem. Based on the skeleton of words, the authors search for trajectories with minimum curvatures. Also, in [43], each component of a word is represented by a graph. The trajectory of the whole word is obtained by concatenating, from left to right, the most likely trajectory of each component. The authors define both global and local criteria that allow to select the best trajectory. The path search was based on the best-first search algorithm The authors in [44] propose to construct a graph to use to determine the types of each edge from the skeletons. Then, they develop a writing order recovery algorithm to traverse such a graph without applying any graph search algorithm. The goal of the method proposed in [45] is to use as little heuristic knowledge as possible. To this end, the proposal applies the maximum weighted matching of general graphs to find double-written lines. It exploits the minimum energy cost criterion as a guiding principle for recovering the trajectories. Successful results in single-stroke images are obtained.

A graph transformation is proposed in [46] to ensure that all graphs' nodes had an even number of incident arcs. This property allows to traverse the graph by using Fleury's algorithm, combined with handwriting generation models. The approach exhibited a reduced computational cost because it divided the whole graph into sub-graphs by detecting the graphs' bridges. This method is improved

[2] github.com/gioelecrispo/wor

in [47] by introducing a feedback connection between the unfolding module and a module that extracts elementary movements from the recovered trajectory. The method exploits the analogies between unfolding and segmentation processes and those occurring in the brain when a trajectory plan is learned and executed.

Furthermore, the search for the optimal trajectory along the graph representing the skeleton of a word is executed in [48]. The authors use both the Greedy and the Dijkstra algorithms with a well-defined smoothness function. Further, they focus on reconstructing the optimal trajectory of words by splitting them into many strokes according to their respective curvature values and a set of rules of handwriting. Good continuity criteria derived from both visual perception and movement execution are applied to signatures in [50]. In particular, the implementation focuses on a multiscale analysis of the thinned trajectories and the Dijkstra algorithm.

For learning-based approaches, they require some exemplars of static images with the corresponding drawing orders. Thus, models are trained to recover the trajectories of new static images. In [51], an HMM is adopted for recovering the trajectory of single-stroke handwritten signatures. Each state of the HMM has a probability density function that embeds geometric shape information of the static image, while transition probabilities define the possible pen movements between static image coordinates. A training phase is proposed in [52], in which the original trajectory order and other attributes, such as the length and direction, are extracted from a set of on-line scripts to build a universal writing model. Then, it is used to reconstruct the drawing order during the test phase. The skeleton of the static image is matched to the model by using a dynamic programming algorithm, and the trajectory with maximum likelihood is selected.

Since 2018, deep learning techniques have been exploited in recovering trajectories in Chinese, Japanese, Indic, Arabic, and Latin characters (e.g. [41], [58]). While these techniques may be promising, they have the disadvantage of requiring a huge amount of data for training. Presently, deep learning methods are used to estimate the trajectory of characters and numbers, which are less complex than words and signatures. In [53], an algorithm is proposed based on a regression Convolutional Neural Network (CNN) model to predict the probability of the next stroke point position. The same authors present improvements of such an architecture in [54], based on two CNNs. Another model based on an encoder-decoder LSTM module was introduced in [20]. Here, the encoder module consists of a Convolutional LSTM network, which takes an off-line character image as input and encodes the feature sequence to a hidden representation. The output of the encoder is fed to a decoder LSTM that sequentially predicts the coordinate points. The architecture is tested on characters from three Indic scripts. Experimentation shows that the main limitation of the approach is the need to train a separate model for each individual script.

A handwriting recognition system based on a writing order recovery algorithm is proposed in [55]. The order recovery algorithm exploits an end-to-end system based on a VGG-LSTM, which extracts and encodes features, followed by a BLSTM used as a decoder to generate temporal coordinates. The method could eventually produce human-like velocities [18]. Moreover, a network of two variational auto-encoders is proposed in [56] to convert on-line and off-line handwritten Latin characters to each other. An improved VGG-16 CNN model is proposed in [57] to recover the handwriting stroke order. The CNN model recovers the writing order effectively, even if the accuracy of the network decreases as the number of strokes increases.

As summarized in Table I, an analysis of the state of the art shows that the static handwriting trajectory recovering problem is far from solved despite the very high number of papers that have been published on the subject. Many methods have been designed, and they mostly start from the assumption that the static image consists of a single component, i.e., a pen-down. This assumption does not hold in real applications, where handwriting patterns consist of many pen-downs and pen-ups, such as in signatures. Furthermore, many systems on the subject have not been validated on public datasets, but rather, have only been tested on a few samples. Even more, performance of some systems are only qualitatively measured. Eventually, we notice that studies reported in Table I adopt different metrics and strategies for measuring the goodness of the writing order recovery systems: some authors use the accuracy obtained by recognition systems on the recovered trajectory as an indirect metric, whereas others compare the recovered trajectory with the data acquired by a tablet.

III. Estimating the Writing Order in Handwriting

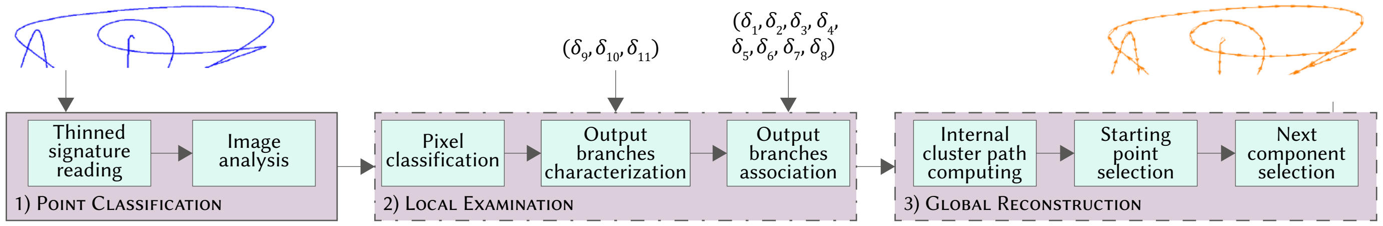
The objective of the proposed system is to estimate the writing order of the 8-connected thin line representation of the handwriting image. The analysis of two elements (components and clusters) plays a vital role in the trajectory recovering process. A cluster is formed following the intertwining of different strokes, resulting in an agglomeration of pixels. Each pixel in the cluster therefore has more than two pixels connected in its 8-neighborhood, and consequently, the clusters complicate the accurate component drawing process.

A flowchart of the proposed system is depicted in Fig. 1. It is composed of three stages. 1) *Point classification*, where the clusters of a thinned specimen are identified, 2) *Local examination*, where individual clusters are resolved by joining their output branches, and 3) Global reconstruction, where we estimate the handwriting order with multiple pen-downs. The mathematical notation used in this work is provided in Table II.

A. Point Classification

We classify each point (or pixel) of the thinned trajectory into one of three categories: (1) end-points, which are pixels with only one 8-neighbor, (2) trace points which have two 8-neighbors, and (3) branch points which have three or more 8-neighbors. Fig. 2 gives a visual example of this categorization. To this end, we identify the clusters as the sets of adjacent pixels labeled as branch points.

To classify the black pixels in the image, we search for their connectivity in their 8-neighbors. The complexity order of this procedure is $O(8 \cdot h \cdot w)$, where (h, w) are the height and width of the image, respectively. Next, the clusters are defined as the sets of adjacent pixels previously labeled as branch points. The clusters are



$(\delta_9, \delta_{10}, \delta_{11})$
$(\delta_1, \delta_2, \delta_3, \delta_4,$
$\delta_5, \delta_6, \delta_7, \delta_8)$
Thinned signature reading
Image analysis
1) Point Classification
Pixel classification
Output branches characterization
Output branches association
2) Local Examination
Internal cluster path computing
Starting point selection
Next component selection
3) Global Reconstruction

Fig. 1. Flowchart of the proposed system for writing order recovering.

TABLE I. Literature Review on Systems for Recovering Static Trajectories

Ref.	Approach	Pattern	Dataset	Evaluation
[34]	Contour-based	Words	Private. 200 images of words written by six different writers.	Quality of ordering ranked by human subjects.
[16], [17]	Contour-based	Words	Public. 10,448 words taken from the IRONOFF dataset. For each word, the on-line and off-line versions are available.	Accuracy of HRS.
[35]	Contour-based	Signatures	Public. SVC 2004 dataset. Images are synthesized from on-line samples.	Preliminary study. The number of detected stroke points.
[36]	Skeleton - Local line	Characters	Private. 10,000 characters written 20 subjects. 2/3 of samples used as a test set.	Quality of ordering ranked by human subjects.
[37], [38]	Skeleton - Local line	Signatures and Numerals	Private. 20 signatures [37] and 150 numerals [38].	Visual inspection for signatures and HRS for numerals.
[39]	Skeleton - Local line	Words and Numerals	Private. 1000 images from US mail service.	Visual inspection.
[40]	Skeleton - Local line	Signatures	Public. The last fifty users of the MCYT-100 dataset wrote the signatures. Images are synthe-sized from on-line samples.	FAR and FRR of ASV.
[41]	Skeleton - Local line	Words	Public. Hundreds of samples from LMCA, IRONOFF, and IFN/ENIT datasets.	Visual inspection.
[24]	Skeleton - Local line	Signatures	Public. Fifty users of the BiosecurID dataset. Images are synthesized from on-line samples.	EER of ASV.
[42]	Skeleton - Graph-based	Characters	Private.	Visual inspection.
[43]	Skeleton - Graph-based	Words	Private. 150 words written by five subjects.	Accuracy of HRS.
[44]	Skeleton - Graph-based	Words	Private. 100 images.	Visual inspection.
[45]	Skeleton - Graph-based	Characters	Public. 708, 811 images obtained by converting on-line data of the Unipen dataset.	Rate of correct complete trajectory recovery, Ac-curacy of HRS.
[46],[47]	Skeleton - Graph-based	Words	Private. 6500 images containing cursive handwrit-ing.	Similarity measure between automatic and man-ual drawing order.
[48]	Skeleton - Graph-based	Words and Characters	Private. 6868 images taken from 3 different datasets. Pubic. Images were taken from IRONOFF [49].	RMSE, DTW, Accuracy of HRS.
[50]	Skeleton - Graph-based	Signatures	Public. 1953 and 2820 on-line signatures from the SigComp2009 and SUSIG - Visual datasets, respectively. The thinned version of the on-line signatures is used.	RMSE, DTW, Number of clusters correctly solved.
[51]	Learning-based	Signatures	Public. 710 single-stroke on-line signatures from fifty users taken from US_SIGBASE and Dolfing datasets. Images are synthesized from on-line samples.	Accuracy score that measures the alignment of the recovered order and the ground truth.
[52]	Learning-based	Signatures	Private. 300 images of signatures.	The rank of the proposed recovering trajectories.
[53], [54]	Learning-based	Characters and digits	Public. OLHWDB 1.1 dataset containing 3755 Chinese characters; 2000 English letters and Arabic digit symbols in UNIPEN.	Rate of correct: end point selection [54], branch points resolution [54], complete trajectory recov-ery [53], [54].
[20]	Learning-based	Characters	Public. LIPI Toolkit dataset (Tamil, Telugu and De-vanagari characters). Around 21, 000 characters per script.	Rate of correct: starting point selection, junction points resolution, and complete trajectory recovery.
[55]	Learning-based	Characters and digits	Public. LMCA and IRONOFF datasets.	Accuracy of HRS.
[56]	Learning-based	Characters	Public. Unipen.	DTW.
[57]	Learning-based	Characters	Public. OLHWDB 1.1 dataset.	Rate of correct complete trajectory recovery.

HRS: Handwriting Recognition System, ASV: Automatic Signature Verifier.

TABLE II. Notation Used

Notation	Description
r	Rank of the cluster
$c_{i,j}$	Curvature between output branches i and j
α_i, β_i	External and internal angle for branch i
$\pi_{i,j}$	Weighted angle direction between branches i and j
$\omega_{ext}, \omega_{int}, \omega_{cur}$	External, internal and curvature weights for computing the anchor point direction
$\delta_1, \delta_2, \delta_3$	Parameters for identifying retracing
$\delta_4, \delta_5, \delta_6$	Parameters for identifying T-patterns
δ_7, δ_8	Parameters for identifying coupled clusters
δ_9	Number of trace points
δ_{10}	Parameter for the brotherhood
δ_{11}	Number of points for computing the curvature

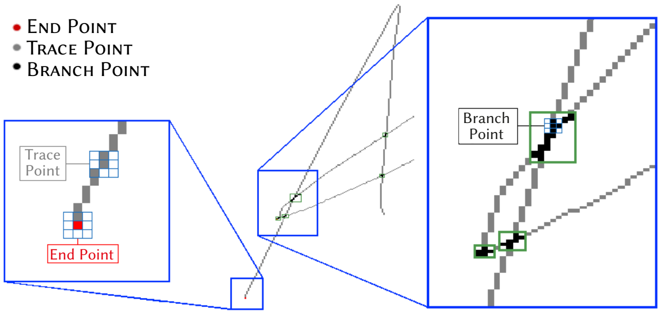


Fig. 2. Point classification in a thinned trajectory (center of the image): details inside the blue rectangles with clusters inside the green ones.

thus identified by performing the connected component labeling algorithm over the branch points. As this algorithm requires at most two scans of the image, its computational complexity is $\mathcal{O}(2 \cdot h \cdot w)$.

B. Local Examination on the Clusters

Recovering a skeleton from an end-point to another through the trace points is an effortless operation. However, the task becomes more demanding when clusters are encountered, since we must then decide on the adequate output branch to recover. The output branches can be also defined as all the trace points that converge in a single cluster.

For this writing order recovery, given a cluster, firstly, its pixels are classified. Secondly, the output branches are characterized, and, finally, they are paired off in input-to-output paths.

C. Pixel Classification in a Cluster

The pixels inside a cluster are classified according to their connectivity as follows:

- **Cluster points**. These are branch points.
- **Anchor points**. These are branch points of the clusters having at least one trace point (i.e., a point outside the cluster) as a neighbor. Since the cluster output branches are anchored on them, they are denoted as *anchor points*. Moreover, the number of anchor points establishes the rank of the cluster, denoted by r.
- **False trace points**. These are labeled as cluster points whether at least one of the following conditions hold: (1) they are connected to two different cluster points of the same cluster; (2) they are connected to a cluster point and to a false trace point of the same cluster; (3) they are connected to two false trace points of the same cluster.

Such a classification in a single cluster requires a computational complexity of $\mathcal{O}(p^2)$, with p being the total number of cluster points. Fig. 3 shows an example of a cluster and its classified pixels.

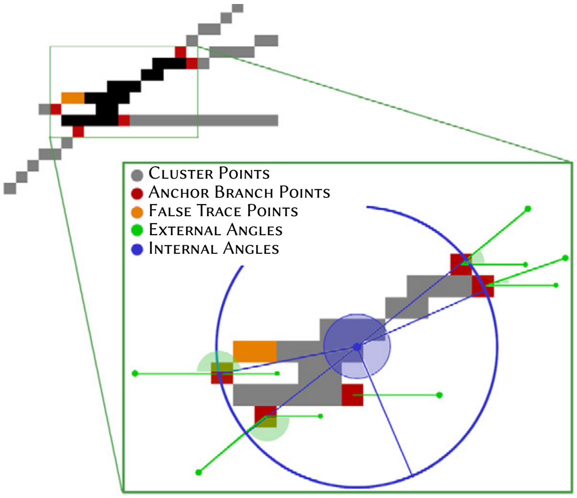


Fig. 3. Example of a cluster: cluster points (in gray), anchor points (in red), false trace points (in orange), external angle directions (in green), and internal angle directions (in blue).

D. Output Branches Characterization

Cluster points and false trace points are identified through their horizontal and vertical coordinates in the images. Beyond the Cartesian coordinates, the anchor points are characterized through the angle direction of the output branches. We developed three strategies for computing the direction of the output branches.

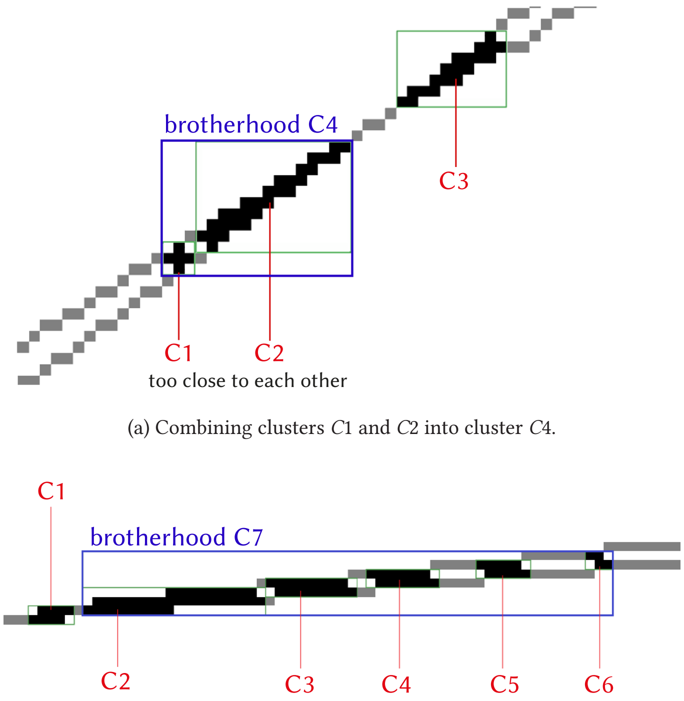
- **External angle**. A multiscale approach [59] is devised to ompute this angle by using only information external to the cluster. This algorithm considers the $\{x_i, y_i\}_{i=1}^{i=\delta_9}$ coordinates of an anchor point (AP_x, AP_y) and δ_9 numbers of trace points of the output branch (see Table II). The output of this algorithm is the external angle α_i, $\forall i \in 1, ..., r$. The algorithm is formalized in Algorithm 1, which runs in time $\mathcal{O}(\delta_9^2)$. Additionally, Fig. 3 illustrates the external angles on the anchor points.
- **Internal angle**. The internal angles can compensate for some shortcomings of the external angles when the output branches contain a few trace points. For this compensation, the cluster center of gravity of the anchor points is calculated. The (x, y) coordinates of the first δ_9 trace points from an output branch are considered and the internal angles β_i, $\forall i \in 1, ..., r$ are computed by the procedure reported in Algorithm 2, its complexity being $\mathcal{O}(\delta_9^2)$.
- **Curvature**. We combine all pairs of output branches, $C_2^r = \binom{r}{2}$, chosen among the r available anchor points in a cluster. The continuity connection of two output branches (i, j) and its shortest path is obtained through Dijkstra's algorithm [60]. Each of the 8-connected traces thus created is divided into n equidistant points and a curvature representative value is calculated ($\rho = [\rho_1, \rho_2, \ldots, \rho_n]$). Finally, the maximum of the difference between adjacent elements of ρ is used to quantify the curvature of the trace generated with the output branches (i, j). The curvature $c_{i,j}$ goes from 0, if the curve is perfectly straight, to 180 degrees, if the curve is bent over itself. The procedure is detailed in Algorithm 3, which has a complexity of $\mathcal{O}(2\delta_{11}^2)$.

Algorithm 1. Compute External Angle

```
procedure COMPUTEEXTANG($AP_x$, $AP_y$, $x$, $y$)
            ▷ $\delta_9$ being the scale of the multiscale approach and the length of $(x, y)$
  for $s = 1$ to $\delta_9$ do
    $k \leftarrow 1$
    $nextPixel \leftarrow (AP_x, AP_y)$
    for $i = 1$ step $s$ to $\delta_9$ do
      $distX \leftarrow nextPixel_1 - x_i$
      $distX \leftarrow nextPixel_2 - y_i$
      $angInt_k \leftarrow atan2(distY, distX)$
      $k \leftarrow k + 1$
      $nextPixel \leftarrow (x_i, y_i)$
    end for
    $minAngle \longleftarrow$ min of $angInt$
    $maxAngle \longleftarrow$ max of $angInt$
    $angMS_{(s,:)} \longleftarrow$ $linear\ interpolation$
  end for
  for $s = 1$ to $\delta_9$ d
    $resMultiscale \longleftarrow$ circular mean of $angMS_{(s,:)}$
  end for
  $\alpha \longleftarrow$ circular mean of $resMultiscale$
  return $\alpha$
```

E. Handling Special Cases: the Brotherhood

The brotherhood refers to a set of joined clusters. Two clusters can be merged when there are less than δ_{10} trace points from each other, and are connected by at least two branches. The brotherhood is a full-fledged cluster, and we apply the same rules of a single cluster to it. Fig. 4 shows two examples of brotherhood process. It is worth pointing out that cluster C3 (Fig. 4a) and cluster C1 (Fig. 4b) are not included in the brotherhood because they are connected with the other clusters only through one branch.



(a) Combining clusters $C1$ and $C2$ into cluster $C4$.

(b) Combining clusters $C2$, $C3$, $C4$, $C5$ and $C6$ into cluster $C7$.

Fig. 4. Examples of two brotherhood application.

Algorithm 2. Compute Internal Angle

```
procedure COMPUTEINTANG($ccg_x$, $ccg$, $x$, $y$)
                                        ▷ $\delta_9$ being the length of $(x, y)$
  for $i = 1$ to $\delta_9$ do
    $vx_i \leftarrow ccg_x - x_i$
    $vy_i \leftarrow ccg_y - y_i$
    $ang_i \leftarrow atan2d(vx_i, vy_i)$
  end for
  $\beta \longleftarrow$ circular mean of $ang$
  return $\beta$
```

Algorithm 3. Compute Curvature

```
procedure COMPUTECURV($x$, $y$, $\delta_{11}$)
  $m \leftarrow$ length of $(x, y)$
  $n \leftarrow \min(\delta_{11}, m)$
  if $m \leq 2$ then
    return 0                                   ▷ The curve is straight
  else
    s $\leftarrow$ floor(linspace(1, $m$, $n$))
                                  ▷ Indexes of $(x, y)$ to be considered
    for 1 = 1 to n do
      for a = 1 to $\delta_{11}$ do
                                                   ▷ Foward points
        if $1 \geq 1$ and $1 \leq n - 1$ then
          $k \leftarrow \min(n - l, a)$
          for f = 1 to $k$ do
            $v_f \leftarrow$ atand2 $(y_{s_l} - y_{s_{l+f}}, x_{s_l} - x_{s_{l+f}})$
          end for
                                                 ▷ Backward points
        if $1 \geq 2$ and $1 \leq n$ then
          $k \leftarrow \min(l - 1, a)$
          for b = 1 to $k$ do
            $v_b \leftarrow$ atand2 $(y_{s_l} - y_{s_{l+b}}, x_{s_l} - x_{s_{l+b}})$
          end for
        $v \leftarrow$ concat$(v_b, v_f)$
        $ang_a \leftarrow$ circular mean $(v)$
      end for
      $\rho_l \leftarrow$ circular mean $(v)$
    end for
    $c = \max(\Delta\rho_l)$
    end $c$
```

The complexity of the brotherhood is $O(nC \cdot r \cdot \delta_{10}^2)$, where nC is the number of clusters in the brotherhood, r the rank of the cluster and δ_{10} the number of trace points.

F. Output Branches Association

The Gestalt theory [61] states that all elements of sensory input are perceived as belonging to a coherent and continuous whole. Moreover, such criteria are supported by the studies of motor control theories [62], in particular those related to the execution of rapid and smooth movements that involve the principle of energy minimization. Under these perspectives, good continuity criteria are taken into account.

As such, we pair the two exit directions whose external difference is closer to 180 degrees than are the others, remove them from the cluster, and repeat the pairing until the rank of the cluster is either zero (meaning there are no more exit directions to pair) or three. Let

i and j be two branches of a cluster. To implement this criterion, the following weighted angle direction, $\pi_{i,j}\ \forall(i,j) \in r$, is calculated:

$$\pi_{i,j} = \omega_{ext} \cdot |\alpha_i - \alpha_j| + \omega_{int} \cdot |\beta_i - \beta_j| + \omega_{cur} \cdot c_{i,j} \quad (1)$$

where (α_i, α_j) refer to the external angles, (β_i, β_j) to the internal angles and ci,j denotes the curvature between the considered branches. The weights in the angles and curvature, (ω_{ext}, ω_{int}, ω_{cur}) (see Table II), have to satisfy that the sum of their values be equal to one, with each weight ranging in the (0, 1) interval. According to eq. (1), the smaller $\pi_{i,j}$ the smoother the line connecting the i-th and j-th output branches. Once all the $\pi_{i,j}$ are calculated, we process the clusters depending on their ranks:

- for even-rank clusters, we select the pair corresponding to the smallest values of $\pi_{i,j}$ and removing the paired branches. These steps are repeated $\frac{r}{2}$ times, until all the branches are paired;
- for odd-rank clusters, the same procedure as above is applied $\frac{r-1}{2}$ times. The remaining branches constitute a 3-rank 2 cluster.

The 3-rank clusters are by far the toughest to manage [45],[63]. They can be classified according to their geometrical and morphological properties (see Fig. 5) as follows:

- **T-pattern clusters**: These are clusters whose shape is similar to a “T”. With these, one out of the three angles $\pi_{i,j}$ had to satisfy $180\,(1 - \delta_4) \le \pi_{i,j} \le 180\,(1 + \delta_4)$, and the other two, the following condition: $\pi_{i,j} \frac{100}{360} \le \delta_5$. To avoid 360 misclassification, any branch of a T-pattern cluster should not be too close to an end-point and to another 3-rank cluster. Let dep be the distance in pixels between an anchor branch point and the nearest end-point and let d^{3rc} be the distance between an anchor branch point and the nearest 3-rank cluster. Then, the last condition to satisfy to consider a T-pattern would imply: $\min(d^{ep}, d^{3rc}) \ge \delta_6$.
- **Retraced clusters**: These appear in closed handwritten loops, and consequently, at least one of the branches has to end in an end-point. Let d_k^{ep} be the minimum distance in pixels from the branch k to its end-point. Then a retracing should satisfy $d_k^{ep} \le \delta_2$. Also, as the candidate retraced branch is expected to be straight, the value of its curvature c_k should be less than or equal to δ_3. Finally, the opposite angle $\pi_{i,j}$ should also satisfy: $\frac{100\pi_{i,j}}{360} \le \delta_1$.
- **Coupled clusters**: These are two-neighbor 3-rank clusters, none of which is a T-pattern or a retraced cluster, which share one output branch whose length $d_{db} \le \delta_7$. As such, the other branches of both clusters compose a 4-rank cluster, which has to respect the following relation:

  $$\max[\operatorname{avg}(\pi_{1,3}, \pi_{2,4}), \operatorname{avg}(\pi_{2,3}, \pi_{4,1}), \operatorname{avg}(\pi_{2,1}, \pi_{3,4})] \le \delta_8 \quad (2)$$

  This condition means that the good continuity criterion must be strongly observed for these clusters to be classified as coupled.
- **Normal cluster**: This is a 3-rank cluster that does not belong to the previous classes. In this case, the branches corresponding to the smallest values of $\pi_{i,j}$ are associated, and the remaining one is disjoined from the cluster.

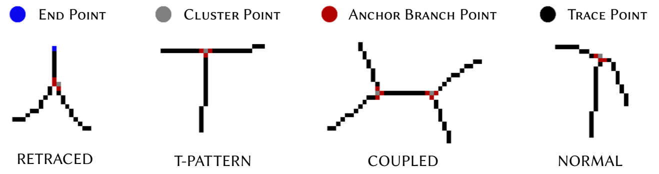


Fig. 5. Classification of the 3-rank clusters.

In our implementation, firstly, we pair the branches in even rank clusters, because these clusters are the less ambiguous ones. Secondly, we work out the odd cluster whose rank is higher than three. Finally, we deal with the three rank clusters since they are the most confusing type.

G. Global Reconstruction

The goal of this section is to recover the writing order of the pen-downs. To this end, we need to define a path that traverses the cluster to link two previously paired branches. Moreover, the starting points of each component and their order need to be estimated as well.

H. Computing Internal Cluster Paths

Once the anchor branch points are paired, we define the path that connects them. Let a cluster be composed of p pixels. The adjacency matrix $\boldsymbol{A}$ is then a $p \times p$ matrix defined as follows: Given a pair of neighbor pixels (p_i, p_j) within the cluster, $\boldsymbol{A}(i, j)$ would have a value of two or three. If (p_i, p_j) are not neighbors or $p_i = p_j$, the value in the adjacency matrix is zero. Based on observations in trajectory generation [31], the weights assigned operate in compliance with the principle of minimizing energy. Accordingly, to connect two neighboring pixels, we exercise a preference in choosing a straight connection rather than an oblique one. At the same time, an oblique connection is preferred against two straight connections.

The adjacency matrix $\boldsymbol{A}$ is therefore processed by the Dijkstra algorithm [60], which approaches the Gestalt theory perspective of good continuity [61] and rapid movement trajectories.

I. Starting Point Selection

To choose the starting point, we model the spatial distribution of the starting points with a two-dimensional Gaussian function.

To this end, we store the starting end-point of a random number of handwriting. The median of this Gaussian function is experimentally located at $0.15 \cdot h$ and $0.35 \cdot w$, with h and w being the height and the width of the writing area on the top-left part of the images. When there are no end-points within the ellipse defined by the mean and the two standard deviations of the Gaussian, the leftmost end-point is selected as the starting point, as usual in Western handwriting.

J. Next Component Selection

Once the first component is recovered, the next component is chosen according to a proximity criterion; this is the component with the nearest end-point, which has not yet been recovered. Formally, this criterion can be described as:

$$i^* = \underset{i \backslash j}{\arg\min} \left(\sqrt{\left(x_{ep_i} - x_{ep_j}\right)^2} + \left(y_{ep_i} - y_{ep_j}\right)^2 \right) \quad (3)$$

where (x_{ep_i} , y_{ep_i}) are the coordinates of the last recovered end-point and (x_{ep_j} , y_{ep_j}) the coordinates of the end-points of components that have not yet been recovered.

Finally, Fig. 6 illustrates three examples obtained with the proposed method.

IV. Experimental Sensitivity and Stability Analysis

In this section, we analyze the sensitivity and the stability of our system in terms of the values of the parameters listed in Table II, δ_1 to δ_{11}, ω_{ext}, ω_{int}, and ω_{cur} , which have been determined heuristically. The performance of the system depends on the pa-rameter values of the system. We define the accuracy rate θ as a performance measure: $\theta = \frac{\#clusters\ correctly\ solved}{\#total\ clusters}$.

A cluster is counted as *correctly solved* if all its branches are paired similarly to the real trajectories, which are used as ground truth data.

The parameters and weights were heuristically optimized in a trial and error procedure in three steps. Initially, all the weights were set to 0.33 and a coarse tuning of the parameters δ_k was conducted.

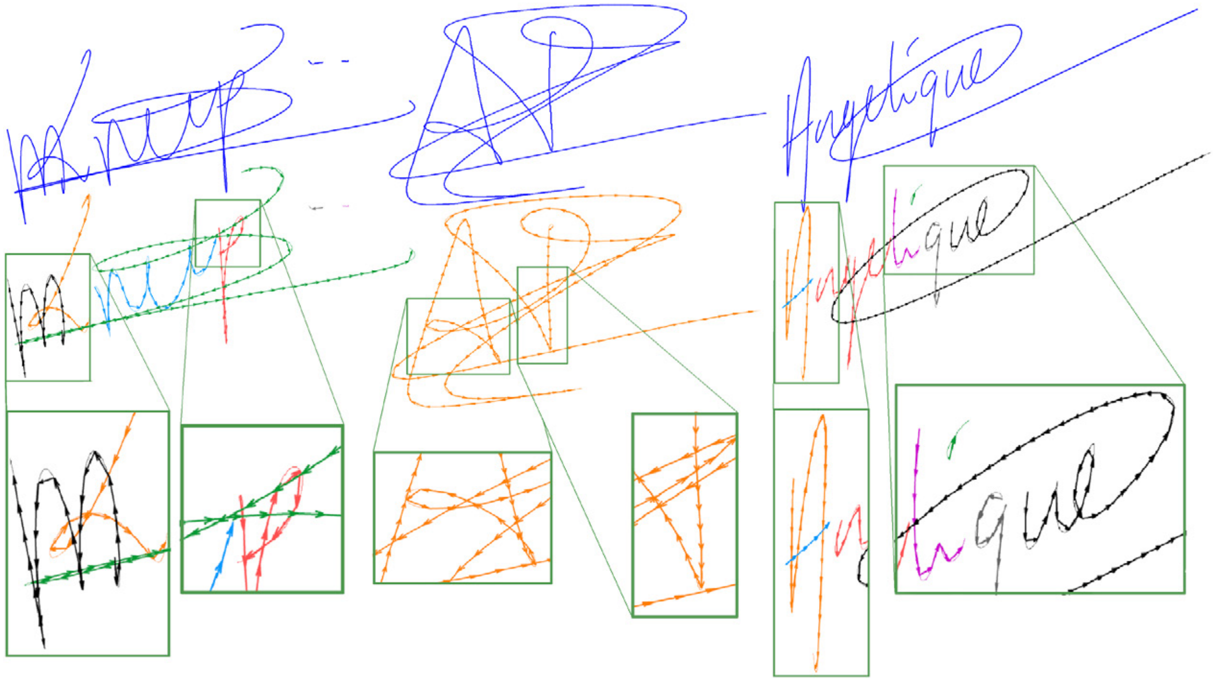


Fig. 6. Examples of writing order recovering in signatures with the proposed method. The recovered writing order of each component is represented by directional arrows, whereas different colors in the estimated trajectories represent different detected components.

Then, the performance of θ was analyzed independently for each decision criterion, cluster rank and type to fine-tune the parameters. Eventually, the weights were fine-tuned for a final maximization of θ.

To investigate whether the heuristic optimization of parameters was biased, we carried out both a sensitivity and a stability study. In both cases, our optimization procedure was performed with 30 % of the specimens randomly selected from the SigComp2009 database [64]. Each experiment was repeated ten times with different values of each parameter under investigation, and the corresponding average accuracy rates were reported.

A. Sensitivity Study: Results

In the study, the *heuristic values* of the parameters δ_k were individually varied, and the results given in terms of the *sensitivity grade*, defined as $\Delta\theta/\Delta\delta$.

TABLE III. Values of δ_k Parameters and their Variability Range for the Sensitivity Study

δ_k	Value	Type	Range
δ_1	28	ND*	(22, 34)
δ_2	20	ND	(10, 30)
δ_3	20	degrees	(10, 30)
δ_4	3	ND	(1, 5)
δ_5	19	ND	(16, 22)
δ_6	8	Pixels	(4, 12)
δ_7	50	Pixels	(46, 56)
δ_8	40	ND	(20, 60)
δ_9	5	Pixels	(3, 7)
δ_{10}	10	Pixels	(6, 14)
δ_{11}	10	Pixels	(6, 14)

*ND stand for non-dimensional.

Table III shows the heuristic values and the variation range we used for each of them. For example, the parameter δ_9 had to include at least two pixels, and as a result, its lowest limit had to be 3.

The sensitivity grade for each parameter variation is shown in Fig. 7. It refers to the accuracy rate variation achieved when only one parameter varies, while the remaining ones assume their heuristic values. We see that, depending on the parameter, the variation of the accuracy rate is in the order of magnitude 10−3 and can therefore be considered negligible. The results confirm that the heuristic criteria we designed, based on the Gestalt theory of perception and motor control theory, capture some properties of the writer's movements when their trajectories exhibit intersections or crossings. We can also see that δ_3 exhibits the highest sensitivity grade, meaning that it is one of the most influential parameters, and conversely, δ_6 seems to be the least influential one.

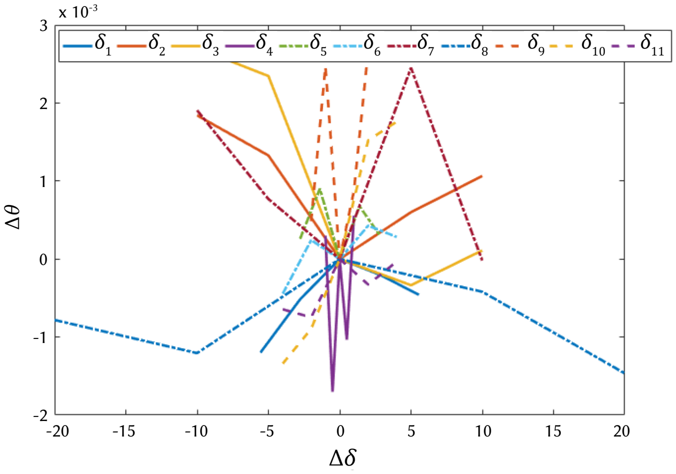


Fig. 7. Sensitivity analysis: Each line shows the variation of the accuracy rate $\Delta\theta$ for a different variation $\Delta\delta$ of the parameter δ_k.

B. Stability Study: Results

In this study, different weight values are obtained by adding Gaussian noise to the heuristic values, as: $\hat{\omega} = \mathcal{N}(\omega, (\eta \cdot \omega)^2)$ where $\eta \in (0.05, ..., 0.5)$ is a distortion factor of the standard deviation of the Gaussian. In each experiment, the value of η was incremented by 0.05 and normalization was carried out in order to satisfy $|\omega_{ext}| + |\omega_{int}| + |\omega_{cur}| = 1$.

Table IV gives the heuristic values of the weights, whereas Fig. 8 shows the accuracy rate obtained for different distortion levels. For each value of η the figure reports the corresponding box plots. As it can be seen, the central mark, in red, is above $\theta = 0.95$ and the majority of data, in blue, are concentrated around the median. Black dots represent the outliers in each case. Consequently, highly stable performances are obtained, along with different η values. It suggested that external and internal angles, as well as the curvature conditions, are representative of the good continuity criteria.

TABLE IV. Weights for Computing the Branch Point Directions

Clusters	ω_{ext}	ω_{int}	ω_{cur}
Normal	0.20	0.05	0.75
T-Pattern/Retracting	0.95	0.00	0.05
Coupled	0.40	0.05	0.55
Odd-Rank	0.70	0.05	0.25

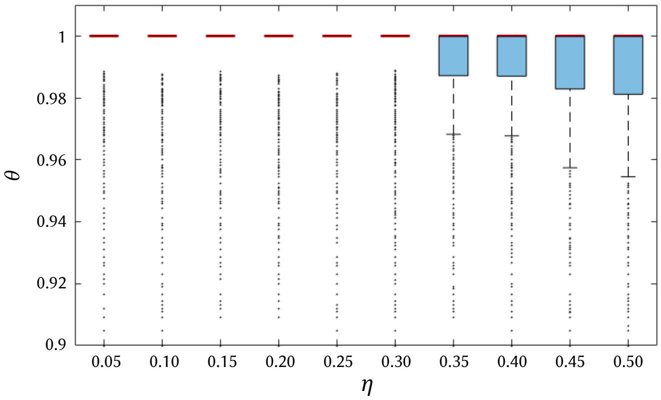


Fig. 8. Stability analysis: Vertical axis shows the accuracy rate (θ) as a function of the amount of distortion (η) applied to the heuristic values of the weights, in the horizontal axis.

V. Performance Assessment

The overall performances of the proposed system were evaluated by comparing the recovered trajectories with the real on-line counterpart trajectories. The experiments aim to answer the following questions:

- Q1: Are the clusters correctly solved?
- Q2: Are the components correctly detected in the images?
- Q3: Does the proposed system recover the trajectories in a correct order?

We used the complete SigComp2009, SUSIG-Visual and SVC-Task2 as third-party databases for the experiments (see appendix).

A. Used Metrics

The evaluation is carried out at the pixel level between real and recovered 8-connected trajectories. To this aim, the on-line data of a specimen were interpolated to generate an 8-connected trajectory through Bresenham's line drawing algorithm [65] without any further processing [45]. As metrics, we used the Root Mean Square Error (RMSE) [48], Signal-to-Noise-Ratio (SNR) [66] and Dynamic Time Warping (DTW) [67] to quantify the matching between them. These metrics are defined as follows:

$$\text{RMSE} = \sqrt{\frac{1}{n}\left(\sum_{i=1}^{n}(x_i - \hat{x}_i)^2 + \sum_{i=1}^{n}(y_i - \hat{y}_i)^2\right)} \tag{4}$$

$$\text{SNR} = 10\log\left(\frac{\sum_{i=1}^{n}\left((x_i - \overline{x}_i)^2 + \left(y_i - \overline{y}_i\right)^2\right)}{\sum_{i=1}^{n}\left((x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2\right)}\right) \tag{5}$$

$$d(n,n) = \text{DTW}\left(\sum_{i=1}^{n}\sum_{j=1}^{n}\sqrt{(x_i - \hat{x}_j)^2 + (y_i - \hat{y}_j)^2}\right) \tag{6}$$

where (x, y) and $(\hat{x}, \hat{y})$ are the points belonging to the real and recovered trajectories, respectively. For the sake of comparison [14], the real and estimated recovered trajectories are normalized by using a cubic spline, where its length, n, was the total sampling points in the real on-line trajectory. Next, min-max scaling is worked out.

It follows from the definition that the smaller the RMSE and DTW, the more similar the real and recovered trajectories. Conversely, higher values of the SNR represent a higher similarity between trajectories.

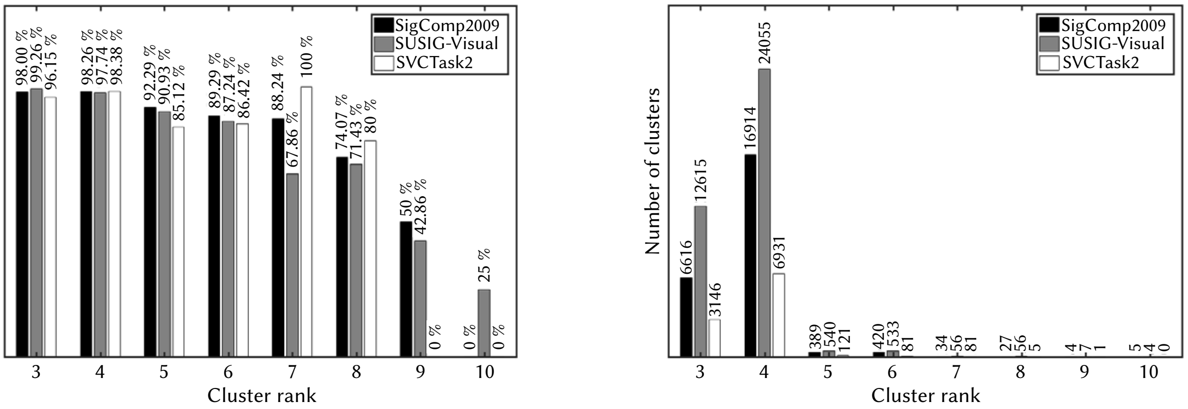
Furthermore, we investigate the relation between the perfor mance and the complexity ($\mathbb{C}$) of the specimens, since the more complex the handwriting, the more difficult the reconstruction of its writing order. Accordingly, we define the complexity as:

$$\mathbb{C} = \alpha_1 \cdot n_c + \alpha_2 \cdot n_{r=3} + \alpha_3 \cdot n_{r>3} \tag{7}$$

where n_c denotes the number of components in the real handwriting, n_r=3 the number of 3-rank clusters and n_r> 3 the number of clusters with a rank greater than three. These three factors represent the difficulty in order recovering trajectories, with the number of components being the most critical factor. Accordingly, their coefficients are empirically adjusted as:$[\alpha_1, \alpha_2, \alpha_3]$ = [0.6, 0.3, 0.1]. Through the complexity formula, a histogram is obtained and divided into three parts with equal frequency binning, in a balanced fashion. To contextualize formula (7), we calculated the complexity using all specimens of the datasets used in the experiments (5972 in total). We found that 1969 samples were categorized as low complexity $\mathbb{C} \in (\alpha_1, 4.0)$; 2002 as medium complexity $\mathbb{C} \in (4.0, 6.6)$ and 2001 as high complexity $\mathbb{C} \in (6.6, 26.3)$.

B. Accuracy of Cluster Resolution (Q1)

Fig. 9b shows that our system detected 183, 877 out of 186, 165 clusters of 3 and 4-rank in all databases. These two cluster types thus represent 98.77 % of the cases handled. The accuracy rate obtained was over 97.97 % on average for the three datasets. As seen in Fig. 9a, for 3-rank clusters, the lowest and highest rates were 96.15 % for SVCTask2 and 99.26 % for SUSIG-Visual, respectively. As mentioned in Section III, 3-rank clusters are the most challenging to solve, as they may exhibit configurations that are very similar to those of the branches attached to them. Achieving an accuracy rate greater than 96.00 % on them thus is a remarkable feat. These results show that the criteria we designed for pairing the cluster branches capture essential pieces of knowledge about human trajectory execution. They also show that the higher the cluster rank, the lower the accuracy rate. This is expected since recovering correcting higher-rank clusters is more difficult than correcting those that are lower-ranked. We also observe that there is room for improvements for branch pairings of clusters with ranks greater than 6. However, this drawback has only a limited impact on the performance because such clusters are present in less than 2.5 % of the total number of samples.



(a) Accuracy of the correctly solved clusters.

(b) Number of detected clusters.

Fig. 9. Performance of the system solving the clusters found in each database.

Overall, we have a global accuracy of $\theta = 98.72$ % on SigComp2009, $\theta = 98.91$ % on SUSIG-Visual, and $\theta = 98.59$ % on SVCTask2, which are little better than the results obtained on the SigComp2009 and SUSIG-Visual datasets [50].

C. Estimation of the Number of Components (Q2)

We assess the estimation of the number of components. Each component has two end-points, corresponding to the points where the pen-tip touches/leaves the tablet. They are used to quantify the number of components in a sample. Therefore, we compare the number of components found by our method with the actual number of components of the on-line samples.

Fig. 10 shows the density functions of the real and estimated numbers of components in the samples. We also quantify the density function similarities in terms of the Area Between Curves (ABC). The more similar the density functions, the smaller the ABC value, and therefore, the better the estimation. For all datasets, we obtain excellent performance, but in the case of the SVCTask2, it is clearly outstanding. It is explained since the other two databases contain more complicated and lengthy specimens than the SVCTask2.

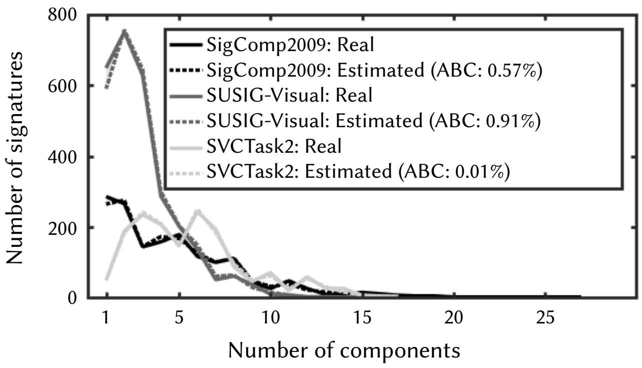


Fig. 10. Density functions of the number of components calculated with real trajectories from different databases and the estimated number of components with our system. ABC denotes the Area Between Curves.

TABLE V. Overall Performance Results When Complete Trajectories Are Recovered in Static Trajectories in Terms of RMSE, SNR, and DTW

Dataset: SigComp2009		Low complexity	Medium complexity	High complexity	Total
SNR	ESTNC	28.93 ± 1.26	8.15 ± 0.63	5.93 ± 0.50	14.39 ± 0.56
	RSENC	33.05 ± 1.25	10.70 ± 0.73	6.06 ± 0.55	16.65 ± 0.60
	RSEOC	41.16 ± 1.07	26.63 ± 0.72	14.63 ± 0.64	27.50 ± 0.55
RMSE	ESTNC	0.14 ± 0.01	0.27 ± 0.01	0.27 ± 0.01	0.22 ± 0.00
	RSENC	0.11 ± 0.01	0.25 ± 0.01	0.29 ± 0.01	0.22 ± 0.01
	RSEOC	0.04 ± 0.00	0.07 ± 0.01	0.16 ± 0.01	0.09 ± 0.00
DTW	ESTNC	2.73 ± 0.18	6.51 ± 0.22	8.82 ± 0.30	6.01 ± 0.15
	RSENC	2.05 ± 0.15	6.16 ± 0.23	9.39 ± 0.31	5.86 ± 0.16
	RSEOC	0.71 ± 0.09	1.82 ± 0.14	5.42 ± 0.29	2.65 ± 0.12

Dataset: SUSIG - Visual		Low complexity	Medium complexity	High complexity	Total
SNR	ESTNC	23.74 ± 0.83	14.35 ± 0.64	7.75 ± 0.51	15.40 ± 0.41
	RSENC	32.32 ± 0.79	19.94 ± 0.69	9.45 ± 0.57	20.73 ± 0.44
	RSEOC	43.34 ± 0.63	32.70 ± 0.59	19.50 ± 0.62	31.98 ± 0.40
RMSE	ESTNC	0.19 ± 0.01	0.23 ± 0.01	0.28 ± 0.01	0.23 ± 0.00
	RSENC	0.11 ± 0.01	0.18 ± 0.01	0.27 ± 0.01	0.19 ± 0.00
	RSEOC	0.03 ± 0.00	0.07 ± 0.00	0.16 ± 0.01	0.08 ± 0.00
DTW	ESTNC	2.71 ± 0.10	3.70 ± 0.12	5.33 ± 0.13	3.90 ± 0.07
	RSENC	1.58 ± 0.09	3.00 ± 0.12	5.33 ± 0.14	3.28 ± 0.07
	RSEOC	0.36 ± 0.04	1.13 ± 0.08	3.05 ± 0.12	1.51 ± 0.05

Dataset: SVCTask2		Low complexity	Medium complexity	High complexity	Total
SNR	ESTNC	23.72 ± 0.84	7.14 ± 0.54	2.62 ± 0.25	11.19 ± 0.41
	RSENC	31.61 ± 0.83	8.70 ± 0.65	2.11 ± 0.34	14.18 ± 0.49
	RSEOC	37.39 ± 0.63	29.30 ± 0.69	20.43 ± 0.60	29.08 ± 0.41
RMSE	ESTNC	0.13 ± 0.01	0.27 ± 0.01	0.31 ± 0.01	0.24 ± 0.00
	RSENC	0.09 ± 0.01	0.28 ± 0.01	0.35 ± 0.01	0.24 ± 0.01
	RSEOC	0.03 ± 0.00	0.06 ± 0.00	0.09 ± 0.00	0.06 ± 0.00
DTW	ESTNC	1.72 ± 0.11	3.62 ± 0.11	4.97 ± 0.10	3.43 ± 0.07
	RSENC	1.16 ± 0.10	3.74 ± 0.12	5.69 ± 0.11	3.52 ± 0.08
	RSEOC	0.34 ± 0.04	0.79 ± 0.07	1.46 ± 0.08	0.86 ± 0.04

D. Matching Between Real and Recovered Trajectories (Q3)

This experiment assesses the performance of our system when the order of a static trajectory is wholly recovered. One of the main factors influencing the complete order recovering is the selection of the starting point of each component. To evaluate the extent to which the selection of the starting points determines the recovery trajectories, we gradually relax the condition of the static samples by adding some information regarding the ending points: 1) the Estimated STarting point Nearest Criterion (ESTNC), which selects as the starting point of the next component the nearest untraced *estimated* starting point; 2) the Real Starting/Ending point Nearest Criterion (RSENC), which provides the system with the coordinates of the *real* starting and ending points of the components, and 3) the Real Starting/Ending point Ordered Criterion (RSEOC), which provides the system with the coordinates of the *real* starting and ending points, as well as the correct order of these coordinates. By comparing the results of the third and second scenarios, it is possible to estimate the efficacy of the criterion for finding the starting point of the next component. As the first scenario represents the operating condition of our method, comparing its performance with the second one allows evaluating the efficacy of the criterion for selecting the starting point of each component.

In Table V, for each database, scenario and level of complexity, we list the performance in terms of the mean and standard error of the performance measures mentioned above.

As expected, we can observe that the simpler the handwriting, the better the reconstruction, for all metrics. They also show that in case of a low complexity handwriting, the performance is very similar across the three datasets, but diverges more and more as the handwriting complexity increases.

With regards to the metrics, the performances in Table V show that the SNR is more sensitive than RMSE and DTW to the errors in recovered trajectories. Nevertheless, all metrics maintain a similar range of values in all cases.

Last but not least, the results in the table show that a significant improvement in the performance is obtained when both the starting/ending points of each component and their orders are made available to our method. This observation is independent of the database and the level of complexity.[3]

TABLE VI. Performance Comparison of Recovered Trajectories of the Estimated STarting Point Nearest Criterion (ESTNC) With Other Works

Paper	Dataset	RMSE*	DTW*
– Related works about recovering words –			
[48]	Private - Single Strokes (on-line transformed in off-line)	400.98	118.12
[48]	Private - Multi Strokes (on-line transformed in offline)	503.81	171.91
[48]	Private - Scanned Words	2538.05	553.57
[48]	IRONOFF [49]	669.03	278.27
[56] (*LSTM*)	Unipen	-	0.04
[56] (*Conv*)	Unipen	-	0.04
[56] (*Class Avg*)	Unipen	-	0.21
– Related works about recovering signatures –			
[14] (*Best system on DTW*)	Public - Arabic signature [33]	0.34	28.81
[14] (*Best system RMSE*)	Public - Arabic signature [33]	0.25	52.40
[50]	SigComp2009	0.06	382.01
[50]	SUSIG-Visual	0.05	300.50
This work	**SigComp2009**	**0.22**	**6.01**
	SUSIG-Visual	**0.23**	**3.90**
	SVCTask2	**0.24**	**3.43**

* Note that different formulas for RMSE and DTW are used in the papers, making a fair comparison more difficult. We used the formulas proposed in the competition presented in [14].

To put our results in context, Table VI shows the performances obtained in related works. We can see different performance ranges among the works, suggesting that no standard procedure has thus far been established for measuring the effectiveness of the writing order recovery. For this reason, beyond the implementation of RMSE and DTW metrics, it is necessary to take into account data normalization, data aggregation, and, above all, the handwriting database used, whose complexity is not easily measurable. Nonetheless, the results of such comparisons are useful, as they generally convey a rough estimate of advancements in the field, even though it does not provide a fair basis of comparison.

VI. Conclusions and Outlook

We have developed a system for recovering the ballistic trajec-tory order of long and complex thinned static handwritten sig-natures. Our system operates in three stages: (*i*) point classifi-cation, (*ii*) local examination, and (*iii*) global reconstruction. In the point classification, the clusters are identified and correspond to the agglomeration of pixels in the images. The agglomerations of lines correspond to crossings of lines in the thinned trajecto-ries. Thus, a cluster can be characterized by the number of input-output lines or branches. In the local examination, input-output branches are paired by exploiting heuristic rules inspired by both good continuity and motor control principles when signing, with preference given to smooth and straight ballistic trajectories. At the global reconstruction stage, the end-points of the components (pen-downs) are identified and sorted. Once a component is re-covered, the system decides on the new component to recovering.

This procedure requires that a number of parameters and weights be adjusted. Their values are determined heuristically by trial and error, as the best matching between the real and reconstructed trajectory is sought. Furthermore, both the sensitivity and stability of the results with respect to these parameters are studied, and we see that the performance of our procedure is barely affected by a variation of up to 10 % of these parameters. To avoid overfitting in these values and make the results more meaningful, the parameters are adjusted with a subset of the SigComp2009 signature database. Then, the results are obtained with different publicly available databases, namely, the complete SigComp2009, SUSIG-Visual and SVCTask2.

The performance of the system is analyzed considering several aspects. As the ground truth of our experiments, we use the on-line trajectories, which contain details of how real signers wrote the trajectory. We first observe a competitive performance when the branches are paired on the clusters. It is worth pointing out that the branch association in the clusters is the first step towards the final writing order recovery. Moreover, a few mistakes may lead to an overall error in the estimation of trajectory order. Secondly, we also study the number of components estimated in the signature and the complete trajectory order recovery. For research purposes, our system can be freely downloaded from GitHub.

Although promising results are observed in this work, more efforts are required to ensure a more reliable estimation of the trajectories. Our experiments suggest that it is of paramount importance to continue investigating the rules for choosing the ending points of the components and their order, taking into account unknown pen-ups trajectories. Indeed, having the availability of on-line trajectories to recover a static one could improve both the cluster resolution and the complete order recovering. In this case, mapping two skeleton-based images is a further strategy that could be explored by using optical flow analysis, diffeomorphism functions, or inkball models, among others.

[3] A video showing the estimated recovered trajectories with different levels of complexity is available at https://youtu.be/TYoZZ8CThhw.

In a challenging framework, which uses an off-line handwriting as input and approximates its corresponding on-line counterpart as output, i.e. $(x(t), y(t))$, our system plays an important role. Nevertheless, this framework also implies that more effort is required in thinning algorithms to improve the handwriting image quality and resolution. Furthermore, estimating temporal properties in the recovered trajectories is another open question. A possible solution is to assign a timestamp sequence to the 8-connected trajectories. This would open the door to working out dynamic properties such as the velocity or acceleration. In the meantime, the proposed system constitutes a reasonable starting point for future research in the challenging field of on-line trajectory estimation from off-line specimens.

Appendix

A. Databases

We evaluate the proposed system on signatures because they represent long and complex handwriting, contain text and flourishes, and their patterns result in multiple pen-downs and a high number of clusters, so constituting a suitable and very challenging benchmark.

We use the following data of three publicly databases:[4]

- SigComp2009 [64]. Contains 1552 on-line Western signatures written by 79 subjects. There are 932 genuine signatures available since each signer gave 12 specimens on average.
- SUSIG-Visual corpus [68]. Includes 2820 on-line Western signatures written by 94 subjects. Each participant produced 20 genuine signatures in two sessions, i.e., 94 × 20 = 1880 specimens.
- SVCTask2 [69]. Consists of 1600 signatures written by 40 subjects, 17 of whom used Oriental scripts and 23, Western scripts. Each subject produced 20 genuine signatures.

SigComp2009 and SUSIG-Visual, which are made up of only Western signatures, were used in a bid to evaluate the independence of the proposed method from the database, whereas the use of SVCTask2, also composed of Oriental specimens, was intended to show the independence of the algorithm from the script type.

Eventually, the skeleton of the off-line handwriting was obtained by converting the on-line trajectories into 8-connected, one-pixel-wide digital lines through the Bresenham's line drawing algorithm [65] without any further processing [45]. The out-put resolution of the images was 600 dpi. This choice, moreover, provides a perfect spatial matching between the on-line and the off-line representations of the trajectories, establishing a solid ground truth for performance evaluation.

Acknowledgment

This study was funded by the Spanish government's MIMECO TEC2016-77791-C4-1-R and PID2019-109099RB-C41 research projects and European Union FEDER program/funds and by the Italian Ministry of Education, University and Research within the PRIN2015 - Handwriting Analysis against Neuromuscular Disease - HAND Project under Grant H96J16000820001.

References

[1] C. D. Stefano, F. Fontanella, A. Marcelli, R. Plamondon, "Graphonomics for the e-citizens: e-health, e-society and e-education," 2019. doi: 10.1016/j.patrec.2018.11.020, Graphonomics for e-citizens: e-health, e-society, e-education.

[2] R. Plamondon, S. N. Srihari, "Online and off-line handwriting recognition: a comprehensive survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 63–84, 2000, doi: 10.1109/34.824821.

[3] R. Plamondon, G. Pirlo, E. Anquetil, C. Rémi, H. L. Teulings, M. Nakagawa, "Personal digital bodyguards for e-security, e-learning and e-health: A prospective survey," *Pattern Recognition*, vol. 81, pp. 633–659, 2018.

[4] C. Tomoiaga, P. Feng, M. Salzmann, P. Jayet, "Field typing for improved recognition on heterogeneous handwritten forms," in *2019 International Conference on Document Analysis and Recognition (ICDAR)*, 2019, pp. 487–493, IEEE.

[5] S. Dash, S. K. Shakyawar, M. Sharma, S. Kaushik, "Big data in healthcare: management, analysis and future prospects," *Journal of Big Data*, vol. 6, no. 1, p. 54, 2019, doi: 10.1186/s40537-019-0217-0.

[6] V. Rowtula, V. Bhargavan, M. Kumar, C. Jawahar, "Scaling handwritten student assessments with a document image workflow system," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018, pp. 2307 – 2314.

[7] R. Rajesh, R. Kanimozhi, "Digitized exam paper evaluation," in 2019 IEEE International *Conference on System, Computation, Automation and Networking (ICSCAN)*, 2019, pp. 1–5, IEEE.

[8] I. H. Hsiao, "Mobile grading paper-based programming exams: automatic semantic partial credit assignment approach," in *European conference on technology enhanced learning*, 2016, pp. 110–123, Springer.

[9] A. Parziale, A. Della Cioppa, R. Senatore, A. Marcelli, "A decision tree for automatic diagnosis of parkinson's disease from offline drawing samples: experiments and findings," in International Conference on Image Analysis and Processing, 2019, pp. 196–206, Springer.

[10] M. Diaz, M. A. Ferrer, D. Impedovo, G. Pirlo, G. Vessio, "Dynamically enhanced static handwriting representation for parkinson's disease detection," *Pattern Recognition Letters*, vol. 128, pp. 204–210, 2019, doi: 10.1016/j.patrec.2019.08.018.

[11] S. Colutto, P. Kahle, H. Guenter, G. Muehlberger, "Transkribus. A platform for automated text recognition and searching of historical documents," in *2019 15th International Conference on eScience (eScience)*, 2019, pp. 463–466, IEEE.

[12] V. Nguyen, M. Blumenstein, "Techniques for static hand-writing trajectory recovery: a survey," in *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, 2010, pp. 463–470, ACM.

[13] Z. Noubigh, M. Kherallah, "A survey on handwriting recognition based on the trajectory recovery technique," in *Arabic Script Analysis and Recognition (ASAR)*, 2017 1st International Workshop on, 2017, pp. 69–73, IEEE.

[14] A. Hassaine, S. Al Maadeed, A. Bouridane, "ICDAR 2013 competition on handwriting stroke recovery from offline data," in *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, 2013.

[15] C. De Stefano, A. Marcelli, A. Parziale, R. Senatore, "Reading cursive handwriting," in 2010 *12th International Conference on Frontiers in Handwriting Recognition*, 2010, pp. 95–100, IEEE.

[16] P. M. Lallican, C. Viard-Gaudin, S. Knerr, "From off-line to on-line handwriting recognition," in *Proceedings of the Seventh International Workshop on Frontiers in Handwriting Recognition*, 2000, pp. 303–312. ISBN: 90-76942-01-3.

[17] C. Viard-Gaudin, P. M. Lallican, S. Knerr, "Recognition-directed recovering of temporal information from hand-writing images," *Pattern Recognition Letters*, vol. 26, no. 16, pp. 2537–2548, 2005, doi: 10.1016/j.patrec.2005.04.019.

[18] B. Rabhi, A. Elbaati, H. Boubaker, A. M. Alimi, "Temporal order and pen velocity recovery for character handwriting based on sequence to sequence gated recurrent unit model," *TechRxiv*, 2020.

[19] L. Rousseau, É. Anquetil, J. Camillerapp, "Recovery of a drawing order from off-line isolated letters dedicated to on-line recognition," in Proceedings of the International Conference on Document Analysis and Recognition, ICDAR, 2005.

[20] A. K. Bhunia, A. Bhowmick, A. K. Bhunia, A. Konwer, P. Banerjee, P. P. Roy, U. Pal, "Handwriting trajectory recovery using end-to-end deep encoder-decoder network," in *2018 24th International Conference on Pattern Recognition (ICPR)*, 2018, pp. 3639–3644, IEEE.

[4] The databases can be downloaded from the following links: Sig-Comp2009 tc11.cvc.uab.es/datasets/SigComp2009_1. SUSIG-Visual biometrics.sabanciuniv.edu/susig.html. SVCTask2 www.cse.ust.hk/svc2004/download.html.

[21] A. Sharma, "Recovery of drawing order in handwritten digit images," in Image Information Processing (ICIIP), *2013 IEEE Second International Conference on*, 2013, pp. 437–441, IEEE.

[22] A. Santoro, A. Parziale, A. Marcelli, "A human in the loop approach to historical handwritten documents transcription," in *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2016, pp. 222–227, IEEE.

[23] B. S. Saroui, V. Sorge, "Trajectory recovery and stroke reconstruction of handwritten mathematical symbols," in *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, 2015, pp. 1051–1055, IEEE.

[24] M. Diaz, M. A. Ferrer, A. Parziale, A. Marcelli, "Recovering western on-line signatures from image-based specimens," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, 2017, pp. 1204–1209, IEEE.

[25] M. Diaz, M. A. Ferrer, D. Impedovo, M. I. Malik, G. Pirlo, R. Plamondon, "A perspective analysis of handwritten signature technology," *ACM Computing Surveys (CSUR)*, vol. 51, no. 6, pp. 1–39, 2019, doi: 10.1145/3274658.

[26] A. Parziale, A. Santoro, A. Marcelli, A. P. Rizzo, C. Molinari, A. G. Cappuzzo, F. Fontana, "An interactive tool for forensic handwriting examination," in *2014 14th International Conference on Frontiers in Handwriting Recognition*, 2014, pp. 440–445, IEEE.

[27] A. Marcelli, A. Parziale, C. De Stefano, "Quantitative evaluation of features for forensic handwriting examination," in *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, 2015, pp. 1266–1271, IEEE.

[28] A. Marcelli, A. Parziale, A. Santoro, "Modeling handwriting style: a preliminary investigation," in *2012 International Conference on Frontiers in Handwriting Recognition*, 2012, pp. 411–416, IEEE.

[29] A. Marcelli, A. Parziale, A. Santoro, "Modelling visual appearance of handwriting," in *International Conference on Image Analysis and Processing*, 2013, pp. 673–682, Springer.

[30] C. Carmona-Duarte, M. A. Ferrer, A. Parziale, A. Marcelli, "Temporal evolution in synthetic handwriting," *Pattern Recognition*, vol. 68, pp. 233–244, 2017, doi: 10.1016/j.patcog.2017.03.019.

[31] A. Marcelli, A. Parziale, R. Senatore, "Some observations on handwriting from a motor learning perspective.," in *2nd International Workshop on Automated Forensic Handwriting Analysis*, 2013, pp. 6–10.

[32] M. Faundez-Zanuy, J. Fierrez, M. A. Ferrer, M. Diaz, R. Tolosana, R. Plamondon, "Handwriting biometrics: Applications and future trends in e-security and e-health," *Cognitive Computation*, vol. 12, no. 5, pp. 940–953, 2020, doi: 10.1007/s12559-020-09755-z.

[33] S. Al-Maadeed, W. Ayouby, A. Hassaine, A. Almejali, A. Al-yazeedi, R. Al-Atiya, "Arabic signature verification dataset," in *Proceedings of the International Arab Conference on Information Technology*, 2012. ISSN: 1812-0857.

[34] R. Plamondon, C. M. Privitera, "The segmentation of cursive handwriting: an approach based on off-line recovery of the motor-temporal information," *IEEE Transactions on Image Processing*, vol. 8, no. 1, pp. 80–91, 1999, doi: 10.1109/83.736691.

[35] B. Kovari, "Time-efficient stroke extraction method for handwritten signatures," in *Proceedings of the 7th International Conference on Applied Computer Science - Volume 7*, ACS'07, Stevens Point, Wisconsin, USA, 2007, pp. 157 –161, World Scientific and Engineering Academy and Society (WSEAS). ISBN: 9789606766183.

[36] G. Boccignone, A. Chianese, L. P. Cordella, A. Marcelli, "Recovering dynamic information from static handwriting," *Pattern recognition,* vol. 26, no. 3, pp. 409–418, 1993, doi: 10.1016/0031-3203(93)90168-V.

[37] S. Lee, J. C. Pan, "Offline tracing and representation of signatures," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 22, no. 4, pp. 755–771, 1992, doi: 10.1109/21.156588.

[38] S. Lee, J. C. Pan, "Handwritten numeral recognition based on hierarchically self-organizing learning networks," in *IEEE International Joint Conference on Neural Networks*, 1991, pp. 1313–1322, IEEE.

[39] D. S. Doermann, A. Rosenfeld, "Recovery of temporal information from static images of handwriting," *International Journal of Computer Vision*, vol. 15, no. 1-2, pp. 143–164, 1995, doi: 10.1007/BF01450853.

[40] J. Hennebert, R. Loeffel, A. Humm, R. Ingold, "A new forgery scenario based on regaining dynamics of signature," in *International Conference on Biometrics*, 2007, pp. 366–375, Springer.

[41] A. Elbaati, M. Kherallah, A. Ennaji, A. M. Alimi, "Temporal order recovery of the scanned handwriting," in *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, 2009.

[42] S. Jager, "Recovering writing traces in off-line handwriting recognition: using a global optimization technique," in *Proceedings of 13th International Conference on Pattern Recognition*, vol. 3, 1996, pp. 150–154 vol.3.

[43] H. Bunke, R. Ammann, G. Kaufmann, T. M. Ha, M. Schenkel, R. Seiler, F. Eggimann, "Recovery of temporal information of cursively handwritten words for on-line recognition," in *Proceedings of the Fourth International Conference on Document Analysis and Recognition*, vol. 2, 1997, pp. 931–935, IEEE.

[44] Y. Kato, M. Yasuhara, "Recovery of drawing order from single-stroke handwriting images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 9, pp. 938–949, 2000, doi: 10.1109/34.877517.

[45] Y. Qiao, M. Nishiara, M. Yasuhara, "A framework toward restoration of writing order from single-stroked handwriting image," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 11, pp. 1724–1737, 2006, doi: 10.1109/TPAMI.2006.216.

[46] L. P. Cordella, C. De Stefano, A. Marcelli, A. Santoro, "Writing order recovery from off-line handwriting by graph traversal," in *20th International Conference on Pattern Recognition (ICPR)*, 2010, pp. 1896–1899, IEEE.

[47] R. Senatore, A. Santoro, A. Marcelli, "From motor to trajectory plan: A feedback loop between unfolding and segmentation to improve writing order recovery," in *15th International Graphonomics Society Conference*, Cancun, Mexico, 2011, pp. 86–69. ISBN: 9780732640033.

[48] M. Dinh, H. J. Yang, G. S. Lee, S. H. Kim, L. N. Do, "Recovery of drawing order from multi-stroke english handwritten images based on graph models and ambiguous zone analysis," *Expert Systems with Applications*, vol. 64, pp. 352–364, 2016, doi: 10.1016/j.eswa.2016.08.017.

[49] C. Viard-Gaudin, P. M. Lallican, S. Knerr, P. Binter, "The IRESTE on/off (IRONOFF) dual handwriting database," in *Proceedings of the Fifth International Conference on Document Analysis and Recognition. ICDAR'99 (Cat. No. PR00318)*, 1999, pp. 455–458, IEEE.

[50] G. Crispo, M. Diaz, A. Marcelli, M. A. Ferrer, "Tracking the ballistic trajectory in complex and long handwritten signatures," in *16th International Conference on Frontiers in Handwriting Recognition*, 2018, pp. 351–356.

[51] E. M. Nel, J. A. Du Preez, B. M. Herbst, "Estimating the pen trajectories of static signatures using hidden markov models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 11, pp. 1733–1746, 2005, doi: 10.1109/TPAMI.2005.221.

[52] K. K. Lau, P. C. Yuen, Y. Y. Tang, "Universal writing model for recovery of writing sequence of static handwriting images," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 19, no. 05, pp. 603–630, 2005, doi: 10.1142/S0218001405004277.

[53] B. Zhao, M. Yang, J. Tao, "Pen tip motion prediction for handwriting drawing order recovery using deep neural network," in *2018 24th International Conference on Pattern Recognition (ICPR)*, Aug 2018, pp. 704–709.

[54] B. Zhao, M. Yang, J. Tao, "Drawing order recovery for handwriting chinese characters," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 3227–3231.

[55] B. Rabhi, A. Elbaati, Y. Hamdi, A. M. Alimi, "Handwriting recognition based on temporal order restored by the end-to-end system," in *International Conference on Document Analysis and Recognition (ICDAR)*, 2019, pp. 1231–1236.

[56] T. Sumi, B. K. Iwana, H. Hayashi, S. Uchida, "Modality conversion of handwritten patterns by cross variational autoencoders," in *2019 International Conference on Document Analysis and Recognition (ICDAR)*, 2019, pp. 407–412.

[57] R. Zhang, J. Chen, M. Yang, "Drawing order recovery based on deep learning," in *2019 Eleventh International Conference on Advanced Computational Intelligence (ICACI)*, 2019, pp. 129–133, IEEE.

[58] H. T. Nguyen, T. Nakamura, C. T. Nguyen, M. Nakagawa, "Online trajectory recovery from offline handwritten japanese kanji characters of multiple strokes," in *25th International Conference on Pattern Recognition (ICPR)*, 2020, IEEE.

[59] C. De Stefano, M. Garruto, A. Marcelli, "A saliency-based multiscale

method for on-line cursive handwriting shape description," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 18, no. 6, pp. 1139–1156, 2004. ISSN:0218-0014.

[60] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959, doi: 10.1007/BF01386390.

[61] F. H. Allport, *Theories of perception and the concept of structure: A review and critical analysis with an introduction to a dynamic-structural theory of behavior.* John Wiley & Sons Inc, 1955.

[62] R. Plamondon, "A kinematic theory of rapid human movements: Part III. kinetic outcomes," *Biological Cybernetics*, vol. 78, no. 2, pp. 133–145, 1998, doi: 10.1007/s004220050420.

[63] T. Steinherz, D. Doermann, E. Rivlin, N. Intrator, "Offline loop investigation for handwriting analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 193–209, 2008, doi: 10.1109/TPAMI.2008.68.

[64] V. L. Blankers, C. E. van den Heuvel, K. Y. Franke, L. G. Vuurpijl, "ICDAR 2009 signature verification competition," in *2009 10th International Conference on Document Analysis and Recognition*, 2009, pp. 1403–1407, IEEE.

[65] J. E. Bresenham, "Algorithm for computer control of a digital plotter," *IBM Systems Journal*, vol. 4, no. 1, pp. 25–30, 1965, doi: 10.1147/sj.41.0025.

[66] M. A. Ferrer, M. Diaz, C. Carmona-Duarte, R. Plamondon, "idelog: Iterative dual spatial and kinematic extraction of sigmalognormal parameters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 1, pp. 114–125, 2020, doi: 10.1109/TPAMI.2018.2879312.

[67] M. Diaz, P. Henriquez, M. A. Ferrer, G. Pirlo, J. B. Alonso, C. Carmona-Duarte, D. Impedovo, "Stability-based system for bearing fault early detection," *Expert Systems with Applications*, vol. 79, pp. 65–75, 2017, doi: 10.1016/j.eswa.2017.02.030.

[68] A. Kholmatov, B. Yanikoglu, "SUSIG: an on-line signature database, associated protocols and benchmark results," *Pattern Analysis and Applications*, vol. 12, no. 3, pp. 227–236, 2009, doi: 10.1007/s10044-008-0118-x.

[69] D. Y. Yeung, H. Chang, Y. Xiong, S. George, R. Kashi, T. Matsumoto, G. Rigoll, "SVC2004: First international signature verification competition," in *International conference on biometric authentication*, 2004, pp. 16–22, Springer.

Moises Diaz

Moises Diaz received the M.Tech., M.Sc., and Ph.D. degrees in engineering from the Universidad de Las Palmas de Gran Canaria, Spain, in 2010, 2011, and 2016, respectively. He joined the University as an Associate Professor in 2021. He is a former Associate Professor at Universidad del Atlantico Medio and Universidad Internacional de La Rioja, Spain. His current research interests include pattern recognition, document analysis, handwriting recognition, biometrics, computer vision, and intelligent transportation systems.

Gioele Crispo

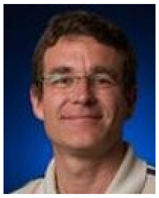
Gioele Crispo is a Machine Learning Engineer focused on NLP and deep learning, AWS Machine Learning certified. He received a M.Sc. in Computer Engineering from the University of Salerno (Italy). He collaborates with the University of Salerno, the University of the Middle Atlantic, and the University of Las Palmas de Gran Canaria (ULPGC) for research in machine learning, pattern recognition, and handwriting. His main interests concern deep learning and the application of transformers and attention mechanisms to images and music.

Antonio Parziale

Antonio Parziale received a Master's degree in Electronic Engineering and a Ph.D. in Information Engineering from the University of Salerno, Italy, in 2009 and 2016, respectively. He does his research at the Natural Computation Laboratory of the University of Salerno since 2010. In 2012, he has co-founded a spin-off company devoted to providing software technologies for automatic document processing. His research interests include handwriting analysis for early diagnosis of neurodegenerative diseases, signature and writer verification, systems neuroscience, and motor control.

Angelo Marcelli

Angelo Marcelli is a professor of computer engineering at the Department of Electrical and Information Engineering and Applied Mathematics of the University of Salerno (Italy), where he is the founder and head of the Natural Computation lab. A. Marcelli is Associate Editor of Pattern Recognition Letters and Associate Editor-in-chief of the Journal of Forensic Document Examination and served as President of the International Graphonomics Society. His research focuses on handwriting, where he has contributed to developing methods and tools for handwriting analysis and recognition and their application to forensic handwriting analysis, historical document processing, on-line and off-line handwriting recognition. More recently, he has been investigating the human motor sys-tem and developed neurocomputational models to elicit the role that different areas of the brain and the spinal cord play in hand-writing learning and generation and the way neurodegenerative diseases affect handwriting production. He has authored more than 150 papers in journals, books, and conference proceedings and holds an international patent on these topics.



Miguel A. Ferrer

Miguel A. Ferrer received an M.Sc. and a Ph.D. from the Universidad Politécnica de Madrid, Madrid, Spain, in 1988 and 1994, respectively. He joined the University of Las Palmas de Gran Canaria, Las Palmas, Spain, in 1989, where he is currently a Full Professor. He established the Digital Signal Processing Research Group in 1990. His current research interests include pattern recognition, biometrics, audio quality, and computer vision applications to fisheries and aquaculture..

ERBM-SE: Extended Restricted Boltzmann Machine for Multi-Objective Single-Channel Speech Enhancement

Muhammad Irfan Khattak[1], Nasir Saleem[2]*, Aamir Nawaz[2], Aftab Ahmed Almani[3], Farhana Umer[4], Elena Verdú[5]

[1] Department of Electrical Engineering, University of Engineering & Technology, Peshawar (Pakistan)
[2] Department of Electrical Engineering, FET, Gomal University, Dera Ismail Khan (Pakistan)
[3] School of Electrical Engineering, Shandong University, Jinan (China)
[4] Department of Electrical Engineering, Islamia University, Bahawalpur (Pakistan)
[5] Escuela Superior de Ingeniería y Tecnología, Universidad Internacional de La Rioja (Spain)

Received 9 March 2021 | Accepted 3 July 2021 | Published 11 March 2022

LA UNIVERSIDAD
EN INTERNET

Abstract

Machine learning-based supervised single-channel speech enhancement has achieved considerable research interest over conventional approaches. In this paper, an extended Restricted Boltzmann Machine (RBM) is proposed for the spectral masking-based noisy speech enhancement. In conventional RBM, the acoustic features for the speech enhancement task are layerwise extracted and the feature compression may result in loss of vital information during the network training. In order to exploit the important information in the raw data, an extended RBM is proposed for the acoustic feature representation and speech enhancement. In the proposed RBM, the acoustic features are progressively extracted by multiple-stacked RBMs during the pre-training phase. The hidden acoustic features from the previous RBM are combined with the raw input data that serve as the new inputs to the present RBM. By adding the raw data to RBMs, the layer-wise features related to the raw data are progressively extracted, that is helpful to mine valuable information in the raw data. The results using the TIMIT database showed that the proposed method successfully attenuated the noise and gained improvements in the speech quality and intelligibility. The STOI, PESQ and SDR are improved by 16.86%, 25.01% and 3.84dB over the unprocessed noisy speech.

Keywords

Restricted Boltzmann Machine, Spectral Masking, Speech Enhancement, Speech Intelligibility, Speech Quality, Supervised Machine Learning.

DOI: 10.9781/ijimai.2022.03.002

I. Introduction

The aim of speech enhancement (SE) is to attenuate/suppress the background noise and recover the clean speech from the noise contaminated speech with better intelligibility and speech quality. The speech enhancement is mainly used in a speech communication system to improve the voice quality, recorded multimedia contents, to boost the automatic speech recognition (ASR) accuracy and for robust hearing aids. Many signal processing-based speech enhancement methods are proposed in literature to improve the performance of aforesaid applications which include spectral subtraction [1] and variants [2]-[4], Wiener filtering [5] and variants [6]-[7], minimum mean square error (MMSE) estimator [8] and the variants [9]-[10]. These methods are apt in various real-time speech applications because of less computational complexity. But, they show poor performance in many non-stationary acoustic conditions. To overcome this problem, supervised learning-based speech enhancement methods are opted [11]-[12]. Learning approaches, such as the regression, spectral-mapping, and spectral-masking [13]-[19], Gaussian mixture models-based SE (GMM) [20]-[21], support vector machines-based SE (SVM) [22] and non-negative matrix factorization (NMF) [23]-[24] have been developed and examined for the speech enhancement. In the past few years, speech enhancement is considered as a supervised learning problem, motivated from the time-frequency (T-F) masking in Computational Auditory Scene Analysis (CASA). In such methods, a trained learning machine directly estimates the clean speech or estimates a T-F mask such as ideal binary mask (IBM) and ideal ratio mask (IRM) which are then applied to the T-F representation of the contaminated speech to reconstruct clean speech [25]-[26]. Perhaps, paradigms of data-driven methods present a convenient explanation to grasp the complex mechanism of the acoustic speech distortion. Recently, a number of deep neural network (DNN) frameworks are developed with encouraging results. Starting from the autoencoders to feed-forward DNN, many frameworks have been designed for speech enhancement [27]-[33]. DNN-based methods deal with three attributes: complementary acoustic features, learning algorithm, and training-target. Pursuant to the above explanation, DNN-based supervised speech enhancement methods are categorized into the masking-based and mapping-based enhancement methods. However, we are dealing with masking-based method in this paper.

II. Related Literature

In the recent past, the supervised learning methods for speech enhancement have achieved enormous performance gain and outperformed the conventional signal processing-based speech

* Corresponding author.

E-mail address: nasirsaleem@gu.edu.pk

TABLE I. Gap Analysis of Literature

Reference	Neural Network	Pre-Training	Phase Estimate
[32]	DBN with multiple Mask Estimation	Networks are pre-trained with RBM without raw data	No Phase Estimation
[38]	DBN with Bayesian Estimators	Networks are pre-trained with RBM without raw data	No Phase Estimation
[37]	MCMC and SGD DBN pre-training with RBM	Networks are pre-trained with RBM without raw data	No Phase Estimation
[39]	Recurrent RBM for Pre-training	Networks are pre-trained with RBM without raw data	No Phase Estimation
[30]	Feed Forward DNN with Mask Estimation	Networks are randomly initialized without raw data	Phase Estimated
[26]	Feed Forward DNN with Mask Estimation	Networks are randomly initialized without raw data	No Phase Estimation
Proposed	Feed Forward DBN with Mask Estimation	Networks are pre-trained with RBM with raw data	Phase Estimated

enhancement. The masking-based SE methods outperformed the mapping-based SE methods; but, large performance deterioration can happen as a result of the mismatch conditions. A large performance gain can be achieved if DNNs are layer-wise pre-trained by stacked-multiple RBM (Restricted Boltzmann Machine) [34]. DNN is proposed for the binary classification and feed-forward DNNs and RBM pre-training are used for subband classification for IBM estimation [27]. DNNs are pre-trained with Fuzzy RBM [32], [35] instead of the regular RBM and achieved significant performance by estimating various T-F masks [36]. A unified method based on Monte Carlo Markov Chain (MCMC) and Stochastic Gradient Descent (SGD) for RBM pre-training is proposed [37]. Bayesian estimators are designed for RBM pre-training [38]. Other variants of RBM such as the recurrent-temporal RBM [39], Gaussian RBM, cardinality RBM, pointwise gated RBM, and conditional RBM have been formulated by modifying regular RBM. Recurrent neural network-based speech enhancement method is formulated which exploited recurrent-temporal RBM to explore temporal-correlation between speech frames [35]. The idea is extended to the features of input and output signals into elemental feature-spaces. The network was fine-tuned by jointly optimized RNN with additional masking layer with a reconstruction constraint. A detailed review of the RBMs and their deep structures can be studied in [40]. Many recent studies on deep learning can also be found in [41]-[47]. The gap analysis is given in Table I. It can be observed that in literature either various networks have initialized the parameters with RBM without phase estimation or randomly initialized the parameters with phase estimation. But, the proposed method initialized the network parameters with a more robust way and also the phase is estimated.

In this paper, we examined the supervised learning algorithms in order to train DBN for time-frequency mask (T-F) estimation. Deep learning in speech enhancement is the arrangement of many hidden layers such that the network learns from the input features. Different from shallow neural networks, DNN should not be trained directly by using standard backpropagation algorithm. Since errors propagate through the network and the gradient becomes infinitesimally small that can affect the weights updating in the previous layers. Gradient-vanishing is one of the core challenges in the deep learning. To address the vanishing problem, a multi-layered framework is used, known as Deep Belief Network, a pre-trained DNN with multiple-stacked RBMs. Following the pre-training, the standard backpropagation algorithm is employed. The acoustic features are progressively extracted by multiple-stacked RBMs during the pre-training phase. The hidden acoustic features from the previous RBM are combined with the raw input data to serve as the new inputs to the current RBM. By adding the raw data to each RBM, layer-wise features related to raw data can be progressively extracted, which is helpful to mine valuable information in the raw data [48]. The aim of this work is not to design a state-of-the-art, but rather to examine the proposed pre-training method and compare the performance with DNN using the typical RBM-based pre-training for speech enhancement. The contributions of this paper are summarized and discussed as. (**i**): A novel pre-training method is proposed by stacking RBMs. The acoustic features are gradually extracted by multiple-stacked RBMs during pre-training phase. The hidden acoustic features from a previous RBM are combined with the raw input data to serve as the new inputs to the current RBM. The network parameters are initialized in the unsupervised fashion using RBM. The parameters are further fine-tuned via adaptive gradient descent and backpropagation algorithm. It is observed that the proposed pre-training method outperformed the DNNs which are initialized randomly or pre-trained with typical RBM. (**ii**): Less computational complexity and fast convergence is achieved by the proposed method as compared to the conventional DNN and DBN frameworks. With similar number of the hidden layers and quantity of hidden neurons, the proposed method achieved better speech quality and intelligibility. The reason for the quick network convergence (less MSE errors) is the adaptation of new pre-training method.

The reminder of this paper is organized as follows. Section III recapitulates the RBM and DBN frameworks. The proposed RBM for speech enhancement is presented in Section IV. Experiments are given in Section V. Results are explained in Section VI. The conclusions are given in Section VII.

III. Restricted Boltzmann Machine (RBM) and Deep Belief Network (DBN)

Restricted Boltzmann Machine [34] is an elemental part of DBN framework and is mainly composed of visible and hidden layers used for many applications including speech enhancement. Unlike Boltzmann machine (BM), a RBM confines the interconnections of peer neurons in order to guarantee the mutual independence. The typical RBM structure is shown in Fig. 1. RBM gives probabilistic models and their parameters are consisting of the weights and biases. Let a RBM be represented by *v* visible layer and *h* hidden layer, respectively. The joint probability density of *v* and *h* is given as:

$$p(v,h) = \frac{e^{(-E(v,h))}}{\iint_{v,h} e^{(-E(v,h))}} \tag{1}$$

Where, E (*v*, *h*) indicates the energy function where type of function is determined by the nature of variables in the visible layer. Two common variables in the visible layer are the binary and Gaussian. The binary-binary and Gaussian-Gaussian energy functions are given by equations as:

$$E(v,h) = -\sum_{j\in vis} \alpha_j v_j - \sum_{i\in hid} \beta_i h_i - \sum_{j\in vis, i\in hid} v_j h_i w_{ji} \tag{2}$$

$$E(v,h) = \sum_{j\in vis} \frac{(v_j-\alpha_j)^2}{2\sigma_j^2} + \sum_{i\in hid} \frac{(h_i-\beta_i)^2}{2\sigma_i^2} - \sum_{j\in vis, i\in hid} \frac{v_j}{\sigma_j}\frac{h_i}{\sigma_i} w_{ji} \tag{3}$$

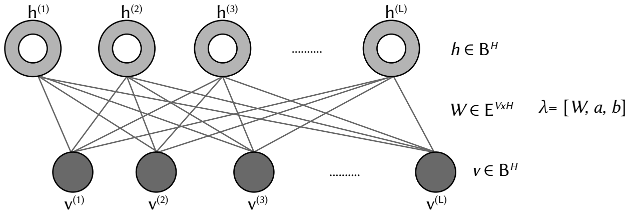


Fig. 1. RBM Network Structure.

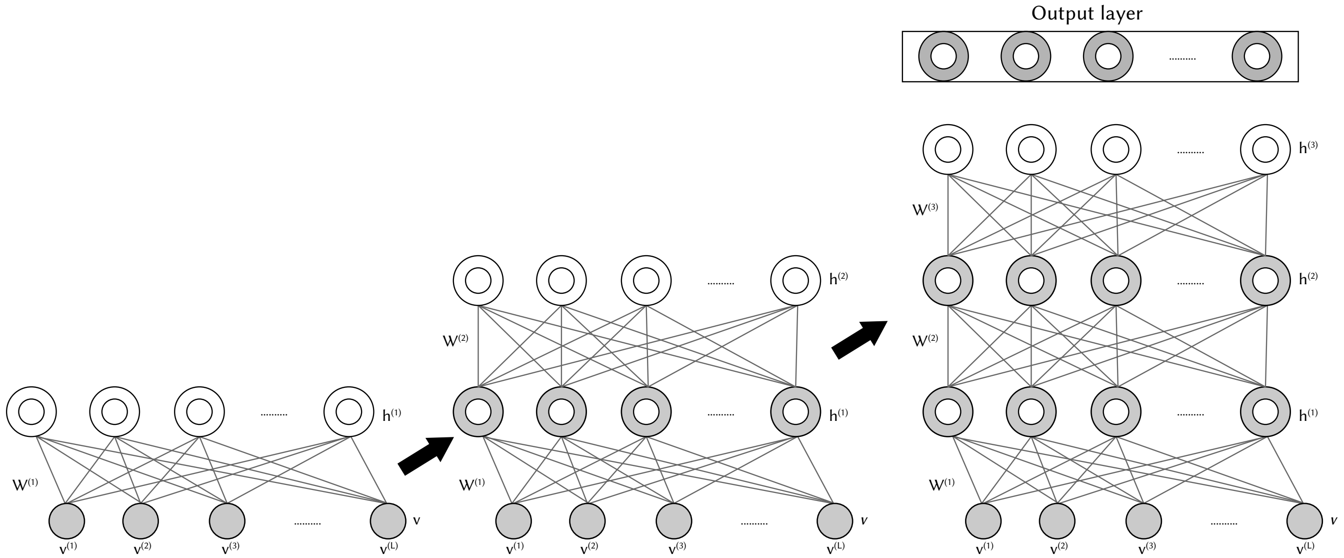


Fig. 2. The DBN framework with Three Hidden Layers.

where v_j and h_i indicate the activation states of the hidden layer neuron j and visible layer neuron i, respectively; α_j and β_i are bias terms whereas w_{ji} indicates the weights used to connect the v_j and h_i. σ_j and σ_i indicate the standard deviation terms. With the joint and marginal probabilistic distributions, i. e, $p(v, h)$, $p(v)$ and $p(h)$, the conditional probabilistic distributions $p(h|v)$ and $p(v|h)$ can be achieved by the Bayesian presumption as:

$$p(h|v) = \frac{p(v,h)}{p(v)} = \frac{e^{(-E(v,h))}}{\int_h e^{(-E(v,h))}} \tag{4}$$

$$p(v|h) = \frac{p(v,h)}{p(h)} = \frac{e^{(-E(v,h))}}{\int_v e^{(-E(v,h))}} \tag{5}$$

For Gaussian neurons, the conditional distributions follow the normal distributions. Deep belief networks consist of multiple stacked RBMs and an output layer added over the final RBM, as shown in Fig. 2. The training process of DBN includes a layer-wise unsupervised pre-training and fine-tuning. During the pre-training step, layer-wise greedy scheme is used for RBM training. Once a RBM is trained, its hidden layer is served as a visible layer to the next RBM. Thus, all RBMs in the network are trained in this fashion by maximizing input data probabilities. Contrastive divergence (CD) method [49]-[50] is applied for parameters updating. After pre-training, a T-F masking layer is appended to the final hidden layer. The entire DBN network is further fine-tuned by reducing the errors between estimated and preset masks. The backpropagation is employed to gradually pass the errors from the final to base input layer. In this way, the entire network parameters are continuously updated.

IV. Proposed RBM-based Speech Enhancement Method

Though DBN framework effectively extracts features and achieves quick convergence by executing pre-training and fine-tuning, yet there can be a room to improve the learning performance. In deep learning, by increasing the number of hidden layers and with layer-wise compression process, important information in the raw data is usually lost in higher layers. To reduce this problem, we extended conventional DBN to amply detain the important information in the raw data by multiple stacked-RBMs. By using the raw data as supplementary inputs to the visible layers to pre-train every RBM, the input raw data participates in entire compression process. As a result, the extracted acoustic features are greatly related to input raw data and the potential important information is copiously kept. Unlike conventional DBN, the proposed extended version of DBN framework can repetitively extract the important information from input raw data, thereby provides deep compressed representations which are in correlation with the input raw data. Fig. 3 illustrates the proposed DBN framework which consists of the pre-training and fine-tuning procedures, respectively. In the pre-training process, the input raw data is appended to visible layers of all RBMs. The weight matrices are composed of w_I and w_H, where w_I connects the hidden layer with the input raw data, and w_H connects the hidden features of the preceding RBM with upper hidden layer. After that, contrastive divergence and the maximum likelihood rules are used to update the RBM parameters. By doing so, we can improve the network learning potential, and can accurately initialize the network parameters for fine-tuning process. During the fine-tuning process, an output layer is added for mask estimation. Finally, the backpropagation is performed iteratively in order to update parameters of network by minimizing the MSE loss function between estimated and preset mask.

A. Pre-Training

The pre-training process of the proposed DBN is to train all RBMs individually. For the first RBM, there is no need to extend input raw data. Every RBM updated its weights and biases which are based on the k-step CD learning (CD-k) method and maximum likelihood rule. In general, maximum likelihood rule is applied to achieve suitable parameters of the network ($\delta = \{w, \alpha, \beta\}$) that excellently fit the input data distribution. By determining logarithmic partial derivatives of data $p(\mathbf{v}=\mathbf{v}_{\mathrm{Data}})$, the gradient updating for network parameters is given as:

$$\frac{\partial \log p\,(v = v_{Data})}{\partial \delta} = -\int_h p(h|v_{Data})\frac{\partial E(v_{Data},h)}{\partial \delta} + \int_v p(v)\int_h p(h|v)\frac{\partial E(v,h)}{\partial \delta} \tag{6}$$

Since it is challenging to determine the second terms in Eq. (6) precisely, CD-k method is used to achieve the estimated solution. It is intended to transfer the data among hidden and visible layers for k times, such that the network states can characterize the model distribution after k iterations. The input to RBM $\mathbf{v}_{\mathrm{Data}}$ can be articulated as $\mathbf{v}^{(0)}$ as shown in Fig. 2. By determining the sampling of the probability $p(\mathbf{h}|\mathbf{v}^{(0)})$, the state of the hidden neurons can be achieved as $\mathbf{h}^{(0)}$. Also, $\mathbf{v}^{(1)}$ can be achieved by sampling of $p(\mathbf{v}|\mathbf{h}^{(0)})$. The learning procedure, from $\mathbf{v}^{(0)}$ to $\mathbf{v}^{(1)}$, is known as one-step Gibbs sampling. After performing k-step Gibbs sampling ($k \longrightarrow \infty$), a stationary distribution is achieved, whereas $\mathbf{v}^{(k)}$ reflects the model distribution. By predicting expectations over $p(\mathbf{v})$, equation (6) can be expressed as:

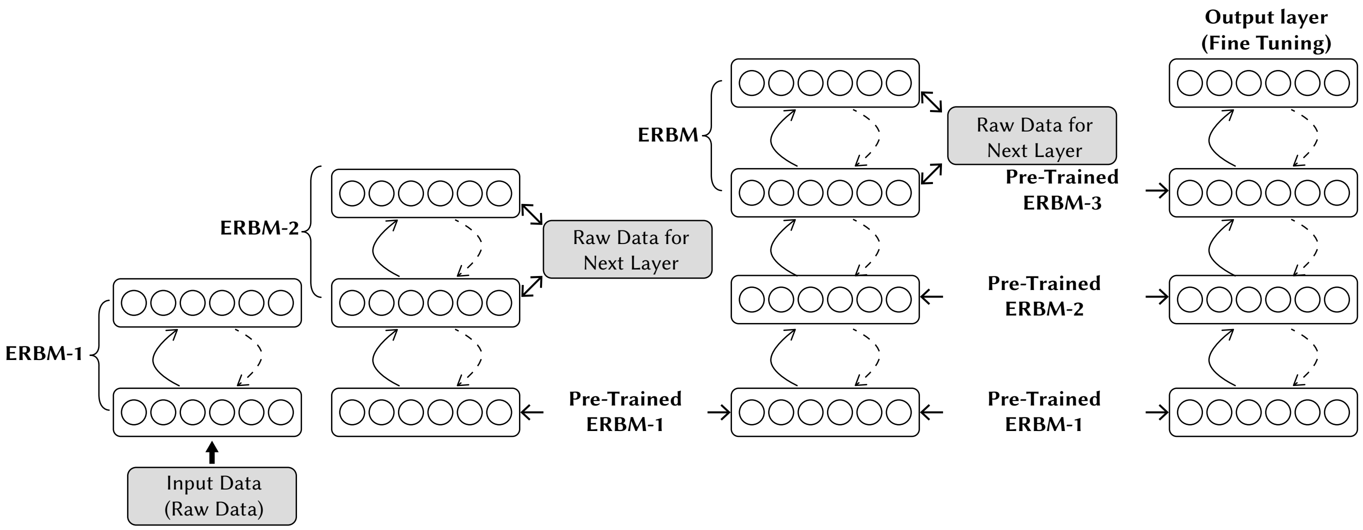


Fig. 3. The Proposed DBN framework structure with three hidden layers during pre-training and fine-tuning.

$$CD_k(\delta, v^{(0)}) = -\frac{\partial E(v^{(0)}-h^{(0)})}{\partial \delta} + \frac{\partial E(v^{(k)}-h^{(k)})}{\partial \delta} \tag{7}$$

The first term is called as negative-term whereas the second term is known as the positive-term, respectively. The negative-term reflects the distribution of raw data and the positive-term reflects distribution of the model. During training, the *k*-step CD takes $k = 1$, that meets the required of calculation for the accuracy. If we substitute Eq. (3) into Eq. (7), equations (Eq. (8)-Eq. (10)) can be obtained. The RBM parameters ($\delta_{\text{PRE}} = W_{\text{PRE}}, \alpha_{\text{PRE}}, \beta_{\text{PRE}}$) can be updated via following equations:

$$\Delta w_{ji} = v_j^{(0)} h_i^{(0)} - v_j^{(k)} h_i^{(k)} \tag{8}$$

$$\Delta \alpha_j = v_j^{(0)} - v_j^{(k)} \tag{9}$$

$$\Delta \beta_i = h_i^{(0)} - h_i^{(k)} \tag{10}$$

$$\delta_{PRE}^{(Epoch+1)} = \delta_{PRE}^{(Epoch)} + \mu \Delta \delta_{PRE}^{(Epoch-1)} + \upsilon \Delta \delta_{PRE}^{(Epoch)} \tag{11}$$

Where, the parameters μ and υ indicate momentum and learning rate, respectively, whereas weighting matrix w_{PRE} is composed of w_I and w_H such that $w_{\text{PRE}} = [w_{\text{I}}, w_{\text{H}}]$. Algorithm-A and Algorithm-B explains the *k*-step CD rule and the pre-training of proposed DBN, respectively.

```
Algorithm-A: k-step CD Process
Input: RBM (v, h), Training Batch B
Output: Estimated Gradient Δw, Δα, Δβ
init $\Delta w_{ji} = \Delta\alpha_j = \Delta\beta_i = 0$, for $j = 1, ..., m; i = 1, ..., n$
for all samples ∈ B do
v^{(0)} ← sample
for t = 0, ..., k-1 do
      for i = 1, ..., n do sample $h_i^{(t)}$ from $p(h_i \mid v^{(t)})$
     for j = 1, ..., n do sample $v_i^{(t+1)}$ from $p(v_j \mid h^{(t)})$
sample $h_i^{(k)}$ from $p(h_i \mid v^{(k)})$ for i = 1, ..., n
for j = 1, ..., m, j = 1, ..., n do
      $\Delta w_{ji} \leftarrow \Delta w_{ji} + v_j^{(0)}.h_i^{(0)} - v_j^{(k)}.h_i^{(k)}$
      $\Delta\alpha_j \leftarrow \Delta\alpha_j + v_j^{(0)} - v_j^{(k)}$
      $\Delta\beta_i \leftarrow \Delta\beta_i + h_i^{(0)} - h_i^{(k)}$
```

```
Algorithm-B: DBN layer-by-layer Pre-Training
Input: Training Set Y
Output: Pre-Trained DBN Framework
for all RBM in DBN framework
init Network Parameters; w, α, β
if training model is RBM then  input ← Y
else input ← combine H and Y
for epoch = 1, ...., e do
      for k=1, ..., floor $(\frac{N_{samples}}{N_{Batchsize}})$ do
      B ← take batch from input
      Δw, Δα, Δβ ← Algorithm-1: k-CD
      w ← w + μ.Δw
      α ← α + μ.Δα
      β ← β + μ.Δβ
H ← Input × w + β
```

B. Fine-Tuning

In fine-tuning of the proposed DBN, the additional layer for output is appended at final hidden layer in order to get probabilities of the samples. The parameters in the proposed DBN $\{(w_{ft}^{(i)}, \beta_{ft}^{(i)})\}_{i=1,2,3,...,m}$ are initialized by pre-trained parameters $\{(w_h^{(i)}, \beta_{PRE}^{(i)})\}_{i=1,2,3,...,m}$ as:

$$\begin{cases} w_{ft}^{(i)} = w_h^{(i)} \\ \beta_{ft}^{(i)} = \beta_{PRE}^{(i)} \end{cases}, i = 1,2,3,4, \ldots, m \tag{12}$$

Where, m indicates the hidden layer's number, the raw data neurons and subsequent parameters that are dropped after pre-training. Random values are used for parameters $(w_{ft}^{(o)}, \beta_{ft}^{(o)})$ of output layer. Thus, the parameters $\delta_{ft}^{(i)}$=$\{(w_{ft}^{(i)}, \beta_{ft}^{(i)}, w_{ft}^{(o)}, \beta_{ft}^{(o)})\}_{i=1,2,3,...,m}$.By using the standard forward-propagation, the loss errors MSE can be computed between estimated and preset mask. Finally, based on the adaptive moment estimation (Adam), parameters of the proposed DBN are further tuned by following equations:

$$\upsilon^{(Epoch+1)} = \upsilon^{(0)} \sqrt{\frac{1-\nu_2^{Epoch}}{1-\nu_1^{Epoch}}} \tag{13}$$

$$\mu_1^{(Epoch+1)} = \nu_1 \mu_1^{(Epoch)} + (1-\nu_1) \Delta \delta_{ft}^{(Epoch)} \tag{14}$$

$$\mu_2^{(Epoch+1)} = \nu_2\mu_2^{(Epoch)} + (1-\nu_2)(\Delta\delta_{ft}^{(Epoch)})^2 \tag{15}$$

$$\delta_{ft}^{(Epoch+1)} = \delta_{ft}^{(Epoch)} - \upsilon^{(Epoch+1)}\left(\frac{\mu_1^{(Epoch+1)}}{\sqrt{\mu_2^{(Epoch+1)}+\rho}}\right) \tag{16}$$

Where $\upsilon^{(Epoch+1)}$ indicates learning rate with initial rate of $\upsilon^{(0)}$ set according to the requirement. The terms $\mu_1^{(Epoch)}$ and $\mu_2^{(Epoch)}$ show the first momentum estimate and second raw momentum estimate. The ν_1, ν_1, ρ are Adam parameters.

V. Experiments

A. Dataset

In experiments, we selected clean speech utterances from TIMIT database [51]. TIMIT corpus includes time-aligned and phonetically balanced 16-bit, 16 kHz speech waveform files. The clean utterances are used for speech enhancement and speech recognition. It contains broadband recordings of 630 speakers of eight major dialects of American English, each reading ten phonetically rich sentences. In order to evaluate the performance of the proposed method in different noisy backgrounds, 10 different noise sources are selected from the Aurora-4 [52] database, given in Table II. The spectrograms of the noise sources are demonstrated in Fig. 4. To produce the noisy speech, we used four signal-to-noise (SNR) levels, -4dB to 2dB with a 2dB step. To train the proposed DBN framework, we have used 2000 speech utterances from different speakers of both genders. For all SNRs, the input training utterances are mixed with 10 noise sources (2000 x 4 =8000 speech utterances). To test the proposed method, 1000 speech utterances from different speakers are used. The experimental results are averaged over 10 noise sources.

TABLE II. Background Noise Sources (N1-N10)

N1: Airport Noise, N2: Babble Noise, N3: Buccaneer, N4: Car Noise, N5: Café Shop Noise, N6: Destroyerengine Noise, N7: Destroyerops Noise, N8: Factory Noise, N9: Hall Noise, N10: Street Noise

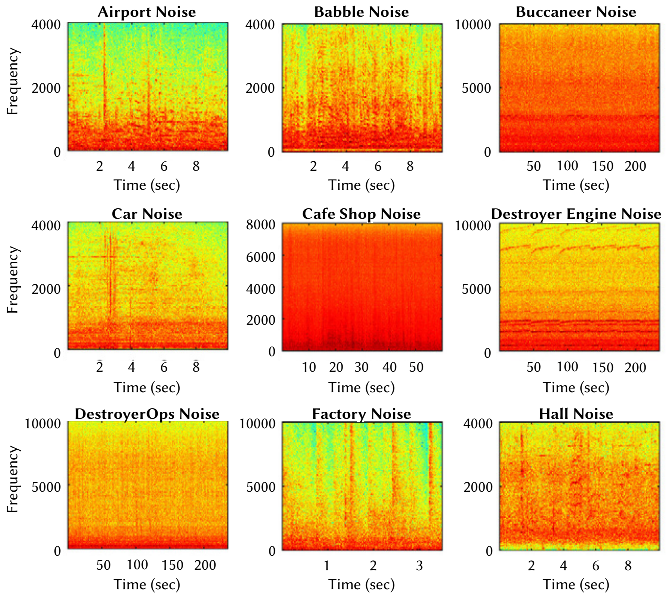


Fig. 4. Spectrograms of Background Noise Sources.

B. Acoustic Features

The acoustic features are extracted from the input speech frames. The frame length and shift in the proposed method are fixed to 20 msec and 10 msec, respectively. The acoustic features set is composed of 13-*d* relative spectral transformed perceptual linear prediction coefficients (RASTA-PLP), 31-*d* Mel-frequency cepstral coefficients (MFCC), 64-*d* gammatone filter-bank energies (GFE), 15-*d* amplitude modulation spectrogram (AMS), where *d* represents feature dimensions. The GFE features are extracted from the T-F representation, known as Cochleagram usually employed in computational auditory scene analysis. Cochleagram expresses the mechanism of the human auditory system. We used a 64-channel gammatone filterbank to extract the GFE features. Furthermore, delta features are computed and affixed to the acoustic feature sets. RASTAMAT toolbox is utilized to extract all the acoustic features. We have used second order auto-regressive moving average filter (ARMA) to obtain the flat temporal trajectories of the acoustic features. The mathematical expression for ARMA is given as:

$$\bar{A}\,(t) = \frac{\bar{A}\,(t-k) + \cdots + A\,(t) + \cdots + A(t+k)}{2k+1} \tag{17}$$

where *A(t)* shows the feature vectors at time frame *t*, *Ā(t)* corresponds to the filtered feature vectors and *k* is the order of filter. In order to add the temporal information, a context window of five frames is used in the proposed method. As a result, we attained 1230-*d* feature vectors. All feature vectors are normalized to zero mean and unit variance before fed to the deep neural networks.

C. Network Architecture

In this paper, a DBN network with a novel pre-training method is employed to learn the magnitude+phase aware spectral-mask. The network architecture is described in this section. DBNs are learning machines and have shown to perform better in speech enhancement. The DBN architecture in this study consists of five layers; an input layer, three hidden layers, and an output layer. The size of the input layer is 1230 neurons, that is, 246*5=1230, including 246-*d* acoustic features and features window composed of 5 frames. Each hidden layer consists of 1024 hidden neurons and the output layer contains 517 visible neurons. From the input to output layer, architecture of the proposed DBN has [1230, 1024, 1024, 1024, 517] neurons. Backpropagation and dropout regularization are used during fine-tuning. Adaptive gradient descent algorithm with a momentum parameter μ is used to optimize DBN. 512 samples batch size is used. The scaling factor for adaptive gradient descent is set to 0.0010 and the learning rate υ is reduced linearly from 0.06 to 0.002. 100 epochs are used during the process. For the first few epochs, the μ is fixed at 0.5 and the rate is increased to 0.8 for remaining epochs. The MSE loss function based on the mask approximation is considered. In supervised spectral masking-based SE, the loss functions are usually formulated to estimate the masking parameters that efficiently restore the clean speech by attenuating undesired noise components in T-F units. The time-domain enhanced speech signals are finally recovered by applying inverse STFT (*i*STFT) using the noisy phase or estimated phase. In this study, the enhanced speech is recovered by using the estimated phase. Spectral-masking methods are found to be successful as T-F masks are dynamically bounded; therefore, achieves quick convergence. In deep learning-based SE, many approaches are opted to estimate a T-F mask and depend on the training-target or the optimization-domain. In mask-approximation (MA) domain, the T-F masks are estimated such that mean square error (MSE) with preset T-F mask is minimized [53], and is given by equation as:

$$MSE_{MA} = \frac{1}{2L}\sum_{k=1}^{K-1}\left[(M_S(t,f) - \widehat{M}_S(t,f))^2\right] \tag{18}$$

Where $\widehat{M}_S(t,f)$ and $M_S(t,f)$ indicates the estimated and preset T-F masks. The rectified linear unit (ReLU) activation converts a weighted sum of the inputs to the model neuron's output. Recent studies show that deep MLPs with ReLU function can successfully be trained by using large training data. Thus, ReLU is used as activation function in hidden layers and sigmoid activation function is used in output layer. The reason for selecting the sigmoid as an output activation function is its dynamic range [0 1]. It is used for models that predict the output probabilities, since probability exists between 0 and 1. Also, the dynamic range of IRM mask exists between 0 and 1. The activation functions are:

$$f(\kappa) = max(0,\kappa); \quad f(\kappa) = \frac{1}{1+e^{\kappa}} \tag{19}$$

D. Evaluation Metrics and Parameters

We extensively evaluated the proposed method by using four objective measures. Perceptual evaluation of speech quality (PESQ) [54] and signal-to-distortion ratio (SDR) are used to quantify speech quality. PESQ, an ITU-T P.862 recommendation calculates the speech quality of enhanced speech with an output value ranging from 0.5 to 4.5. A high PESQ value implies better quality. SDR also measures the quality. Short-time objective intelligibility (STOI) and extended STOI (ESTOI) are used to quantify the intelligibility. STOI [55] and ESTOI [56] measure the intelligibility of the enhanced speech with an output value ranging from 0 to 1. A high STOI and ESTOI value implies better intelligibility. The STOI and ESTOI values are obtained by correlation between clean and enhanced speech signals in short-time overlapped segments. Segmental SNR (SSNR) and output SNR (SNR_O) are used to quantify the residual noise in enhanced speech.

VI. Results

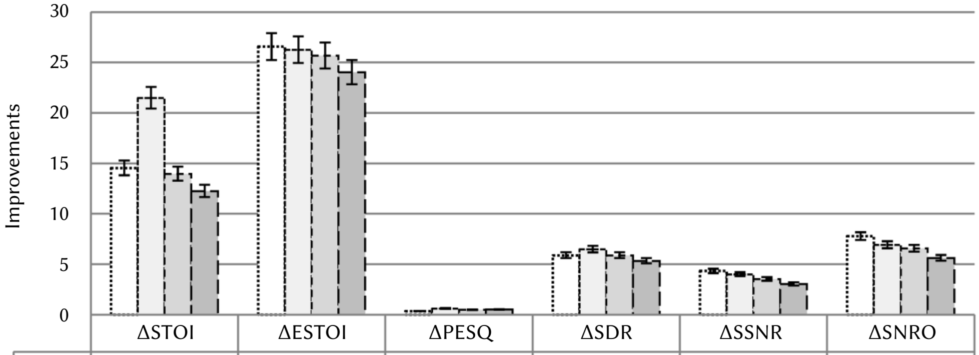
In this section, we provide the major findings of this study. We objectively evaluated the proposed SE method and compared the proposed method with baseline DBN. We additionally compared the proposed method with other related speech enhancement methods from various classes.

TABLE III. Performance Evaluation in Terms of STOI and ESTOI in Four Input SNRs Using TIMIT Corpus and Three Background Noises. DBN_P: Proposed DBN and DBN_B: Baseline DBN

Noise Type →	Airport Noise		Babble Noise		Factory Noise	
SNR -4dB						
Methods	STOI	ESTOI	STOI	ESTOI	STOI	ESTOI
Noisy	62.82	30.61	57.31	23.46	56.58	23.53
DBN_P	78.47	49.48	66.27	37.84	78.22	42.80
DBN_B	76.94	47.82	65.74	37.18	75.69	39.14
SNR -2dB						
Noisy	67.17	36.05	61.18	28.68	60.74	27.71
DBN_P	81.03	53.11	71.53	43.22	69.10	37.38
DBN_B	80.59	52.64	71.62	42.84	68.42	35.22
SNR 0dB						
Noisy	71.83	41.63	65.43	34.01	65.24	33.24
DBN_P	85.05	64.87	76.33	51.63	74.97	48.00
DBN_B	85.02	64.63	76.36	51.64	74.44	47.81
SNR 2dB						
Noisy	76.14	47.70	70.79	40.14	69.87	39.21
DBN_P	86.95	67.73	80.66	57.32	79.11	55.98
DBN_B	86.96	67.36	80.31	57.24	78.88	55.51

TABLE IV. Performance Evaluation in Terms of SDR and PESQ of DBNs in Four Input SNRs Using TIMIT Corpus and Three Background Noises. DBN_P: Proposed DBN and DBN_B: Baseline DBN

Noise Type →	Airport Noise		Babble Noise		Factory Noise	
SNR -4dB						
Methods	SDR	PESQ	SDR	PESQ	SDR	PESQ
Noisy	-3.79	1.58	-4.88	1.44	-3.71	1.34
DBN_P	3.31	1.83	1.52	1.64	-0.17	1.61
DBN_B	2.94	1.70	1.16	1.41	-0.53	1.48
SNR -2dB						
Noisy	-1.83	1.72	-1.80	1.60	-1.81	1.44
DBN_P	4.79	1.96	2.98	1.78	4.51	1.89
DBN_B	4.34	1.90	2.91	1.63	4.47	1.87
SNR 0dB						
Noisy	0.12	1.82	0.13	1.73	0.15	1.56
DBN_P	6.09	2.23	4.56	1.94	6.31	2.11
DBN_B	5.75	2.01	4.55	1.77	5.97	1.98
SNR 2dB						
Noisy	2.10	1.95	2.12	1.85	2.12	1.69
DBN_P	7.38	2.35	6.26	2.15	7.44	2.29
DBN_B	7.32	2.17	6.27	2.07	7.32	2.09



	ΔSTOI	ΔESTOI	ΔPESQ	ΔSDR	ΔSSNR	ΔSNRO
-4	14.53	26.56	0.34	5.88	4.32	7.77
-2	21.49	26.26	0.6	6.48	3.99	6.91
0dB	13.96	25.68	0.48	5.88	3.53	6.57
2dB	12.25	24.02	0.5	5.33	3.03	5.62

Fig. 5. The average PESQ, SDR, STOI, and ESTOI improvements in all noise sources.

TABLE V. Average Comparison Performance of DBN_P and DBN_B in All Noise Sources at Four Input SNRs Using TIMIT Corpus

Input SNRs	DBN_P						DBN_B					
	STOI	ESTOI	SDR	PESQ	SSNR	SNRo	STOI	ESTOI	SDR	PESQ	SSNR	SNRo
-4dB	73.43	45.08	1.77	1.82	2.48	3.55	71.59	43.16	1.09	1.63	2.13	3.13
-2dB	77.49	51.62	4.67	2.08	3.43	4.57	75.99	49.89	4.14	1.89	3.01	4.34
0dB	81.46	57.35	6.04	2.21	3.80	6.43	80.13	55.90	5.72	2.06	3.09	5.98
2dB	84.51	63.01	7.44	2.33	4.95	7.38	83.31	61.78	7.26	2.23	4.39	7.23
Avg.	79.22	54.27	4.98	2.11	3.66	5.48	77.75	52.68	4.55	1.95	3.15	5.17

A. Objective Evaluation

We reported the in depth evaluation for three noise sources, for example, using TIMIT database in Table III and Table IV, where we have used MA-based MSE loss function for training networks. We examined DBN with the proposed pre-training scheme and compared to the DBN with regular pre-training scheme. The T-F mask with the proposed pre-training scheme significantly improved the quality and intelligibility of the noisy speech. Clearly, DBNP outscored the conventional baseline DBNB. For example, at -4dB airport noise, DBNP improved the STOI and ESTOI by 9.65% and 18.87% over the unprocessed noisy speech. Similarly, the STOI and ESTOI at -4dB airport noise are improved by 1.54% and 1.66% over the DBNB. In addition, DBNP improved the STOI and ESTOI by 8.96% and 14.38% over noisy speech at -4dB babble noise. Equally, DBNP improved the STOI and ESTOI at -4dB factory noise by 2.53% and 3.66% over the DBNB. DBNP improved the SDR at -4dB and -2dB airport noise by 7.10dB and 6.62dB over the noisy speech. Similarly, DBNP improved the SDR at -4dB and -2dB babble noise by 0.40dB and 0.45dB over the DBNB. At 0dB factory noise, the DBNP improved the SDR by 5.96dB and 0.34dB over the unprocessed noisy speech and DBNB, respectively. Similarly, the PESQ at -4dB, -2dB and 0dB babble noise are improved by 13.88%, 11.25%, and 12.13% over the noisy speech. Also, the PESQ at -4dB, -2dB and 0dB airport noise are improved by 15.83%, 13.95%, and 10.4% over the DBNB, respectively. The PESQ, SDR, STOI, and ESTOI gains of the proposed pre-training scheme are improving in all noise sources. The average PESQ, SDR, STOI, ESTOI, SSNR and SNRO improvements are demonstrated in Fig. 5. The average PESQ, SDR, STOI, ESTOI, SSNR and SNRO scores with the DBNP and DBNB are given in Table V. The outputs of various objective measures indicate that the proposed pre-training scheme is performing better. The average outputs (STOI, ESTOI, PESQ, SDR, SSNR and SNR_O) are improved over DBN_B by 1.47%, 1.59%, 9.45%, 8.02%, 16.20% and 6.0%, respectively. In order to examine the noise reduction potentials of the proposed method, we used Segmental SNR (SSNR) and output SNR (SNR_O). It is clear from Table V that the proposed method attenuated the background noise and achieved better SSNR and SNR_O as compared to other neural networks with the conventional pre-training scheme. Time-varying spectral analysis graphically demonstrates the vital speech patterns over the time at different frequency bands. In order to envisage performance of the proposed SE, spectrograms of the clean, noisy and enhanced speech samples are plotted in Fig. 6. For better understanding, PESQ, STOI, SDR and SSNR are pointed out over the spectrograms. It is noticeable that DBN_P successfully attenuated the background noise frequencies, and provides a better reconstructed speech compared to the DBN_B. In order to envisage the impacts of phase estimation in the proposed method, spectrograms of the clean, noisy speech, DBN_P, and DBN_B outputs are plotted in Fig. 6. The proposed pre-training scheme considerably improved the speech quality and intelligibility.

B. Comparison With Related Methods

The proposed DBN-based SE method is further judged against other related SE methods including baseline DBN (DBN_B), DNN, deep denoising autoencoder (DDAE) [57], and LMMSE to validate the performance. It is observed that the DBN with proposed pre-training scheme (DBN_P) achieved considerable improvements in terms of the PESQ, STOI, and SDR as well as outscored the related SE methods. On the other hand, the PESQ, STOI, and SDR scores of the baseline DBN underperformed as compared to the DNN and DDAE. Table VI validated that DBN_P outscored the baseline DBN, DNN and DDAE, as well as LMMSE with reasonable margins. For illustration, the STOI is improved from 67.12% with DBN_B at -4dB airport noise to 72.13% with DBN_P and improved STOI by 5.01%. Similarly, the PESQ is improved from 1.56 with DBN_B at -4dB airport noise to 1.79 with DBN_P and improved PESQ by 14.74%. Also, the SDR is improved from 0.98dB with DNN, 0.73dB with DDAE and 0.23 with LMMSE to 1.57dB with

TABLE VI. Average Performance Evaluation Against Related Speech Enhancement Methods

Processing Methods	-4dB			-2dB			0dB			2dB		
	STOI	PESQ	SDR	STOI	PESQ	SDR	STOI	PESQ	SDR	STOI	PESQ	SDR
Noisy	58.90	1.45	-4.11	63.0	1.58	-1.81	67.50	1.70	0.13	72.26	1.83	2.11
DBN_P	72.13	1.79	1.57	76.45	2.02	4.33	80.76	2.18	5.94	83.11	2.23	7.23
DBN_B	67.12	1.56	1.09	73.54	1.81	3.97	78.60	1.95	5.52	82.05	2.11	6.97
DNN	71.33	1.61	0.98	75.87	1.87	4.02	80.43	2.02	5.23	83.47	2.19	7.08
DDAE	70.16	1.54	0.73	73.77	1.73	3.21	77.71	1.89	4.32	80.82	2.00	6.42
LMMSE	65.33	1.48	0.23	69.31	1.67	2.18	71.33	1.78	2.98	75.11	1.93	3.87

TABLE VII. Output SNR and SSNR Performance at Input SNRs Against Related SE Methods

Methods	-4dB			-2dB			0dB			2dB		
	SNR_O	ΔSNR	SSNR	SNR_O	ΔSNR	SSNR	SNR_O	ΔSNR	SSNR	SNR_O	ΔSNR	SSNR
DBN_P	3.56	7.67	2.43	4.75	6.75	3.31	6.36	6.36	3.73	7.21	5.21	4.94
DBN_B	2.98	6.98	1.92	4.03	6.03	2.92	5.76	5.76	3.08	6.31	4.31	3.67
DNN	3.01	7.01	1.98	4.12	6.12	3.01	5.89	5.89	3.11	6.53	4.53	3.98
LMMSE	2.23	6.23	1.61	2.73	4.73	1.79	4.88	4.88	2.74	5.79	2.73	3.27

DBN_P and improved SDR by 0.56dB, 0.84dB and 1.34dB, respectively. At 2dB SNR, the average STOI is improved from 82.05% with DBN_B to 83.11% with DBN_P and improved STOI by 1.06%. Similarly, the PESQ is improved from 2.11 with DBN_B to 2.23 with DBN_P and improved PESQ by 5.68%. Also, the SDR is improved from 7.08dB with DNN, 6.42dB with DDAE and 3.87dB with LMMSE to 7.23dB with DBN_P and improved SDR by 0.15dB, 0.81dB and 3.36dB, respectively.

Table VII demonstrates the performance of the proposed pre-training method in terms of the SNR_O, ΔSNR, and SSNR, respectively. The SSNR measure is employed in order to quantify the residual noise distortion in the output speech. The proposed DBN_P considerably improved the SNR_O and achieved significant performance gain in terms of the SNR_O. The ΔSNRs for DBN_P are higher that for the related SE methods. For example, the SNR_O at -4dB is improved from 2.98dB with DNN_B, 3.01dB with DNN and 2.23dB with LMMSE to 3.56dB with DNN_P and increased SNR_O by 0.58dB, 0.55dB, and 1.33dB, respectively. In case of the SSNR, a consistent output score signifies that DBN-based SE with the proposed pre-training notably attenuated the background noise, confirmed by time-varying spectrograms in Fig. 6.

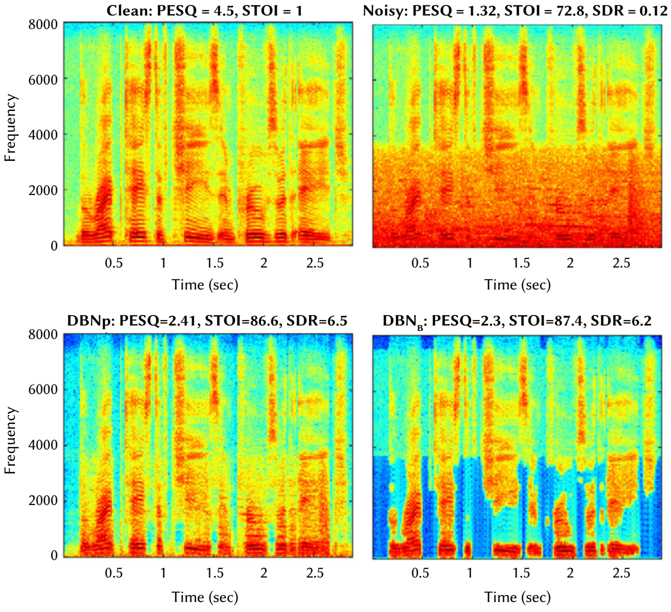


Fig. 6. Time-varying spectral analysis.

C. Network Complexity and Convergence

The complexity of DBN_B/DBN_P relies on the number of network parameters and the forward-backward propagation during tuning of the neural network. In the proposed DBN-based SE method, we have initialized the network parameters by a novel pre-training scheme instead of the randomly initialization. We observed that a network initialized with the proposed pre-training scheme converges quickly as compared to the randomly initialization or initialization with typical pre-training scheme. Moreover, the network complexity also relies on the quantity of hidden neurons and their weights. Greater the quantity of hidden neurons greater will be the network complexity. All DBNs have similar network architecture, quantity of hidden layers, neurons in the hidden and visible layers; however, the proposed DBN converged quickly and showed less complexity. The reason behind the quick convergence (less MSE loss) is incorporation of the novel pre-training scheme. With similar hidden neurons quantity, the proposed DBN-based SE method provided lower MSE errors, and this fact can be observed in Fig. 7. The complexity of the proposed DBN is illustrated in Table VIII, symbolized by "*O*". The forward-backpropagation relies on input features dimension: F_D, training data quantity: T_D, quantity of hidden neurons: T_H, quantity of output neurons T_O, and quantity of epochs for parameters tuning T_E.

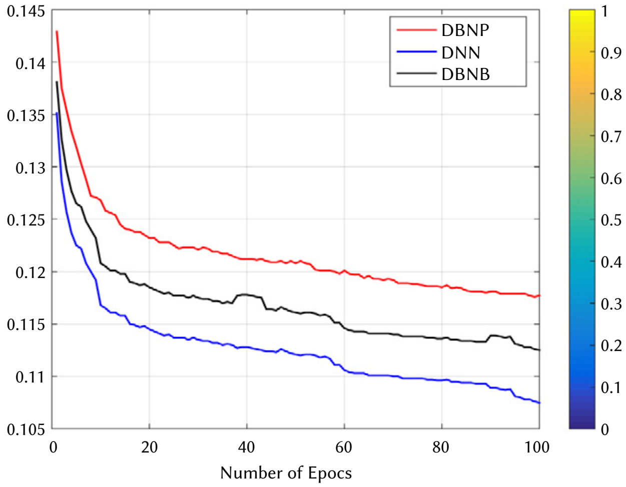


Fig. 7. MSE loss function.

TABLE VIII. Complexity of the Network

Operations	Proposed Network
Forward-Backward Propagation	$O(T_D T_E (F_D + T_H + 2T_H^{\,2} + T_H T_O))$
Average Pre-Training Time	3.33 Hours for 2000 Utterances
Average MSE at 100 Epochs	0.08 Approximately

VII. Discussion and Conclusions

We have proposed and examined a supervised DBN-based speech enhancement method to reduce/attenuate the background noise in single-channel systems. We pre-trained and fine-tuned the DBNs by employing a novel pre-training scheme which incorporates important information available in the raw data to learn a T-F mask. The estimated mask is applied to the noisy speech by using the noisy phase to achieve the enhanced version of degraded speech. In the proposed DBN framework, the acoustic features are progressively extracted by multiple-stacked RBM during the pre-training. The hidden acoustic features from the preceding RBM are combined with raw input data to serve as the new inputs to in-progress RBM. By feeding the raw data to RBMs, layer-wise features related to the raw data can progressively be extracted, which showed useful to mine valuable information in the raw data. The proposed study used the estimated phase during the speech reconstruction to further improve the performance. All acoustic features are the integration of the raw acoustic features in windows, since temporal-dynamics provides important information for speech. The fundamental perception to utilize temporal-dynamics is to employ the DBN architecture, an extension of the feedforward DNN. The DBN framework grabs the long-term temporal-dynamics by using the pre-trained RBM parameters. DBN_P are pre-trained to estimate the IRM, and achieved by 1.47%, 1.59%, 9.45%, 8.02%, 16.20% and 6.0% improvements over the DBN_B in terms of the STOI, ESTOI, PESQ, SDR, SSNR and SNR_O, respectively. The achieved improvements are significant in the speech enhancement. In order to test the generalization ability of the proposed DBN, we have employed the TIMIT database which is composed of male and female speakers. The ΔSNRs and SSNR for DBN_P are higher compared to the related SE methods. We achieved less computational complexity and quick convergence as compared to the baseline DBNs. The spectrogram of the DBN_P indicates a better reconstructed speech signal, suggesting the benefits of the proposed pre-training scheme. To summarize, the proposed DBN-based SE is simple and performed better in terms of improving the intelligibility and quality in the background noisy environments.

future work

Presently, most of the speech processing algorithms operate only with the spectral magnitude, leaving the spectral phase unexplored. With recent advancement in deep neural networks, the phase processing became more important as an innovative and emergent prospective of the DNN-based speech enhancement. In the future, the authors will develop the DBN with phase estimation to test the intelligibility and quality potentials in the complex noisy environments. The unsupervised learning algorithms with modifications can also lead to comparable performances [58] - [60].

References

[1] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction," IEEE Transactions on acoustics, speech, and signal processing, vol. 27, no. 2, pp.113-120, 1979.

[2] Y. Lu and P.C. Loizou, "A geometric approach to spectral subtraction," Speech communication, vol. 50, no. 6, pp.453-466, 2008.

[3] S. Nasir, A. Sher, K. Usman, U. Farman, "Speech enhancement with geometric advent of spectral subtraction using connected time-frequency regions noise estimation," Research Journal of Applied Sciences, Engineering and Technology, vol. 6, no. 6, pp.1081-1087, 2013.

[4] B.L Sim, Y.C. Tong, J.S. Chang, C.T. Tan, "A parametric formulation of the generalized spectral subtraction method," IEEE transactions on speech and audio processing, vol. 6, no. 4, pp.328-337, 1998.

[5] J. Lim and A. Oppenheim, "All-pole modeling of degraded speech," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 26, no. 3, pp.197-210, 1978,

[6] P. Scalart, "Speech enhancement based on a priori signal to noise estimation," in 1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings (Vol. 2, pp. 629-632), IEEE, 1996.

[7] Y. Sandoval-Ibarra, V.H. Diaz-Ramirez, V. I. Kober, V.N. Karnaukhov, "Speech enhancement with adaptive spectral estimators," Journal of Communications Technology and Electronics, vol. 61, no. 6, 672-678, 2016.

[8] Y. Ephraim and D. Malah, "Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator," IEEE Transactions on acoustics, speech, and signal processing, vol. 32, no. 6, pp. 1109-1121, 1984.

[9] Y. Ephraim and D. Malah, "Speech enhancement using a minimum mean-square error log-spectral amplitude estimator," IEEE transactions on acoustics, speech, and signal processing, vol. 33, no. 2, pp.443-445, 1985.

[10] K. Paliwal, B. Schwerin, K. Wójcicki, "Speech enhancement using a minimum mean-square error short-time spectral modulation magnitude estimator," Speech Communication, vol. 54, no. 2, pp.282-305, 2012.

[11] N. Mohammadiha, P. Smaragdis, A. Leijon, "Supervised and unsupervised speech enhancement using nonnegative matrix factorization", IEEE Transactions on Audio, Speech, and Language Processing, vol. 21, no. 10, pp. 2140-2151, 2013.

[12] I. Tashev and M. Slaney, "Data driven suppression rule for speech enhancement," in 2013 Information Theory and Applications Workshop (ITA) (pp. 1-6). IEEE, 2013.

[13] Y. Xu, J. Du, L.R. Dai, C.H. Lee, "A regression approach to speech enhancement based on deep neural networks," IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 23, no. 1, pp.7-19, 2014.

[14] Y. Xu, J. Du, L.R. Dai, C.H. Lee, "An experimental study on speech enhancement based on deep neural networks," IEEE Signal processing letters, vol. 21, no. 1, pp.65-68, 2013.

[15] W. Jiang, F. Wen, P. Liu, "Robust beamforming for speech recognition using DNN-based time-frequency masks estimation," IEEE Access, vol. 6, pp.52385-52392, 2018

[16] N. Saleem, M.I. Khattak, A.B. Qazi, "Supervised speech enhancement based on deep neural network," Journal of Intelligent & Fuzzy Systems, vol. 37, no. 4, pp.5187-5201, 2019.

[17] N. Saleem, M. Irfan Khattak, M.Y. Ali, M. Shafi, "Deep neural network for supervised single-channel speech enhancement," Archives of Acoustics, vol. 44, 2019.

[18] T. Hussain, S.M. Siniscalchi, C.C. Lee, S.S. Wang, Y. Tsao, W.H. Liao, "Experimental study on extreme learning machine applications for speech enhancement," IEEE Access, vol. 5, pp.25542-25554, 2017.

[19] Y. Wang, A. Narayanan, D. Wang, "On training targets for supervised speech separation," IEEE/ACM transactions on audio, speech, and language processing, vol. 22, no. 12, pp.1849-1858, 2014.

[20] G. Kim, Y. Lu, Y. Hu, P.C. Loizou, "An algorithm that improves speech intelligibility in noise for normal-hearing listeners," The Journal of the Acoustical Society of America, vol. 126, no. 3, pp.1486-1494, 2009.

[21] B.M. Mahmmod, T. Baker, F. Al-Obeidat, S.H. Abdulhussain, W.A. Jassim, "Speech enhancement algorithm based on super-Gaussian modeling and orthogonal polynomials," IEEE Access, vol. 7, pp.103485-103504, 2019.

[22] J.H. Chang, Q.H. Jo, D.K. Kim, N.S. Kim, "Global soft decision employing support vector machine for speech enhancement," IEEE Signal Processing Letters, vol. 16, no. 1, pp.57-60, 2008.

[23] K. Kwon, J.W. Shin, N.S. Kim, "NMF-based speech enhancement using bases update," IEEE Signal Processing Letters, vol. 22, no. 4, pp.450-454, 2014.

[24] M. Sun, Y. Li, J.F. Gemmeke, X. Zhang, "Speech enhancement under low SNR conditions via noise estimation using sparse and low-rank NMF with Kullback–Leibler divergence," IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 23, no. 7, pp.1233-1242, 2015.

[25] N. Saleem, M.I. Khattak, "Multi-scale decomposition based supervised single channel deep speech enhancement," Applied Soft Computing, vol.

95, p.106666, 2020.

[26] N. Saleem, M.I. Khattak, "Deep Neural Networks for Speech Enhancement in Complex-Noisy Environments," International Journal of Interactive Multimedia and Artificial Intelligence, vol. 6, no. 1, pp.84-90, 2020.

[27] Y. Wang, D. Wang, "Towards scaling up classification-based speech separation," IEEE Transactions on Audio, Speech, and Language Processing, vol. 21, no. 7, pp.1381-1390, 2013.

[28] K. Phapatanaburi, L. Wang, Z. Oo, W. Li, S. Nakagawa, M. Iwahashi, "Noise robust voice activity detection using joint phase and magnitude based feature enhancement," Journal of ambient intelligence and humanized computing, vol. 8, no. 6, pp.845-859, 2017.

[29] P.S. Huang, M. Kim, M. Hasegawa-Johnson, P. Smaragdis, "Joint optimization of masks and deep recurrent neural networks for monaural source separation," IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 23, no. 12, pp.2136-2147, 2015.

[30] N. Saleem, M.I. Khattak, E.V. Perez, "Spectral Phase Estimation Based on Deep Neural Networks for Single Channel Speech Enhancement," Journal of Communications Technology and Electronics, vol. 64, no. 12, 1372-1382, 2019.

[31] X.L. Zhang and D. Wang, "A deep ensemble learning method for monaural speech separation," IEEE/ACM transactions on audio, speech, and language processing, vol. 24, no. 5, pp.967-977, 2016.

[32] S. Samui, I. Chakrabarti, S.K. Ghosh, "Time–frequency masking based supervised speech enhancement framework using fuzzy deep belief network," Applied Soft Computing, vol. 74, pp.583-602, 2019

[33] N. Saleem, M.I. Khattak, A. Jan, "Multi-objective long-short term memory recurrent neural networks for speech enhancement," Journal of Ambient Intelligence and Humanized Computing, pp.1-16, 2020.

[34] R. Karakida, M. Okada, S.I. Amari, "Dynamical analysis of contrastive divergence learning: Restricted Boltzmann machines with Gaussian visible units," Neural Networks, vol. 79, pp.78-87, 2016.

[35] S. Samui, I. Chakrabarti, S.K. Ghosh, "Deep Recurrent Neural Network Based Monaural Speech Separation Using Recurrent Temporal Restricted Boltzmann Machines," in INTERSPEECH (pp. 3622-3626), 2017.

[36] Z. Chen, Y. Huang, J. Li, Y. Gong, "Improving Mask Learning Based Speech Enhancement System with Restoration Layers and Residual Connection,", in INTERSPEECH (pp. 3632-3636), 2017.

[37] A. Fischer, C. Igel, "An introduction to restricted Boltzmann machines," in Iberoamerican congress on pattern recognition (pp. 14-36). Springer, Berlin, Heidelberg, 2012.

[38] M. Aoyagi, "Learning coefficient in Bayesian estimation of restricted Boltzmann machine," Journal of Algebraic Statistics, vol. 4, no. 1, pp. 31-58, 2013.

[39] I. Sutskever, G.E. Hinton, G.W. Taylor, "The recurrent temporal restricted boltzmann machine," in Advances in neural information processing systems, (pp. 1601-1608), 2009.

[40] N. Zhang, S. Ding, J. Zhang, Y. Xue, "An overview on restricted Boltzmann machines," Neurocomputing, vol. 275, pp.1186-1199, 2018.

[41] S.R. Chiluveru and M. Tripathy, "Low snr speech enhancement with dnn based phase estimation," International Journal of Speech Technology, vol. 22, no. 1, pp. 283-292, 2019.

[42] S.K. Roy, A. Nicolson, K.K. Paliwal, "DeepLPC: A deep learning approach to augmented Kalman filter-based single-channel speech enhancement," IEEE Access, vol. 9, pp. 64524-64538, 2021.

[43] K. Tan, D. Wang, "Towards Model Compression for Deep Learning Based Speech Enhancement," IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 29, pp. 1785-1794, 2021.

[44] S.K. Roy, A. Nicolson, K.K. Paliwal, "DeepLPC-MHANet: Multi-Head Self-Attention for Augmented Kalman Filter-based Speech Enhancement," IEEE Access, vol. 9, pp. 70516-70530, 2021.

[45] A. Pandey, D. Wang, "Dense CNN with self-attention for time-domain speech enhancement," IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 29, pp. 1270-1279, 2021.

[46] S. Abdullah, M. Zamani, A. Demosthenous, "Towards more efficient DNN-based speech enhancement using quantized correlation mask," IEEE Access, vol. 9, pp. 24350-24362, 2021.

[47] N. Saleem, M.I. Khattak, M. Al-Hasan, A.B. Qazi, "On Learning Spectral Masking for Single Channel Speech Enhancement Using Feedforward and Recurrent Neural Networks," IEEE Access, vol. 8, pp. 160581-160595, 2020.

[48] Y. Wang, Z. Pan, X. Yuan, C. Yang, W. Gui, "A novel deep learning based fault diagnosis approach for chemical process with extended deep belief network," ISA transactions, vol. 96, pp. 457-467, 2020.

[49] G.E. Hinton, "A practical guide to training restricted Boltzmann machines," In Neural networks: Tricks of the trade (pp. 599-619). Springer, Berlin, Heidelberg, 2012.

[50] G.E. Hinton, S. Osindero, Y.W. Teh, "A fast learning algorithm for deep belief nets," Neural computation, vol. 18, no. 7, pp.1527-1554, 2006.

[51] V. Zue, S. Seneff, J. Glass, "Speech database development at MIT: TIMIT and beyond," Speech communication, vol. 9, no. 4, pp. 351-356, 1990.

[52] D. Pearce and J. Picone, "Aurora working group: DSR front end LVCSR evaluation AU/384/02," Inst. for Signal & Inform. Process., Mississippi State Univ., Tech. Rep., 2002.

[53] Q. Wang, J. Du, L.R. Dai, C.H. Lee, "A multiobjective learning and ensembling approach to high-performance speech enhancement with compact neural network architectures," IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 26, no. 7, pp.1185-1197, 2018.

[54] A.W. Rix, J.G. Beerends, M.P. Hollier, A.P. Hekstra, "Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs," in 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 01CH37221) (Vol. 2, pp. 749-752). IEEE, 2001.

[55] C.H. Taal, R.C. Hendriks, R. Heusdens, J. Jensen, "A short-time objective intelligibility measure for time-frequency weighted noisy speech," in 2010 IEEE international conference on acoustics, speech and signal processing (pp. 4214-4217). IEEE, 2010.

[56] J. Jensen and C.H. Taal, "An algorithm for predicting the intelligibility of speech masked by modulated noise maskers," IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 24, no. 11, pp. 2009-2022, 2016.

[57] H.P. Liu, Y. Tsao, C.S. Fuh, "Bone-conducted speech enhancement using deep denoising autoencoder," Speech Communication, vol. 104, pp.106-112, 2018.

[58] T. Lavanya, T. Nagarajan, P. Vijayalakshmi, "Multi-Level Single-Channel Speech Enhancement Using a Unified Framework for Estimating Magnitude and Phase Spectra," IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 28, pp. 1315-1327, 2020.

[59] N. Saleem, M.I. Khattak, E. Verdú, "On Improvement of Speech Intelligibility and Quality: A Survey of Unsupervised Single Channel Speech Enhancement Algorithms," International Journal of Interactive Multimedia and Artificial Intelligence, vol. 6, no. 2, 2020.

[60] N. Saleem and T.G. Tareen, "Spectral Restoration based speech enhancement for robust speaker identification," International Journal of Interactive Multimedia and Artificial Intelligence, vol. 5, no. 1, pp. 34-39, 2018.

Muhammad Irfan Khattak

Dr. Muhammad Irfan Khattak is working as an Associate Professor in the Department of Electrical Engineering in University of Engineering and Technology Peshawar. He did his B. Sc Electrical Engineering from the same University in 2004 and did his PhD from Loughborough University UK in 2010. His research interest involves Antenna Design, On-Body Communications, Machine learning, Speech processing and Speech Enhancement.

Nasir Saleem

Dr. Nasir Saleem received BS and M.S degree in Electrical Engineering from UET Peshawar in 2008 and 2012. He received Ph.D. Electrical Engineering, major in Deep Learning for speech processing from UET Peshawar, Pakistan. He is now an Assistant Professor in Department of Electrical Engineering, Gomal University, Pakistan. He published a number of research papers on the deep learning applications in speech processing. His research interests are in the area of Machine learning, digital speech processing and speech enhancement.

Aamir Nawaz

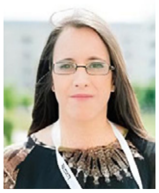
Dr. Aamir Nawaz received BS and MS degree in Electrical Engineering from UET Peshawar and UET Taxila, Pakistan in 2009 and 2014. He received Ph. D in Electrical Engineering, major in Power Systems from Shandong University China. He is serving as Lecturer in the Department of Electrical Engineering, Gomal University, Pakistan. His research interests are in the area of Power Engineering, Machine learning for Power system protection and analysis.

Aftab Ahmed Almani

Mr. Aftab Ahmed Almani received BS and MS degree in Electrical Engineering and pursuing Ph. D in Electrical Engineering, major in Power Engineering from Shandong University China. His research interests are in the area of Power Engineering, Machine learning for Power systems.

Farhana Umar

Dr. Farhana Umar received BS and MS degree in Electrical Engineering from Mehran University of Science & Technology, Jamshoro and Ph.D. Electrical Engineering, from Selçuk Üniversitesi Konya, Turkey. Her research interests are in the area of Power Engg., Machine learning for Power Engg.



Elena Verdú

Elena Verdú Pérez received her master's and Ph.D. degrees in telecommunications engineering from the University of Valladolid, Spain, in 1999 and 2010, respectively. She is currently an Associate Professor at Universidad Internacional de La Rioja (UNIR) and member of the Research Group "Data Driven Science" of UNIR. For more than 15 years, she has worked on research projects at both national and European levels. Her research has focused on e-learning technologies, intelligent tutoring systems, competitive learning systems, accessibility, speech and image processing, data mining and expert systems.

Towards the Grade's Prediction. A Study of Different Machine Learning Approaches to Predict Grades from Student Interaction Data

Héctor Alonso-Misol Gerlache, Pablo Moreno Ger, Luis de la Fuente Valentín *

Universidad Internacional de La Rioja, Logroño (Spain)

Received 3 January 2021 | Accepted 23 September 2021 | Published 23 November 2021

Abstract

There is currently an open problem within the field of Artificial Intelligence applied to the educational field, which is the prediction of students' grades. This problem aims to predict early school failure and dropout, and to determine the well-founded analysis of student performance for the improvement of educational quality. This document deals the problem of predicting grades of UNIR university master's degree students in the on-line mode, proposing a working model and comparing different technologies to determine which one fits best with the available data set. In order to make the predictions, the dataset was submitted to a cleaning and analysis phases, being prepared for the use of Machine Learning algorithms, such as Naive Bayes, Decision Tree, Random Forest and Neural Networks. A comparison is made that addresses a double prediction on a homogeneous set of input data, predicting the final grade per subject and the final master's degree grade. The results were obtained demonstrate that the use of these techniques makes possible the grade predictions. The data gives some figures in which we can see how Artificial Intelligence is able to predict situations with an accuracy above 96%.

Keywords

Artificial Intelligence, Grade Prediction, Machine Learning, Prediction Technology.

DOI: 10.9781/ijimai.2021.11.007

I. Introduction

With the current change in the digital and business paradigm, society's education has a fundamental role to play. It is not only a question of the anachronistic education systems of the industrial revolution not being valid for a society that is trained for jobs that do not yet exist, but the social and mental models have changed.

Combating failure and early dropout from university is an issue of vital importance, especially because of the economic and social cost it generates [1], becoming an issue that has generated growing concern in recent years. Prevent school failure and increase the quality of teaching is an actual object of the educational. Predicting students' results early enables the university and the teacher to carry out more focused teaching work, as well as allowing students to focus their efforts and plan their studies better. Therefore, predicting student grades will have a direct impact on improving education at all levels, helping to combat school dropout and enabling continuous improvement in the academic process, with a consequent positive impact on society and economy.

Teaching today requires great flexibility to provide useful content to a highly changing and dynamic society. This being so, the work of universities is not merely the transmission of knowledge but must be a focus of innovation to teach students how to face the new challenges and opportunities of society, where flexibility is necessary in both teachers and students, educating in knowledge and skills [2]. In this sense, the online university is presented as a great alternative to face-to-face studies, being increasingly successful and accepted.

Today, the online academic offer is growing considerably, not only as the solution to combine work and training, but the recent COVID19 pandemic has boosted this modality in places where it was previously unthinkable. Online learning is defined as "learning experiences in synchronous or asynchronous environments using different devices (e.g., mobile phones, laptops, etc.) with internet access. In these environments, students can interact with instructors and other students through the different platforms that the market offers [3].

It seems a long time ago, in 1995, when the UOC (Universidad Oberta de Catalunya) appeared in Spain and became the first online university in the world. Since then, the number of students has increased, reaching 900% growth from 2000 to 2018, according to the GAD3 (www.gad3.com), with a forecast of a multiplication of students by 10 in 2026, with growth in both bachelor's and master's students.

There are many advantages to online studies, which has the particularity that students can combine study with other activities, mainly work, and allow them to study anywhere, at a self-controlled rhythm and at any time, depending on the obligations and needs of each student, simply by needing an internet connection [4].

The extensive use of technologies in the current social panorama makes it possible to use technologies that can collect, analyzing and

* Corresponding author.

E-mail addresses: h.gerlache@gmail.com (H. Alonso-Misol Gerlache), pablo.moreno@unir.net (P. Moreno Ger), luis.delafuente@unir.net (L. de la Fuente Valentín).

extracting information from the data generated in all areas. In the educational panorama, this is no exception, and online universities are great generators of data thanks to the use of technological platforms that they use to reach all corners of the world. In fact, the online university studies mode favors the generation of data that allows us to carry out a subsequent study and analysis of the same in order to offer and improve all levels of learning and education in our society [5].

Educational institutions generate and collect huge amount of data. This may include students' academic records, their personal profile, observations of their behavior, their web log activities and faculty profile. This large data set is basically a storehouse of information and must be explored to have a strategic edge among the Educational Organizations [6]. The potential for data analysis in education must focus on developing robust applications that will improve student outcomes, enhance the pedagogy of instructors, improve the curriculum and increase graduation rates for all students, regardless of their background, from kindergarten to university. Today, higher education institutions face the critical challenge of retaining students and ensuring their successful graduation [7].

Today we have enough data to carry out an exhaustive analysis of them, through artificial intelligence techniques, to search for patterns within them that will allow us to improve our knowledge. In addition, the technological platforms integrated into our systems, such as educational ones, allow us to continue generating data that will provide knowledge about future situations. Whether through the processes of knowledge extraction from data (KDD) to the use of Machine Learning or Deep Learning techniques, thanks to the data we are able to make predictions with a high degree of certainty.

The use of this data must be focused on combating the major problems of education, and thus take advantage of the power of artificial intelligence. A problem related to higher education that concerns education authorities worldwide is the high rate of university dropouts. Data from the Spanish Ministry of Education, Culture and Sport (MECD, 2016) indicate that approximately one in five students drop out of university in the first year [8]. Of course, before making predictions, it is essential to find out an algorithm that is best suited for the problem, which requires comparison of algorithms based on certain metrics [6].

Data predictions are possible thanks to the algorithms, their use in educational environments is no exception [9]. As shown in this document, four algorithms with very high success rates are analyzed and compared in order to determine which of them is best suited to the dataset analyzed for grade prediction. These algorithms are Naive Bayes, Decision Tree, Random Forest and Neural Networks.

In this document we are going to try two different approaches in order to check whether they provide promising results. These two approaches are the prediction of the final Master's degree and the final grade of an exam. Both are going to be analyzed with the same dataset and in the conclusions phase, we are going to view the results of each other.

As we said, combating failure and early dropout from university is an issue of vital importance, especially because of the economic and social cost it generates [1], becoming an issue that has generated growing concern in recent years. Currently, many advances are being made, where Artificial Intelligence stands out as a powerful tool to help solve educational problems in the future, forming a scenario for improving educational quality, where technology, focused on the analysis of educational data, can be applied to prevent school failure and increase the quality of teaching, thus improving the student-teacher-university relationship.

The structure of the document is described below. After the introduction of this first chapter, the state of the question is addressed in Chapter II, where the background necessary to address the issue is explored in greater depth. Chapter III presents the main objective and secondary objectives, as well as a description of the methodology used. Chapter IV explains in detail the contribution of the paper and the experimentation carried out. Chapter V analyses the results and, finally, Chapter VI draws a final conclusion to the work and defines future lines of action for the continuation of the work.

II. State of the Art

The International Organization for Standardization stated in 2002 that the ability of educational institutions to manage their students on an individual basis was a key factor in achieving excellence in higher education [10]. This requires that teachers know the characteristics of their students and can guide them adequately to help them achieve their goals and avoid academic failure at the university [11]. This incipient need to improve the quality of education is the reason why many institutions are implementing learning platforms such as Blackboard, Moodle or Sakai, in order to offer their students a complete online platform, where the relationship between students, teachers and academic management is combined and managed. These systems present a learning opportunity that is delocalized and tailored to the interests of each student, and they are major generators of information which, using Artificial Intelligence techniques, can evaluate predictive models for different situations, such as student enrolment or grades [12]. However, this generated data, where a student's past academic history can be reviewed, is not a noise free source of information, which increases the complexity of the already complex problem regardless of the noise data [13], which degrades the quality or performance of the prediction, and it is necessary to discover the underlying correlation between the data and their degree of affectation [14]. In the field of education, Educational Data Mining (EDM) is taking advantage of the large amount of data in the sector, seeks to develop methods that discover the knowledge of data from educational environments [15], with the challenge of making good use of the data to improve the educational process [16]. The analysis of the prediction of grades and dropouts has led to research by different authors, which shows that one model is not better than another in a generalized way, but that the best prediction is given using a combination of models, such as neural networks, support vector machines and ensembles [17]. In the absence of concrete results from research on which algorithms are best suited to this type of problem, there are certain investigations in which the use of decision trees versus Bayesian or neural networks has yielded better results with relatively small data [18]. The analysis of university datasets references a problem with a search space of multiple parameters, of great diversity among them, and while some studies show better figures with the use of decision trees, others show better results using genetic algorithms, which confirms that at present there is no definitive study on the analysis of performance based on qualitative data of students, where it is determined which is the best model to carry out the analysis nor has it been found which of all the parameters of the students is the most influential on their academic performance [19]. Despite this paradigm, surprisingly good figures are being achieved that support the trend in the use of these techniques in the education sector. Thus, studies on prediction of results and university dropouts in the first year of electrical engineering have achieved accuracies of between 75% and 80% through the decision trees [12], even achieving predictions of over 96% to predict student grades before the final exam [15]. Machine learning for education has gained much attention in recent years, with a focus on predicting student performance [20], making clear the usefulness of Artificial Intelligence as a tool for predicting grades in the educational environment. Among the most used supervised learning algorithms in EDM, we find Naive Bayes, k-Nearest Neighbors, Decision Tree based algorithms [21], Random Forest, Support Vector Machines (SVM) and Neural Networks [16].

To the best of our knowledge, grade prediction is possible within the academic environment, but there is no obvious conclusion as either algorithm best suits the conditions of these data sets. Although, depending on the data and the treatment we give them, as well as on the configuration of the algorithms, different data will be obtained, this work provides a new study that compares the algorithms that have given the best results to date, with two clearly differentiated objectives. The first is to find out which of them best fits the data set and, subsequently, to find out which of them can predict students' grades, both for the final exam and for the master's.

III. Objectives and Methodology

Artificial Intelligence and Data Mining bring great possibilities to the field of predicting academic results. However, there are external factors that are not considered in this work, such as the socio-economic data of students, but we have a sufficiently broad set of data to address the problem of grade prediction at university environment, specifically in the University Master's Degree in Computer Security, since the data provided by the university correspond to four courses of this degree.

A. Main Objective

Contribute to the problem of grade prediction by analyzing and comparing different algorithms. The comparison of the algorithms will be done by determining which algorithm predicts more accurately and on which of the two lines of work, the prediction of the final master's degree or the prediction of the final grade of an exam.

In order to achieve objective, other intermediate milestones will need to be achieved, considerate as specifics objectives, as detailed below

B. Specific Objectives

1. Determine the feature extraction model that best fits the data set.
2. Determine the most influential characteristics in the student's academic outcome.
3. Compare the results of the two predictions to be made: final exam grade and final master's grade.
4. Perform training and validation of the selected A.I. algorithms.
5. Perform a comparative analysis of the results and technologies used in the prediction.
6. Predict the student's final exam grade and the master's degree grade.

C. Methodology

As we have seen in the main and specific objectives, the proposal of this work is to determine the best model of exploitation of the dataset for the prediction of students' grades. A methodology is proposed that follows the following five steps: Step 1 – Construction: Construction of a single dataset with the relevant information from each of the eleven files provided by the university. Step 2 – Cleaning: Starting from the single dataset, the data must be cleaned in order to eliminate the possible noise from the data, treating null values, missing data, identification of anomalous values, identification of out-of-range values and elimination of duplicated values. Step 3 – Relationship between the input characteristics: A statistical analysis of the data should be carried out to determine the behavior of the variables, comparing the means, standard deviations and quartiles of all the numerical variables, as well as the correlation between the dataset variables. Step 4 – Implementation of AI algorithms: Division of the dataset in two, so that one is prepared for the prediction of the master's degree grade and the other is prepared for the prediction of student's final exam grade. The following algorithms are considered and compared: Naïve Bayes, DecisionTree, RandomForest and Neural Networks. Step 5 – Analysis of results and conclusions: Finished the implementation of the AI algorithms, this step analyzes the results of each one of them according to the objectives 3 and 6, comparing their use for the two predictions that the work research. The document compares these four algorithms, as they are the best suited to this type of problem, as shown in the state of the art. All the steps could be viewed in the diagram of the Fig.1.

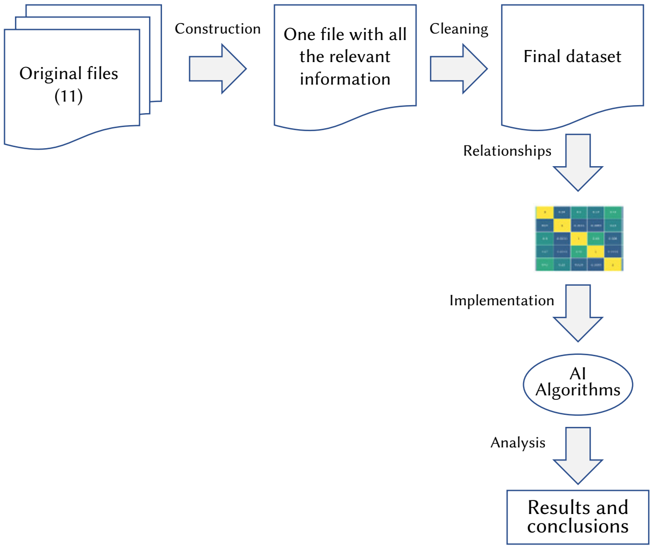


Fig. 1. Diagram of the methodology.

IV. Contribution

A. Data Preparation

The data to be analyzed comes from the LMS (Learning Management System) platform used by the International University of La Rioja (UNIR), corresponding to the University Master's Degree in Computer Security, courses from 2015 to 2018 in on-line mode. This information is provided in 11 different files that need to be unified in a single file, so that it can be analyzed and processed later. These files contain different information about the students and the grade, like the calcifications, evaluate elements, events, forums, users (students and teachers), messages, forums, information about all sessions, the relation of the tasks sent by each student, information related to the tasks and the topics to discuss in the forums. The number of rows and columns is described in Table I.

TABLE I. Number of Rows and Columns in Each File of the Dataset

Id	File	Rows	Columns
1	grades	49513	12
2	evaluate_elements	682	24
3	events	3350557	7
4	forums	197	27
5	users	699	1
6	messages	19230	28
7	rooms	65	4
8	sessions	675064	9
9	task_send	30318	7
10	tasks	315	3
11	topics	530	35

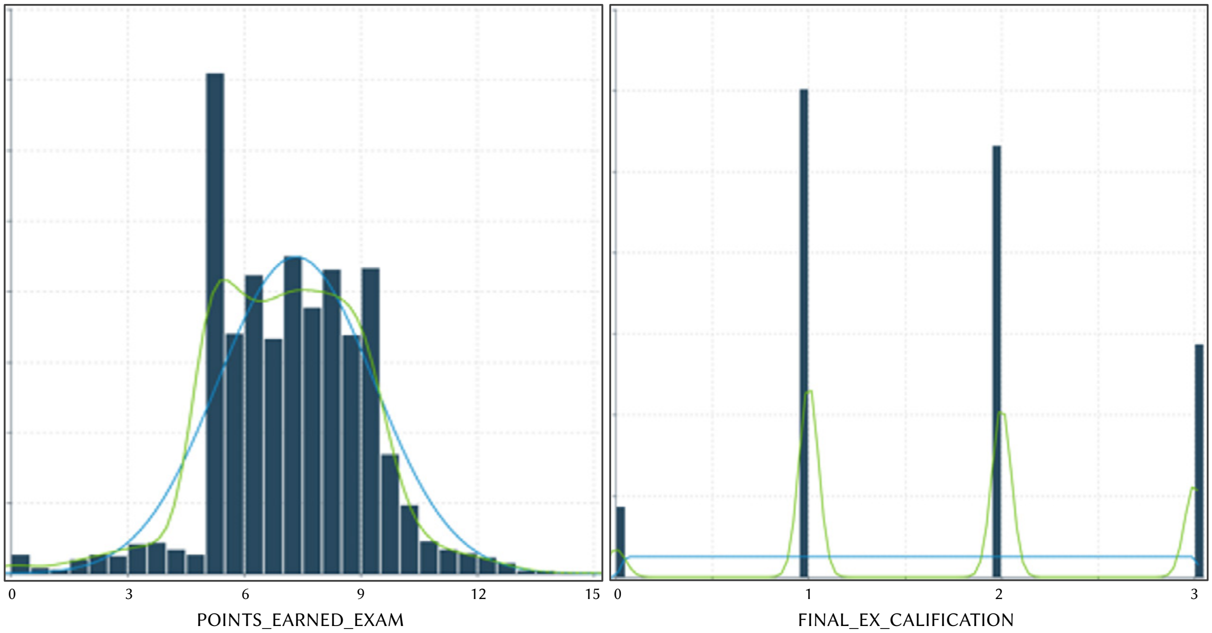


Fig. 2. Data distribution of the final exam grade.

It is important to keep in mind that the data to be worked with is data from student grades that are aseptic in terms of context. In this sense, all students are considered "equal", not according to personal or demographic data of the student, but only those data are collected that the university has as a result of learning under its model, being therefore all data of academic context. In the UNIR evaluation system, which is the origin of the dataset to be processed, there are two clearly differentiated blocks.

On the one hand, there is continuous evaluation based on evaluation activities, data, attendance at virtual classroom sessions and the performance of test-type tests. On the other hand, there is the final exam, which is the most important, and without which the subjects cannot be passed with a mark of more than 5.

The data guarantees the absolute anonymity of the data, not being able to identify any student through the data contained in the files that make up the dataset. With the 11 files, a merge has been realized in order to obtain a unique file with all the relevant data, deleting all those has no relevance in the objective of the scope of this document, like identifiers, versions, external links, etc.

The final dataset consists of 4522 records with student and subject data, and a total of 27 columns, which make up the set of input features for the AI models to be used in the machine learning algorithms. The description of each attribute could be seen at Table II.

B. Analysis Process

It is necessary to identify and understand the behavior of the predictor variables, which according to the proposed objectives 3 and 6. Their behavior can be observed in the Fig. 2 and Fig. 3, where a Gaussian distribution and its categorized correspondence can be seen.

For data analysis, Python 3.7 is used as the main tool for data processing and algorithm generation, in addition to the Watson Studio tool.

Null value analysis (missing values). For each one of the columns of the dataset, the percentage of null values is determined, eliminating directly all those columns that show null data in a percentage equal or superior to 25%. For the rest of the null values, which are less than 25%, the null value is replaced by the average of the column.

TABLE II. Description of the Final Dataset

Attributes description
Student ID.
Subject studied by the student.
Sum of points obtained in continuous assessment.
Possible points in the continuous assessment.
Number of evaluable activities.
Possible points in the course.
Number of the course activities
Number of the course events
Number of the course sessions
Number of the course messages
Number of the read course messages
Number of the evaluable course messages
Number of the read evaluable course messages
Number of the evaluable task and events
Number of the sent evaluable activities.
Points obtained in the continuous assessment.
Possible points of the continuous assessment.
Relation between the earned points of the continuous assessment and the possible points of the continuous assessment.
Average_cont is the 40% of the points earned in the continuous assessment.
Points earned in the exams.
Maximum number of points that could be earned in exams.
Relation between the earned points of the exams and the maximum number of points that could be earned in exams.
Average_exam is the 60% of the points earned in the exams.
Final grade, as the sum of the weighted fields.
Grade description: 4 ⟶ Honourable mention, 3 à ⟶ Merit, 2 ⟶ Notable, 1 ⟶ Pass, 0 ⟶Fail
Final exam grade for each student

Processing of out-of-range values. There are grade values of students with scores above 10, which are the result of having taken the ordinary assessment test and the remedial test. For these values, 187 out of the total of the dataset, which represents 4.1%, the exam grade is determined according to the final grade, putting a 4, 6, 7 or 9 for the final grades of Suspended, Pass, Notable and Merit respectively.

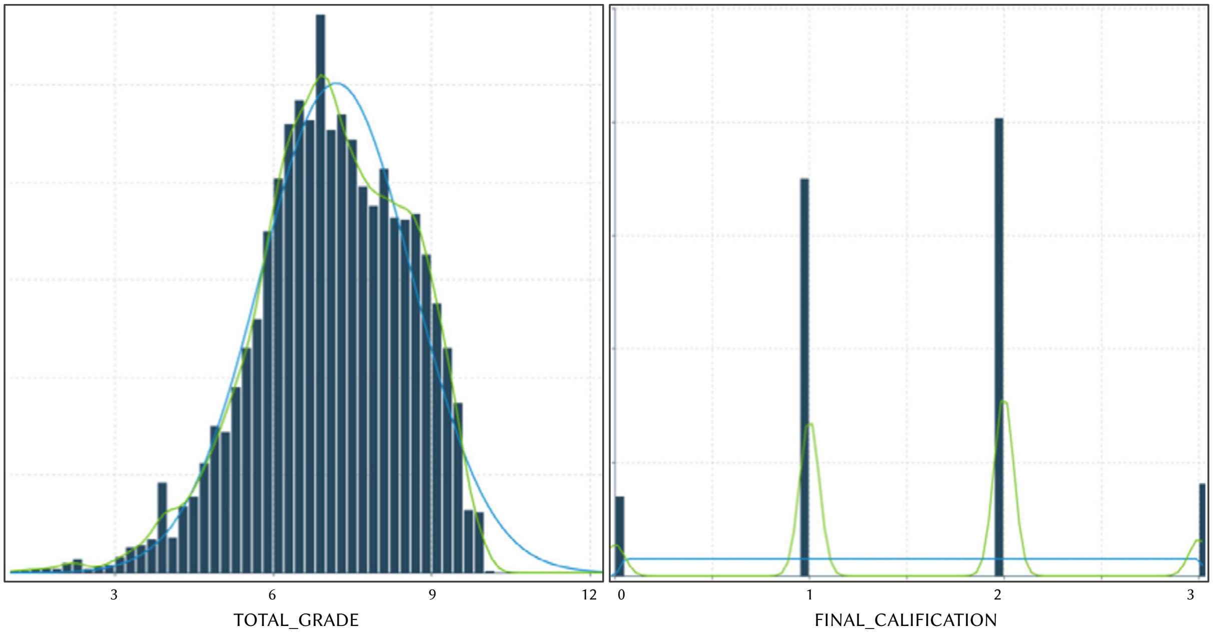
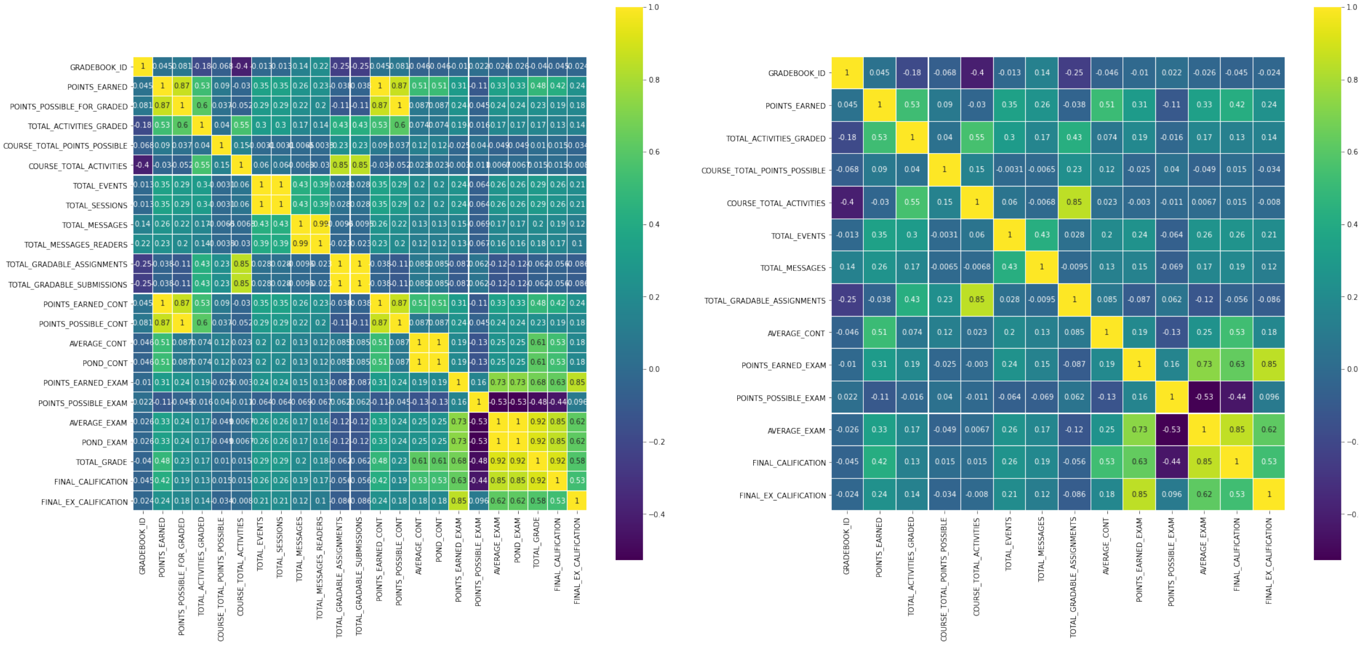


Fig. 3. Data distribution of the master's degree grade.



(a)

(b)

Fig. 4. Correlation of model characteristics, (a) before and (b) after treatment.

Analysis of the correlation of variables. It is important to analyze the degree of correlation between variables, in order to determine which of them do not contribute information to the model, incurring a problem of consumption of unnecessary time and resources. To do this, the correlation matrix of the variables is obtained, as can be seen in Fig. 4.

In this dataset, all those variables that exceed 75% of correlation have been eliminated, having to eliminate a total of 11 input characteristics.

Statistical description of the data. Once the noise has been removed from the data, the data is checked from a statistical perspective, analyzing the mean, standard deviation, minimums, maximums and quartiles, in order to detect possible outliers.

Different behavior of the variables can be observed, as it is the case of Fig. 5, where values out of range are observed and outliers must be solved before proceeding to their use in the Artificial Intelligence algorithms, as can be viewed in Fig. 6.

Duplicate elimination. Once the model's input feature set is clean, it is necessary to ensure that there are no duplicate records. At this point, all duplicate records are checked and removed if necessary.

The objective will always be to provide a solution to a classification problem, where the output of the algorithms will be the probability of obtaining a prediction of the final grade of the exam or of the final qualification of the master, being in both cases a prediction of between four possible values: 0 (fail), 1 (pass), 2 (notable) and 3 (merit). The algorithms Naïve Bayes (BernoulliNB model), Decision Tree (DecisionTreeClassifier model), Random Forest

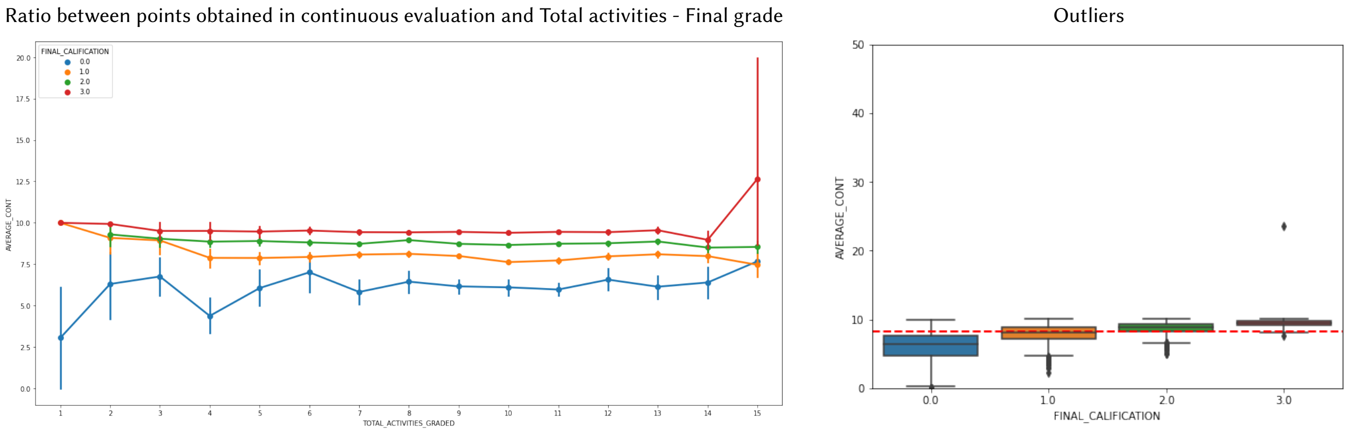


Fig. 5. Behavior of variables where abnormal values are detected.

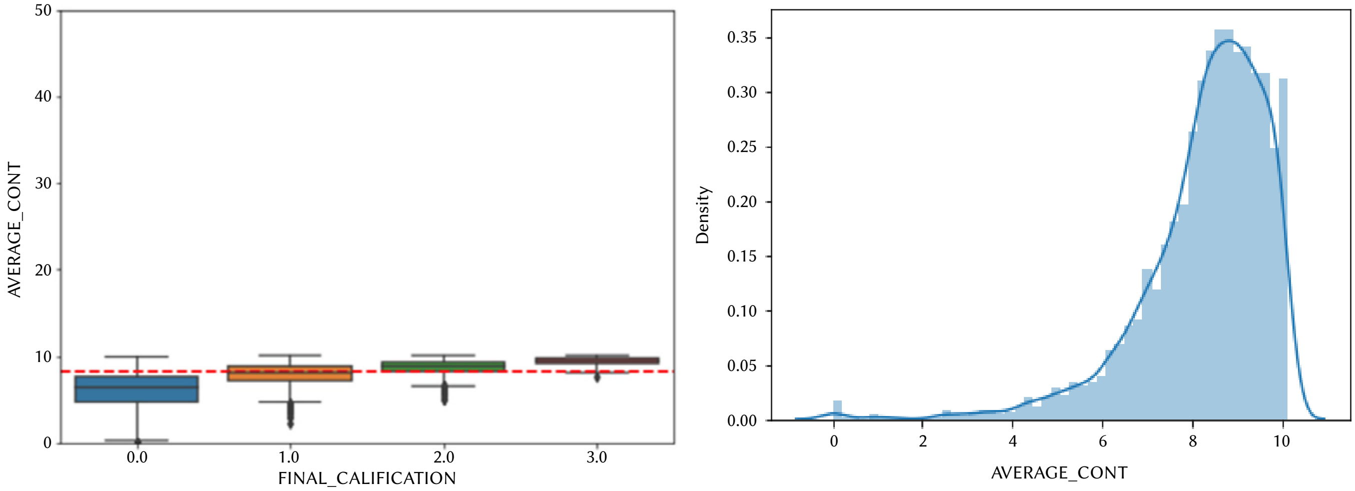


Fig. 6. Abnormal data removal.

(RandomForestClassifier model) and Neural Networks (Sequential model with two hidden layers, Relu activation, Adam optimizer and output activation function for the four Softmax classes) have been parameterized and used.

V. Analysis of Results

Note that under the same input data set, a double classification problem is being addressed. On the one hand, the problem of predicting the final grade of an exam and, on the other hand, the prediction of the final master's degree course. For both predictions, the same configuration of the algorithms will be used, so that the comparison is homogeneous. The comparison of results obtained can be seen in the tables of this section, where the results of each of the algorithms used for each of the predictions made are shown, differentiating between Machine Learning algorithms (ML) and Deep Learning algorithms (DL).

The configuration of the Naïve bayes algorithm configuration is described in Table III.

The configuration of the Decision Tree algorithm configuration is described in Table IV.

The configuration of the Random Forest algorithm configuration is described in Table V.

TABLE III. Naive Bayes Algorithm Configuration

Naive Bayes		
Parameter	Value	Range
Model	BernoulliNB	BernoulliNB
Alpha	1.0	0.5-1.0
Binarize	True	True-False
Fit_prior	False	False
Class_prior	None	None

TABLE IV. Decision Tree Algorithm Configuration

Decision Tree		
Parameter	Value	Range
Model	DecisionTreeClassifier	DecisionTreeClassifier
criterion	entropy	entropy
min_samples_split	20	10-30
min_samples_leaf	4	4-10

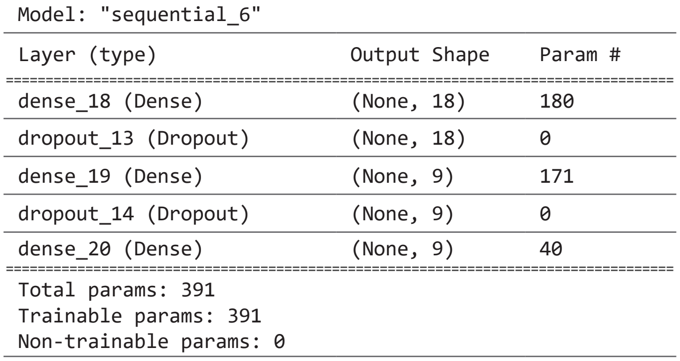
TABLE V. Random Forest Algorithm Configuration

Random Forest		
Parameter	Value	Range
Model	RandomForestClassifier	RandomForestClassifier
bootstrap	True	True
criterion	gini	gini
n_estimators	20	10-100

The configuration of the Neural Network algorithm configuration is described in Table VI and Fig. 7.

TABLE VI. Neural Network Configuration

Red Neuronal		
Parameter	**Value**	**Range**
Model	Sequential	Sequential
Input	9 dimensions	9 dimensions
Hide layers	2	2-5
Output layer	1	1
Optimizer	Nadam	Nadam, Adam, sgd,
Loss	mean_squared_error	mean_squared_error
Metrics	Accuracy	Accuracy
Activation function. Hidden layers.	Relu	Relu,tanh
Activation function. Output	Softmax	Softmax
Batch_size	20	10-80
Epochs	300	10-500
Num_classes	4	4



```
Model: "sequential_6"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_18 (Dense)             (None, 18)                180
_________________________________________________________________
dropout_13 (Dropout)         (None, 18)                0
_________________________________________________________________
dense_19 (Dense)             (None, 9)                 171
_________________________________________________________________
dropout_14 (Dropout)         (None, 9)                 0
_________________________________________________________________
dense_20 (Dense)             (None, 9)                 40
=================================================================
Total params: 391
Trainable params: 391
Non-trainable params: 0
_________________________________________________________________
```

Fig. 7. Fully-connected Neural Network configuration.

It can be seen that the student's final exam grade is easier to predict than the master's degree grade, which shows that the work of continuous assessment is clearly reflected in the final exam grade. In the case of the final exam grade, results are achieved with an accuracy of 96%, while in the master's degree grade, maximum figures of 70% accuracy are achieved.

In the prediction of the test score, in Table VII, the algorithms that have made the best prediction are Decision Tree and Random Forest, all exceeding a 75% prediction, where the worst result has been Naive Bayes.

TABLE VII. Comparative Table of Results

Algorithm	Type	Master Result	Exam Result
Naïve Bayes	ML	63%	76%
Decision Tree	ML	68%	96%
Random Forest	ML	70%	96%
Neural Networks	DL	62%	81%

It is also important to note that the neural network has a much higher cost of configuration and execution than Random Forest, so, under this configuration, it is convenient to go deeper into Random Forest than into the Neural Network. Regarding the prediction of the final grade of the Master, it is a much more complex prediction, since all the subjects and their results must be taken into account, both in continuous evaluation and in the final exam, but whose relationship is not as direct as in the prediction of the exam grade. In this case, once again the model that has worked best is Random Forest, with 70% correct predictions, while the Neuronal Network has the lowest accuracy, with 62% correct predictions, being significantly worse than Naïve Bayes. The algorithms used give really good figures to be a first approximation, suggesting that the data processing is correct and the methodology appropriate.

VI. Conclusions and Future Lines

A. Preamble

Predicting the grades of students is a powerful tool that helps the student and the university in a remarkable way, and it is a reality today. Artificial Intelligence has put the strings on so that we are able to predict and infer future situations. To make these predictions we need order, data, method and tools that enable us to make them. Thanks to the dataset provided by the university, today we have a set of data from the University master's in computer security, which has allowed us to undertake this project successfully. On the data obtained directly from the LMS (Learning Management System) platform, we have been able to compose a unique set of data with which we have been able to carry out the comparisons of the Artificial Intelligence techniques that best fit, based on a proposed methodology. With the help of tools such as Watson Studio and Python, it has been possible to obtain a multipurpose dataset, which allows its use in algorithms for the prediction of student's final exam grade and the master's degree grade. It's not a simple or fast task, nor is it problem-free, but in the end, a coherent and tangible comparison has been achieved. Through the proposed model, and using Random Forest as a prediction tool, figures of over 95% correct prediction have been obtained and, what is more remarkable in comparison to the objectives of the present work, we have a well-founded comparison of the algorithms used and the proposed methodology, which enable the original dataset to be used for this purpose.

B. Summary of Contributions

The present work has made a comparison of Artificial Intelligence algorithms with the aim of addressing the problem of grade prediction in university environments, seeking to contribute to this problem by providing new information on the treatment of such classification problems in a very defined environment. The following contributions can be identified in Table VIII.

In the State of the Art we have addressed the question of which algorithm is best suited to this type of prediction, demonstrating that there is no one algorithm that is clearly better than another, but that it will depend to a large extent on the data and the treatment that is carried out. With this work, we have been able to verify that with decision tree we have obtained results of 96% accuracy, which gives grounds to continue working with this algorithm, as suggested in future work.

C. Future Work

The results and conclusions obtained in this work present an opportunity to continue working on the prediction of academic grades in UNIR students. Within the future work that can be done, as future lines that can take this work as a basis, the following actions are proposed: i. To deepen in the parameterization of the proposed algorithms, with special focus in Random Forest, in search of higher prediction values than those obtained in this work. Although very high values have been obtained in the student's final exam grade prediction, the same figures are not obtained in the master's degree grade, so there is an exciting field of research in this regard. ii. Increase the dataset with data from more years and make the model and methodology proposed to see how it behaves. In this sense, it would be interesting to train the model with all the available data and use the current year to make an inference of results, thus validating the model. iii. To increase the dataset with demographic data of the students, which would allow to extend the scope of the study and to be able to conclude in more directions, such as the impact of the family situation on the academic performance, curricular adaptations, new support subjects, etc. iv. Test the same methodology and algorithms in different UNIR studies to see if it can begin to evolve towards generalization at the University.

TABLE VIII. Summary of Contributions

Contribution	Description	Tangible
Dataset collection	Generation of a single dataset from the 11 files provided by the SAKAI platform	The dataset of the SAKAI platform has been obtained and a working methodology has been presented in order to address the problem. Finally a dataset with the 27 most significant characteristics and 4522 records has been obtained
Dataset increase	Generation of the qualification master degree grade from the data of course of each student	Obtaining the characteristic variable, called FINAL_CALIFICATION
Dataset cleaning	Removal noise from the input variables of the final dataset	The treatmen of the data carried out, before being used by the AI algorithms, is exposed
Implementation of AI algorithms	Implementation of AI algorithms uset to contribute to the problem of university grade prediction (classification problem)	The implementation of the algorithms used in this work can be found and used freely through the following link: https://github.com/HectorAMG/Algoritmos-IA
Algorithms comparision	Comparison of the results obtained	Identification, configuration, use and comparative results of the use of the different algorithms to address the problem of predicting grades of university students

References

[1] A. Elbadrawy, A. Polyzou, Z. Ren, M. Sweeney, G. Karypis, and H. Rangwala, “Predicting Student Performance Using Personalized Analytics”, *Computer*, vol. 49, no. 4, pp. 61–69, 2016.

[2] L. Gerritsen, “Predicting Student Performance with Neural Networks,” Ph.D. dissertation, School of Humanities, Tilburg University, Tilburg, Netherlands, 2017.

[3] Y. Jiang, R. S. Baker, L. Paquette, M. San Pedro, & N. T. Heffernan, “Learning, moment-by-moment and over the long term”, in International Conference on Artificial Intelligence in Education, Madrid, Spain, 2015, pp. 654–657, doi: https://doi.org/10.1007/978-3-319-19773-9_84

[4] T. Mishra, D. Kumar, & S. Gupta, “Students’ employability prediction model through data mining”, International Journal of Applied Engineering Research, vol. 11, no. 4, pp. 2275–2282, 2016.

[5] V. Singh & A. Thurman, “How Many Ways Can We Define Online Learning? A Systematic Literature Review of Definitions of Online Learning (1988-2018),” American Journal of Distance Education, vol. 33, no. 4, pp. 289–306, 2019.

[6] R. Stillwell, & J. Sable, “Public School Graduates and Dropouts from the Common Core of Data: School Year 2009–10”, National Center for Education Statistics, US Department of Education, USA, 2013. Accessed: Feb. 15, 2019. [Online]. Available: https://nces.ed.gov/pubs2013/2013309rev.pdf.

[7] R. J. Sternberg. “Teaching College Students that Creativity Is a Decision”, Guidance & Counselling, vol. 19, no. 4, pp. 196-200, 2004.

[8] J.M. Tomás & M. Gutiérrez, “Aportaciones de la teoría de la autodeterminación a la predicción de la satisfacción escolar en estudiantes universitarios”, Revista de Investigación Educativa, vol. 37, no. 2, pp. 471–485, 2019.

[9] C. J. Villagrá-Arnedo, F. J. Gallego-Durán, F. Llorens-Largo, R. Satorre-Cuerda, P. Compañ-Rosique, & R. Molina-Carmona. “Time-Dependent Performance Prediction System for Early Insight in Learning Trends”, International Journal of Interactive Multimedia and Artificial Intelligence, vol. 6, no. 2, pp. 112-124, 2020, doi: 10.9781/ijimai.2020.05.006.

[10] V.M. Cojocariu, I. Lazar, V. Nedeff, & G. Lazar, “SWOT Analysis of E-learning Educational Services from the Perspective of their Beneficiaries”, Procedia-Social and Behavioral Sciences, vol. 116, pp. 1999–2003, 2014, doi: 10.1016/j.sbspro.2014.01.510.

[11] P. Colás Bravo, “El abandono universitario”, Revista Fuentes, no. 16, pp. 9–14, 2015, doi: 10.12795/revistafuentes.2015.i16.

[12] S. Regha R. & D. U. Rani, “An Efficient Clustering Based Feature Selection for Predicting Student Performance”, International Working Group on Educational Data Mining, vol. 9, no. 2, pp. 524–531, 2017, doi: 10.21817/ijet/2017/v9i2/170902328.

[13] G. W. Dekker, M. Pechenizkiy, & J. M. Vleeshouwers, “Predicting students drop out: A case study”, in International Working Group on Educational Data Mining 2009, Córdoba, Spain, 2009, pp. 41–50.

[14] Q. Hu, A. Polyzou, G. Karypis, & H. Rangwala, “Enriching course-Specific regression models with content features for grade prediction”, in 2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA), 2017, pp. 504–513. doi: 10.1109/DSAA.2017.74.

[15] I. Lykourentzou, I. Giannoukos, V. Nikolopoulos, G. Mpardis, & V. Loumos, “Dropout prediction in e-learning courses through the combination of machine learning techniques”, Computers & Education, vol. 53, no. 3, pp. 950–965, 2009, doi: 10.1016/j.compedu.2009.05.010.

[16] J. Xu, K. H. Moon & M. van der Schaar, “A Machine Learning Approach for Tracking and Predicting Student Performance in Degree Programs,” in IEEE Journal of Selected Topics in Signal Processing, vol. 11, no. 5, pp. 742-753, 2017, doi: 10.1109/JSTSP.2017.2692560.

[17] R. Heredia, J. Jobany, H. Rodríguez, G. Aida & J.A. Vilalta, “Predicting performance in a subject using ordinal logistic regression”. Estudios pedagógicos (Valdivia), vol. 40. no 1, pp. 145–162, 2014. doi: 10.4067/s0718-07052014000100009.

[18] J. G. Cleary and L. E. Trigg, “K*: An Instance-based Learner Using an Entropic Distance Measure” in Machine Learning Proceedings, M. Kaufmann Publishers, 1995, pp. 108–114. doi:10.1016/b978-1-55860-377-6.50022-0.

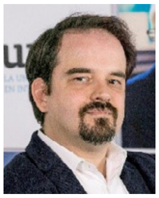
[19] T. Miranda Lakshmi, A. Martin, and V. Prasanna Venkatesan, “An Analysis of Students Performance Using Genetic Algorithm”, Journal of Computer Sciences and Applications, vol. 1, no 4, pp. 75-79, 2013, doi: 10.12691/jcsa-1-4-3.

[20] E. Osmanbegovic, M. Suljic, “Data Mining Approach for Predicting Student Performance” in Journal of Economics and Business, University of Tuzla, Faculty of Economics, vol. 10, no. 1, pp. 3-12, 2012. [Online] Available: http://hdl.handle.net/10419/193806.

[21] A. Hamoud, A. S. Hashim, & W. A. Awadh, “Predicting student performance in higher education institutions using decision tree analysis”, International Journal of Interactive Multimedia and Artificial Intelligence, vol. 5, no. 2, pp. 26-31, 2018, doi: 10.9781/ijimai.2018.02.004.

Héctor Alonso-Misol Gerlache

Héctor Alonso-Misol Gerlache was born in Madrid in 1976, he has been dedicated to the IT environment for 20 years. Besides his family, he has three passions, technology, education and people, which are reflected in the company he founded in 2020, GradientIA TMC, S.L., providing agile services of SW development with Artificial Intelligence technologies.



Pablo Moreno Ger

Dr. Moreno-Ger was born in Madrid in 1981. He finished his doctorate in Computer Engineering from Universidad Complutense de Madrid (UCM) in 2007 and was an Associate Professor in the Department of Software Engineering and Artificial Intelligence at UCM. Now he is with Universidad Internacional de La Rioja (UNIR), where he is currently the Vice-Rector for Research. Formerly, he was the Director of the School of Engineering and Technology at UNIR, as well as Vice-Dean for Innovation at the School of Computer Engineering at UCM. His main research interests are in technology-assisted teaching, artificial intelligence, learning analytics and serious games. He has published more than 150 academic works in these fields.

Luis de la Fuente Valentín

Luis de la Fuente Valentín is a full-time associate professor at Universidad Internacional de La Rioja, UNIR. He got his PhD at Universidad Carlos III de Madrid, in 2011. He has authored more than 40 papers and participated in several national and European public funded projects, one of them as investigator in charge. His current research interest is on machine learning tools applied to the educational field.

Predictive Model for Taking Decision to Prevent University Dropout

Argelia B. Urbina-Nájera*, Luis A. Méndez-Ortega

UPAEP-Universidad (México)

Received 17 December 2020 | Accepted 2 June 2021 | Published 25 January 2022

unir
LA UNIVERSIDAD
EN INTERNET

Abstract

Dropout is an educational phenomenon studied for decades due to the diversity of its causes, whose effects fall on society's development. This document presents an experimental study to obtain a predictive model that allows anticipating a university dropout. The study uses 51,497 instances with 26 attributes obtained from social sciences, administrative sciences, and engineering collected from 2010 to 2019. Artificial neural networks and decision trees were implemented as classification algorithms, and also, algorithms of attribute selection and resampling methods were used to balance the main class. The results show that the best performing model was that of Random Forest with a Matthew correlation coefficient of 87.43% against 53.39% obtained by artificial neural networks and 94.34% accuracy by Random Forest. The model has allowed predicting an approximate number of possible dropouts per period, contributing to the involved instances in preventing or reducing dropout in higher education.

Keywords

Artificial Neural Network, Decision Trees, Dropouts, Educational Data Mining, Higher Education.

DOI: 10.9781/ijimai.2022.01.006

I. Introduction

DROPOUT in higher education is a difficult topic to explain; for this reason, [1] proposes to analyze the phenomenon from different perspectives: student, institutional, and state or national. From the student's perspective, there are expectations, goals, intellectual capacities, and socio-economic origin. In the institutional aspect, those options offered to students to include them in university life follow them up and retain them. At the state or national level, desertion must be considered the interruption of studies in any modality, and the policies that promote retention must be analyzed.

In Mexico, 38% of the people who can access higher education do not graduate, which is why the OECD (Organization for Economic Cooperation and Development) places Mexico and Turkey (with the same percentage) as the countries with a severe problem in terms of school dropouts. This percentage contrasts with Germany and Finland, where they have 4.03% and 0.45%, respectively [2]. Additionally, the effects of dropping out are reflected in labor and social inequality because the probability of finding better jobs with greater privileges is permeated by this problem [3]. According to OECD data (2018), 85% of people with higher education (25 to 64 years old) are employed, compared to 75.2% of those with only an average higher education, which shows that there are more significant job opportunities for those who advance and complete their academic studies. In Mexico's case, this proportion is 80% for those with higher education and 70.6% for those with lower education, which is below the OECD average [4].

On the other hand, according to the report on Higher Education in Mexico (2018), by the OECD, the hiring of young people between 25-34 years of age with higher education is 80.7%, which is less than 84.1% of the average of the other member countries. Thus, [3] indicates that reducing student dropout in higher education impacts positively by promoting a society better prepared to meet global challenges in which we are immersed and improve people's quality of life, get better jobs, wages, opportunities for intellectual growth, among others.

In [5], authors comment that academic desertion results from several factors such as personal situation, educational quality, facilities, socio-cultural and economic factors. This causes students to have to interrupt their studies and, therefore, affect their academic life, the institution, and society. In [1] the basis to face this problem is established and it suggests analyzing the factors in personal, institutional, and state perspectives, the first efforts to understand the desertion phenomenon was directed to explain the factors that trigger it. Recently, in [6], authors suggest applying for the latest advances in information and communication technologies (ICTs) and data science to explain this situation, not only to detect explanatory factors, but also to create predictive models that prevent it and, thus, to make decisions that reduce the mentioned indexes.

This study aims to find a model based on computational learning algorithms (decision trees and neuronal networks) that anticipates university desertion aimed at reducing the desertion rate in degree programs in Engineering, Social and Administrative Sciences.

This experimental study was based on educational data mining methodologies and computational learning algorithms such as neural networks and decision trees. The document was organized as follows: Section II describes the related works that give theoretical support to the study. Section III details the decision tree and neural network algorithms. Section IV presents the results obtained, and section V details the conclusions and future work from various approaches.

* Corresponding author.

E-mail address: argeliaberenice.urbina@upaep.mx

II. Related Work

This section presents related work divided according to the method used to predict college dropout. For example, in [7][8][9][10][11], they used a statistical analysis to perform an analysis of social implications and preventive actions [8], determine actions to prevent dropout during the first semester [9] or identify factors influencing university student satisfaction, dropout, and academic performance [7][10][11].

On the other hand, various methods have also been used, such as retention theory, to identify whether the student's perspective, the institution, and the state influence university dropout [1]. In [6], heuristic and projective analysis were applied to analyze whether economic differences, vocation, attitudes, and expectations influence dropout. Also, in [5], the non-probabilistic and propositional method was used to determine whether the economic, school, and institutional core variables determine the dropout decision.

Recently, however, artificial intelligence, data mining, and computational learning methods have gained great importance in predicting college dropouts. For example, commonly used methods for predicting college dropout are: neural networks [12]-[16], K-nearest neighbors and logistic regression [14],[17], random forest [14],[17], Bayesian networks [18], decision trees [16],[19],[20], support vector machines [21], [29], statistical methods [22]-[27] and finally, deep learning in [28].

A summary of the results obtained for the best classifiers with different methods is presented below. A work [12] proposes a data mining application to make a prognosis of desertion in higher education students. The data used comprises 2007 to 2014, with 421,282 records, which the university's data warehouse provided. They analyzed data such as age, gender, location, level of studies of the tutors, year of entry to the career, and subjects failed and approved. To generate the model, they used Microsoft Azure Machine Learning's cloud service with the algorithm of Two-Class Bayes Point Much and Neural Networks. Finally, the model had an accuracy of 66%, which allowed concluding that the forecasts' results must be taken with particular caution since, although they can be improved, several factors may not be considered to assume that a way to forecast dropout was found.

In [8], authors used 5,288 student records from four generational cohorts and a decision tree model was implemented in RapidMiner Studio, with demographic variables, economic status, and some data collected at the time of entry, such as knowledge test scores. The tree used had a maximum depth of 20 and an accuracy of 87.27% to detect three factors that explain dropout: grade point average, progression period, and entrance exam score, which encourages the use of decision tree algorithms to counteract student dropout. Finally, Table I presents the works analyzed whose efforts are focused on detecting those factors that affect dropout, and few are applying computer learning methods to make a prognosis.

On the other hand, another study [13] analyzes the performance of Random Forest, Neural Networks, Support Vector Machines, and logistic regression in order to predict college student dropouts. They found that Random Forest was the best predictor by obtaining 91% of correctly classified dropouts with a sensitivity of 87%. In the study, they used a set of 80,527 records and 21 variables classified in the categories: dropout (3), demographic (4), program (7), and academic history (7).

One of the best results obtained in the analyzed articles was achieved in [14], with a data set of 61,340 records and 18 variables it was obtained a recall of 92.4% and sensitivity of 68.6% by applying logistic regression. They determined that out of the four classifiers used (Random forest, logistic regression, K-nearest neighbors, and neural networks), the classifiers logistic regression and the neural network had proven superior to all the other classifiers' highest performance metrics when the over-sampling technique was employed.

Finally, in [15] authors used a set of 2,670 records and 11 variables, applied neural networks with basis radial function and perceptron multilayer; with the first obtained an accuracy of 96.8% in training and 98.1% with test data, while with the latter achieved 96.3% with training data and 98.6% with test data, determining that the neural networks provide a model that helps to determine university dropout and with it, contributed institution administrators to make decisions before a dropout occurs.

The analysis presented in this section provides a guideline for

TABLE I. Abstract to Related Works

Year	Description	Method
Tinto (1989) [1]	Perspectives: student, institutional and state.	Retention theory
Rodríguez y Hernández (2008) [42]	Factors: work, family economy, school performance, study and orientation.	Survey and qualitative analysis
SEP (2011) [8]	Analysis of social implications and preventive actions	Statistical analysis
Moine (2013)[22]	Evaluation of methodologies and software	KDD, CRISP-DM, SEMMA y KATALYST
Report PEM 2016 [2]	Preventive actions in the first semesters	Statistical analysis
Ramírez, Espinosa y Millán 2016 [6]	Economic differences, vocation, attitudes, expectations	Heuristics and projective scope
López y Beltrán 2017 [5]	Economic, school and institutional core	Non-probabilistic and propositional
Chinkes 2017 [12]	Demographic data analysis	Machine Learning, algorithm Two-Class Bayes Point Much and artificial neural network
Carvajal, González y Sarzoza, 2017 [23]	Institutional data analysis	Descriptive, correlational and inferential statistics
Cendejas, Acuña, Cortez y Bolaños, 2017 [24]	Evaluation of methodologies and software	CRISP-DM y SEMMA
Zavala, Álvarez, Vázquez, Gonzales y Bazán, 2018 [25]	Internal student, external and bilateral factors	Correlation factors
Muñoz-Camacho, S., Gallardo,T. Muñoz-Bravo, M. y Muñoz-Bravo, C., 2018 [26]	Institutional data analysis	Logit Discrete Choice Model
Gallegos, Campos, Canales y González, 2018 [27]	Institutional data analysis	Logit probability model
Ramírez y Grandón, 2018 [19]	Analysis of demographic, economic and other data	Decision trees
Villagrá-Arnedo, et al. 2020 [29]	Performance prediction	Support Vector Machine

choosing the computational learning methods to be applied. In this case, we chose to use oversample and undersample techniques, to have a balanced data set [14],[31], as well as neural networks despite being used in [12]-[16] with a smaller number of records in some cases, our challenge is to use a more extensive data set covering an analysis period of 9 years classified in 3 areas of knowledge (engineering, social sciences, and administrative sciences) to obtain equal or better results in accuracy and sensitivity. The random forest mentioned in [14],[17] will also be applied, with the contribution in this study of using different attribute selection methods that are explained in later sections and that are not described in the analyzed articles.

III. Educational Data Mining

Educational Data Mining (EDM) is an area of computer knowledge focused on creating methods to examine the unique types of data that come from large volumes of data in educational environments to provide answers to educational questions or improve educational or administrative processes automated manner. EDM methods are drawn from various areas, including data mining, computer learning, psychometrics, statistics, information visualization, and computer modeling [31]. Currently, some algorithms have been applied in various real-world contexts to provide solutions with high precision, to mention a few, in improving a person's response time (Neurobiology), molecular regulation of alpha-viruses (Molecular Biology), geographic sales trends (Finance), monitoring of manufacturing processes (Control), classification and characterization of patients (Medicine).

These algorithms include decision trees, k-means, vector support machines, artificial neural networks, Bayesian learning, instance-based methods, and Bayesian models. In this section, decision trees and artificial neural networks are briefly described, and the metrics used to evaluate their performance.

A. Decision Trees

Decision tree algorithms are supervised learning techniques, easy to implement and very useful, composed of a single initial node and underneath other independent trees that indicate the predictive attributes [32]. The decision trees are located within a branch of automatic learning called symbolic learning, in which there are also decision rule models closely related to the trees.

Learning using decision trees is a technique that allows the analysis of sequential decisions based on the use of results and associated probabilities. An article [33] defines it as "A method of approximation of an objective function of discrete values in which a decision tree represents the objective function. Learned trees can also be represented as a set of rules ...". On the other hand, they are among the most widely used inductive learning methods in inductive inference algorithms and have been successfully applied to learning how to diagnose medical cases and assess credit risk in loan applications. It should be noted that some of the applications of this algorithm are Binary searches, expert systems, medical diagnostics, scheduling, risk analysis, among others [30].

The decision tree algorithm C5.0 (in its commercial version known as C4.5) is an extension of ID3. It can work with continuous values for the attributes, separating the possible results into two branches. The trees it generates are less leafy because each leaf does not cover a particular class but a class distribution. C5.0 forms a decision tree from the data employing recursively executed partitions, according to the depth-first strategy. Before each data partition, the algorithm considers all possible tests that can divide the data set and select the test results in the highest information gain or the highest information gain ratio. For each discrete attribute, a test with n results is considered, where n is the number of possible values the attribute can take [33].

B. Artificial Neural Network

Artificial neural networks (ANN) are computer models that try to mimic the neurons in the human brain and solve complex learning problems. They are composed of algorithms that process a set of data to find non-linear relationships. They can learn and improve their functioning [32]. The simplest type of ANN is the so-called perceptron, which takes a vector of real values as input, calculates a linear combination of these inputs, and produces a value (usually 0 or 1) according to a function. A typical ANN is formed by interconnected neurons arranged in three layers (this may vary). The data entry through the input layer passes through the hidden layer and exits through the output layer. It is worth mentioning that the hidden layer can create several layers. In other words, neurons organize in layers (monolayer and multilayer), and the output of some neurons are the inputs of other neurons, then they are forward (feedforward) if they have connections backward, then they are feedback [30].

Therefore, an artificial neural network architecture is the structure or pattern of network connections, usually grouped into structural units called layers; within a layer, neurons can be of the same type. This architecture includes three layers, the input layer that receives data or signals from the environment, the output layer that provides the network response to input stimuli, and the hidden layer that does not receive or provide information to the environment, and then they are used as internal network processing.

C. Performance of Algorithms

To estimate and compare the algorithms' performance, one of the techniques used is based on the confusion matrix (Table II).

TABLE II. Confusion Matrix

Real Values	**Prediction**	
	Positive	Negative
Positive	a	b
Negative	c	d

From Table II the metrics accuracy, precision, specificity, and recall are derived. Accuracy: percentage of true positives and negatives against all data. Precision: percentage of true positives out of the total number of classified positives. Specificity: of the true negatives, how many did you classify correctly?. Recall: of the true positives, how many did you classify correctly?.

Similarly, a study [35] recommends using the Matthew Correlation Coefficient (MCC) as a metric to evaluate the performance of binary classification models globally. It evaluates in a range of -1 to 1, where 1 represents the perfect classification, 0 a random classification, and -1 an inverse classification. Therefore, the MCC and balanced accuracy are considered in the performance evaluations. The MCC will be represented in percentage for practical purposes. The metrics to evaluate the algorithms' performance will be the same, to be able to compare and select the model with the best results.

IV. Methodology

Fig. 1 shows the method used in this study, which comprises two stages: 1) data processing and 2) classification model. The first stage bases on the knowledge discovery process (KDD), starting from describing the data set, processing, and model construction, described in points A and B of this section. In the second stage, ranking methods are applied to find the descriptive attributes. The decision trees algorithm and neural networks are applied to obtain classification models. Finally, the classification model with the best performance is obtained; this process is described in the results section.

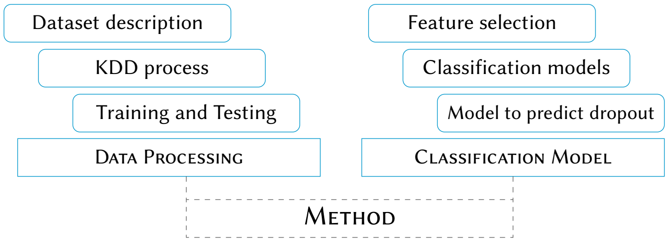


Fig. 1. The method applied to find the best predictive model of dropout.

A. Dataset Description

Table III presents the distribution of data used in this study, obtained from three deaneries distributed among 25 programs collected from fall 2010 to fall 2019 (Table III).

TABLE III. Dataset Description

Deanery	#Programs	#Records	%programs
Engineering	10	19,174	37.23
Social Science	6	11,092	21.45
Administrative Science	9	21,231	41.23
Total	25	51,497	100

TABLE IV. Attribute Description

#	Attribute	Description
1	PERIODO	Spring, summer and autumn
2	PROM_PANTE	Grade point average of the previous period
3	PROM_INI	Grade point average at the beginning of the period
4	EDAD	The age with a range between 18 to 24 years old
5	FALTAS_PANTE	Absences last period
6	ASIST_PANTE	Attendance until your last period
7	GENDER	M=Male, F= Female
8	SEMESTRE_PANTE	Semester to the period before
9	SUPPORT	Indicates if you receive any type of financial support such as discounts or agreements
10	REPRO_PANTE	Courses failed up to their last period
11	REPRO_1X	Courses failed only once
12	REPRO_2X	Courses failed twice
13	REPRO_3X	Courses failed three or more times
14	PERIODOS_INSCRITOS	All periods in which you have registered
15	PROMEDIO_AC	Current grade point average
16	CRED_PANTE	Credits taken
17	AVANCE_XCRED	Progress according to approved credits
18	ASIG_INSCRITASP	Courses registered in the period
19	FALTASP	Absences during the period
20	ESTADOCIVIL	Marital status
21	NINGRESO	This indicates that he/she is a new student
22	FORANEO	Indicates if you come from out of state
23	REPROBADAS1P	The proportion of courses failed in the first partial period
24	REPROBADAS2P	The proportion of courses failed in the second partial
25	REPROBADAS3P	The proportion of courses failed in the third partial
26	DESERTOR	Defines the deserter class in a dichotomous way (Yes or no)

Each record contains 25 attributes plus the Deserter class described in Table IV. The Deserter class labeled the data, of which Deserter(S) is 6,282, and No Deserter(N) is 45,215, so there is an unBalanced class. This class type is a problem for predictive models since it can lead to erroneous results in the prediction. That is, the class with a more significant number of samples obtains better predictive performance than the class with few samples, which is often the one of most significant interest [36]-[38].

Then, as mentioned above, the Deserter(S) class represents 12.19% of the total samples, thus exposing a class balance problem. To solve this problem, we will use the oversample and undersample techniques, because as mentioned by [30], they are easy to implement and obtain excellent results. The proposal uses an oversample for the Deserter(S) class and an undersample for the Deserter(N) class. In this way, a new set of data will be obtained with a balanced class, moving on to the training phase. The description of the attributes is shown in Table IV.

Additionally, a data set of 2,244 records concerning spring 2020 will be used without labeling to predict possible desertion in the final model.

B. KDD Process

According to [8][39][40], knowledge discovery in KDD (Knowledge Discovery in Databases) was the first model for knowledge extraction methodologically and works as a tool for decision making. In this way, the KDD process applied as follows:

The selection of attributes utilizing algorithms allows improving the input data's quality with the elimination of attributes that are not relevant [41]. In this study, we use subsection selection with CFS (Correlation Feature Selection), the filter method (Chi-square and gain information), and wrapping (Random Forest).

Pre-processing and transformation: activities such as missing data processing, noise reduction, among others, are performed. They start with data extraction from the institutional data warehouse and other sources such as Salesforce and Excel files. This information is already clean and automated by a data integration system, which employing process maps, performs the queries, validations, and transformations in an intermediate environment between the extraction and the loading of the data. All data is finally in dimension and fact tables, following the literature's standards [38]. Relational databases such as Oracle 11g, MySQL, and MSQL Server are used.

Data mining (classification model): The methods: X^2, Relief, and SOAP (Selection of Attributes by Projections) are applied since they increase accuracy, decrease overtraining since they eliminate data with better significance, and increase training speed [28]. The C5 decision tree algorithm and the artificial neural network algorithm with multilayer perceptron with 1 and 2 layer topologies with different numbers of neurons also obtain the best classification model obtained between both. For the evaluation of each algorithm's performance, the metrics of precision, accuracy, specificity, and sensitivity were used, as well as the balanced accuracy and the Matthew's Correlation Coefficient (MCC). For the validation of the models, cross-validation with base ten is employed. In the training part, the balanced subset of data applies. The proportion for the Deserter(S) class is 34.5%.

C. Training and Testing

Executions perform with different proportions in several instances; the following strategies are applied: 1) training with 70%, 80%, and 90% of the available data and 2) For the initial evaluations of the models, base ten cross-validations use.

D. Feature Selection

Two methods for feature selection are used to find the most relevant attributes. To determine the order of importance of the attributes rank methods (chi_square and gain information) were used and filter methods were used to determine a subset with the most significant attributes.

E. Classification Model

Two methods are used to build classifier models to predict attrition: 1) decision trees using the Random Forest [14],[17] method and 2) artificial neural networks [12]-[16]. Both methods are described in more detail in the results section.

F. Model to Predict Dropout

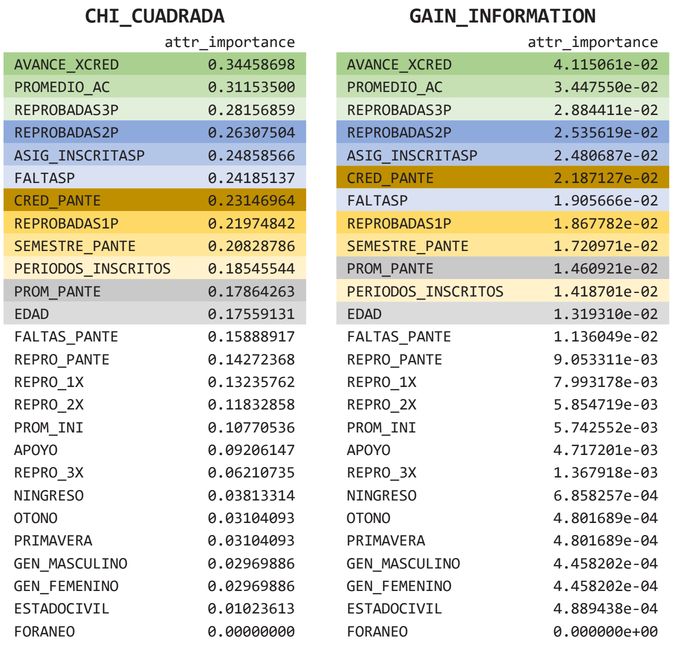
Finally, the model with the best metrics (accuracy, sensitivity, precision, among others) is chosen after performing the different experiments with test and training data. The process is described in detail in the results section.

V. Results

In this section, the results are presented in the following order: First, the descriptive attributes of the dropout phenomenon are listed; second, the predictive model obtained using decision trees is shown; third, the predictive model applying neuronal networks is presented; and finally, the model with the best performance and the prediction obtained for the spring 2020 period is described.

A. Features Selection

Two feature selection methods are applied to identify the most relevant attributes. The first method, feature selection algorithms, was used with rank methods (chi_square and gain information) to determine the most relevant attributes. In which it is observed that similar results were obtained with both methods (Fig. 2). A second method was performed with filter methods to determine a subset with the most significant attributes—also, the wrapping algorithm applies with the random forest (Table V).



CHI_CUADRADA

	attr_importance
AVANCE_XCRED	0.34458698
PROMEDIO_AC	0.31153500
REPROBADAS3P	0.28156859
REPROBADAS2P	0.26307504
ASIG_INSCRITASP	0.24858566
FALTASP	0.24185137
CRED_PANTE	0.23146964
REPROBADAS1P	0.21974842
SEMESTRE_PANTE	0.20828786
PERIODOS_INSCRITOS	0.18545544
PROM_PANTE	0.17864263
EDAD	0.17559131
FALTAS_PANTE	0.15888917
REPRO_PANTE	0.14272368
REPRO_1X	0.13235762
REPRO_2X	0.11832858
PROM_INI	0.10770536
APOYO	0.09206147
REPRO_3X	0.06210735
NINGRESO	0.03813314
OTONO	0.03104093
PRIMAVERA	0.03104093
GEN_MASCULINO	0.02969886
GEN_FEMENINO	0.02969886
ESTADOCIVIL	0.01023613
FORANEO	0.00000000

GAIN_INFORMATION

	attr_importance
AVANCE_XCRED	4.115061e-02
PROMEDIO_AC	3.447550e-02
REPROBADAS3P	2.884411e-02
REPROBADAS2P	2.535619e-02
ASIG_INSCRITASP	2.480687e-02
CRED_PANTE	2.187127e-02
FALTASP	1.905666e-02
REPROBADAS1P	1.867782e-02
SEMESTRE_PANTE	1.720971e-02
PROM_PANTE	1.460921e-02
PERIODOS_INSCRITOS	1.418701e-02
EDAD	1.319310e-02
FALTAS_PANTE	1.136049e-02
REPRO_PANTE	9.053311e-03
REPRO_1X	7.993178e-03
REPRO_2X	5.854719e-03
PROM_INI	5.742552e-03
APOYO	4.717201e-03
REPRO_3X	1.367918e-03
NINGRESO	6.858257e-04
OTONO	4.801689e-04
PRIMAVERA	4.801689e-04
GEN_MASCULINO	4.458202e-04
GEN_FEMENINO	4.458202e-04
ESTADOCIVIL	4.889438e-04
FORANEO	0.000000e+00

Fig. 2. Rank of feature selection.

Fig. 3 shows the order of attributes from least to most important with box diagrams, thus showing the dispersion of data for each. It can be noted that the variable FORANEO is not significant, and the variable REPROBADASP is significant. That means that the most significant variables are related to the failure rate, absences, semesters completed, age, average income, current average, and progress according to credits. This analysis also allowed us to detect some inconsistencies in the attributes PROM_PANTE, AVANCE_XCRED, and FALTASP:

TABLE V. The Most Relevant Attributes

#	Attributes	Wrapper-Random Forest	Consistency Based	CFS
1	ASIG_INSCRITASP	x	x	x
2	FALTASP	x	x	x
3	AVANCE_XCRED	x	x	x
4	PROMEDIO_AC	x	x	x
5	PROM_PANTE	x	x	
6	REPROBADAS3P	x	x	x
7	CRED_PANTE	x	x	
8	PROM_INI	x	x	
9	EDAD	x	x	
10	REPROBADAS2P	x	x	x
11	FALTAS_PANTE	x	x	
12	SEMESTRE_PANTE	x	x	x
13	PERIODOS_INSCRITOS	x	x	
14	REPROBADAS1P	x	x	x
15	APOYO	x	x	
16	REPRO_PANTE	x		
17	REPRO_1X	x	x	
18	OTONO	x	x	
19	PRIMAVERA	x		
20	REPRO_2X	x	x	x
21	GEN_FEMENINO	x		
22	GEN_MASCULINO	x	x	
23	NINGRESO	x	x	
24	REPRO_3X	x		
25	ESTADOCIVIL	x	x	
26	FORANEO			
	Total Attributes	25	21	9

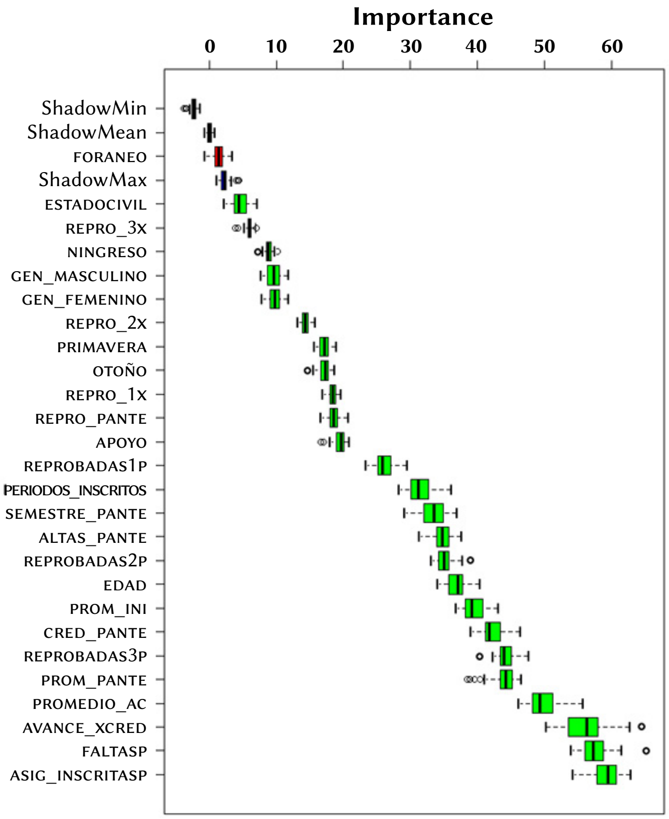


Fig. 3. Analysis with the wrapper method + random forest for the selection of attributes.

- PROM_PANTE. Most of them represent zeros values, which is not necessarily incorrect since these variables are affected by new entries and re-entries. When they are new entries, it is expected that they do not have a previous average because there is no history. The same happens with their advance by credits and faults. However, atypical data is found, where it is not typical for values greater than 0 and less than 6 to exist, since it indicates that someone who failed the previous semester and registered for the next period without credit. Thus, 90 records are discarded due to the inconsistency described.
- FALTASP. Five cases were eliminated from the data set with absences greater than 500, which could indicate an inadequate capture in the system.
- ADVANCE_XCRED. 2 cases with incorrectly calculated data from the system were discarded.

The next phase was to determine which attributes are applied for model training. For this task, decision trees with base-10 cross-validation is used as a classification algorithm. Fig. 4 shows the results with three training ratios, 70%, 80%, and 90%, and with attributes according to the methods of Correlation (CFS), Consistency, and Wrap (random-forest). It can see that the best performances were obtained with a training ratio of 90% and 80% with selection methods by Consistency and Wrap-RF. The selection of attributes by CFS causes the model to lose sensitivity with any training ratio, being the least desired option. However, it highlights that it reduces up to 9 significant attributes, which could be useful when training time must be optimized at the expense of loss of fit.

From these results (Fig. 4), it determines that the most important attributes are those obtained by the wrapper (Random Forest) due to the performance obtained. These are ASIG_INSCRITOSP, FALTASP, AVANCE_XCRED, PROMEDIO_AC, PROM_PANTE, REPROBADAS3P, CRED_PANTE, PROM_INI, AGE, REPROBADAS2P, FALTAS_PANTE, SEMESTRE_PANTE, REGISTERED_PERIODS, REPRODUCED1P, SUPPORT, PANT_PLAY, REPRODUCED1X, AUTUMN, SPRING, REPRODUCED2X, FEMALE_GEN, MALE_GEN, NINGROS, REPRODUCED3X, and STADIUM.

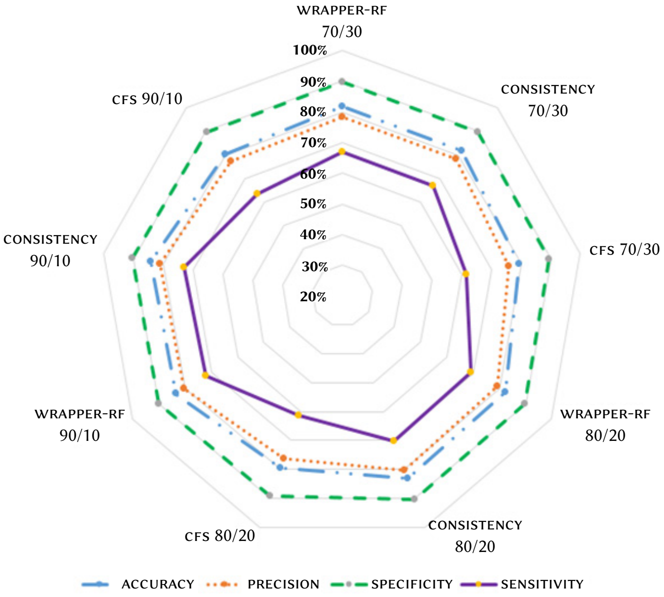
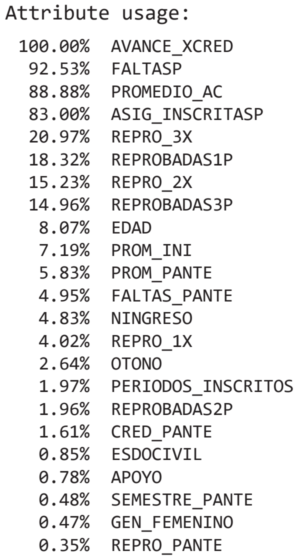


Fig. 4. Feature selection methods and their performance with decision trees.

B. Classifier Model With Random Forest Decision Trees

The decision tree algorithm C5.0 applies balanced data, and the attributes of Table IV. Fig. 4 confirms that the attributes selected by RF-Wrap give the best results, although they are very similar to those obtained by Consistency. The lowest evaluation is obtained using only the attributes by CFS. It can be seen that only the accuracy metric is lower (~88%). This metric indicates the proportion of true positives of those classified as dropouts.

The model's evaluation improvement, the balanced accuracy, and Matthew's correlation coefficient were obtained. The balanced accuracy is maintained for the attributes obtained by Consistency and RF envelope; Matthew's correlation coefficient indicates a robust positive relationship above 70%. In this way, the list of the most relevant attributes shown in Fig. 5 is obtained.



```
Attribute usage:

 100.00%  AVANCE_XCRED
  92.53%  FALTASP
  88.88%  PROMEDIO_AC
  83.00%  ASIG_INSCRITASP
  20.97%  REPRO_3X
  18.32%  REPROBADAS1P
  15.23%  REPRO_2X
  14.96%  REPROBADAS3P
   8.07%  EDAD
   7.19%  PROM_INI
   5.83%  PROM_PANTE
   4.95%  FALTAS_PANTE
   4.83%  NINGRESO
   4.02%  REPRO_1X
   2.64%  OTONO
   1.97%  PERIODOS_INSCRITOS
   1.96%  REPROBADAS2P
   1.61%  CRED_PANTE
   0.85%  ESDOCIVIL
   0.78%  APOYO
   0.48%  SEMESTRE_PANTE
   0.47%  GEN_FEMENINO
   0.35%  REPRO_PANTE
```

Fig. 5. Percentage of use of each attribute in the generation of the tree.

C. Classifier Model With Artificial Neural Network

The balanced data set obtained from the decision trees with the same attributes shown in Table V is used to train the neural network.

Two configurations of hidden layers apply, one of (2,2) and another of (12) neurons, to contrast results. The notation (2,2) refers to the number of hidden layers and neurons in the neural network architecture; the comma separates the hidden layers, the digit indicates the number of neurons per layer, so the configuration (2,2) means two hidden layers with two neurons each. At the same time, the notation (12) is one hidden layer with 12 neurons. Base 10 cross-validation was applied to evaluate the neural networks with their configurations (Fig. 6).

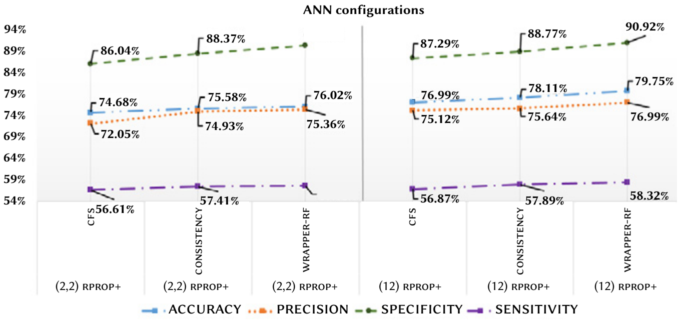


Fig. 6. Configurations for artificial neural networks with evaluation for accuracy, precision, specificity, and sensitivity.

Fig. 6 shows that the attributes obtained by CFS have the lowest values. However, as attributes are added, there is an improvement in the results. Although accuracy is high, sensitivity is low, meaning that it can detect about 57% of defectors (depending on network configuration and several attributes). On the other hand, it is analyzed the result with the Matthews correlation coefficient (MCC) and balanced accuracy.

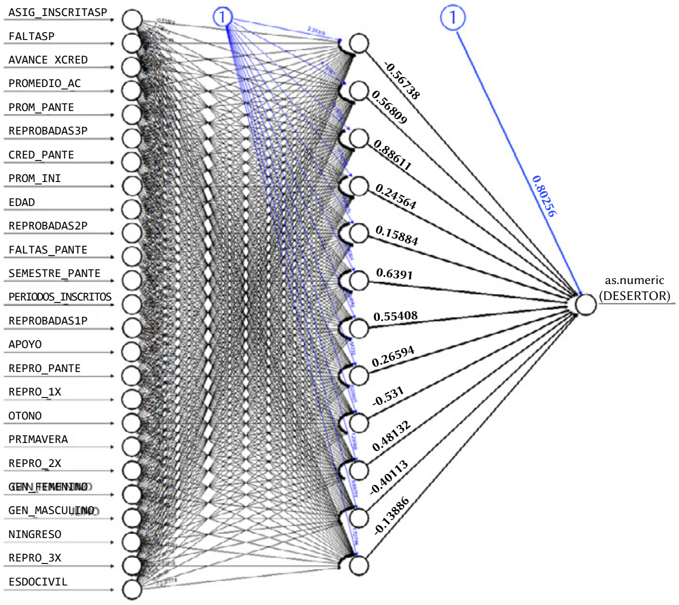


Fig. 7. A trained neural network with 12 neurons in a hidden layer.

This way, it is observed that the MMC is between 49% and 54%, while the balanced accuracy is between 71% and 75%. The Matthew coefficient indicates that the evaluations are between 49.66% and 53.69%, translating into a strong positive relationship. Besides, it can be seen that in both metrics, the minimum values are per CFS and the maximum values per Wrapper-RF. The selected model can be seen in Fig. 7 that shows only the neural network configuration with one hidden layer and 12 neurons (which resulted in the best neural network performance).

D. The Best Classifier Model

According to the classifier and configuration used, Fig. 8 shows the best results of artificial neural networks and decision trees. For the case of decision trees, the best results are given by attributes taken by RF-Wrap. For Neural Networks, the most appropriate configuration was a hidden layer with 12 neurons. The results show that decision trees are the best classification algorithm for detecting deserters.

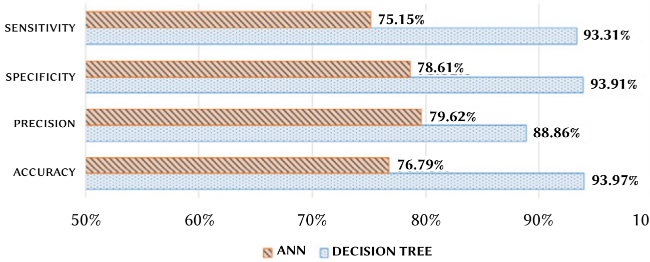


Fig. 8. Comparison between neural networks and decision trees.

Overall, when comparing the balanced accuracy and Matthew's correlation coefficient for the performance evaluation of both classification models. Similarly, Fig. 8 indicates that decision trees perform better than neural networks since decision trees achieve 94.34% balanced accuracy and a Matthew correlation coefficient of 87.43%, higher than those obtained by ANNs, 74.33%, 53.40%, respectively.

Since the objective is to detect possible defectors, an analysis was made by a period with defections from three previous periods (Fall 2018, Spring 2019, and Fall 2019) plus the current period (Spring 2020). According to the algorithm, two models were obtained per classifier and a balanced class (3x.80%) and data from autumn 2010 to spring 2018. Fig. 9 shows the actual dropouts from fall 2018 to fall 2019 and projects in spring 2020.

The prediction made by decision trees is closer to the real values from spring 2019, but it retains a uniformity in predictions between periods, while artificial neural networks are more separated from the real values than decision trees from spring 2019. This behavior is represented by the dotted line in Fig. 9. On the other hand, in the case of neural networks, training results show that, although it has better performance in the validations, Mathew's correlation coefficient is above 50%, which indicates a strong positive correlation.

Fig. 9 shows the actual results obtained at the end of the spring 2020 period. The prediction was a dropout of 436 students, and actually, 373 students dropped out in the three deaneries analyzed. The results in the Figure show the work done by the student monitoring department, which due to the Covid-19 pandemic, implemented provisional actions such as easy payment, more scholarships, agreements, personalized counseling, constant personalized monitoring, among other actions that allowed reducing the number of students predicted as possible dropouts. In this way, having a predictive model has helped to take preventive measures to reduce the dropout rate.

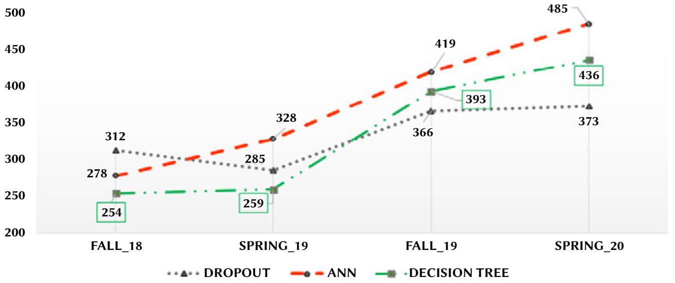


Fig. 9. Forecast of defectors and actual data obtained in the period spring 2020.

Thus, the Table VI presents a comparison of the application of neural networks. There is a difference between the number of records and variables studied in each study, determining factors in obtaining performance metrics such as accuracy and sensitivity. Although in [15], [16] the accuracy was higher than 96%, while in our study, it was 76.79%, this does not mean that neural networks are not adequate to predict dropout university in our case. Nevertheless, they are a function of the type of data, variables used, classification methods used in the neural networks, and the attribute selection methods applied before the classification, as mentioned in the explanatory features section.

TABLE VI. Comparison of Results Obtained Using Neural Networks

Reference	Dataset	# attributes	Accuracy	Sensitivity
7	43,617	Unspecified	88.30%	92.30%
32	Cohort 2010	19	82.00%	71%
35	61,340	18	87.30%	66.00%
36	2,670	11	96.80%	Unspecified
37	456	25	96.71%	Unspecified
own	51,497	25	76.79%	53.40%

The Table VII shows a comparison of the results obtained in our study with those obtained in the literature. It can be observed that the difference between accuracy and sensitivity is notorious and the number of variables and data analyzed. As in neural networks, these results' difference lies in the data set analyzed and the attributes. In summary, the closer the percentage of metrics is to 100%, the better the classifier.

TABLE VII. Comparison of Results Obtained Using Random Forest

Reference	Dataset	# attributes	Accuracy	Sensitivity
33	32,538	784	62.24%	69.40%
35	61,340	18	90.70%	68.40%
own	51,497	25	94.34%	87.43%

VI. Conclusions

In the first analysis of attributes, some that do not seem so relevant can be discarded, such as failed subjects, period (autumn or spring), and gender. However, comprehensive analysis with attribute selection by envelope - random forest shows that the models' maximum performance is obtained using most of the attributes. On the other hand, the class balance allowed us to improve the performance metrics of both algorithms. Mathew's correlation coefficient and balanced accuracy provided a better evaluation of the models, allowing the results to be unaffected by variations in accuracy, precision, specificity, and sensitivity metrics.

Decision trees obtained the best Matthew correlation coefficient of 87.43% and balanced accuracy of 94.34%. On the other hand, the tests performed indicate that increasing the number of neurons in the hidden layer of ANN could improve performance. However, it requires more processing power since server training can be more than an hour-long, and it is the main reason for not having performed more tests.

Since the best model has been the one obtained by decision trees, it has been implemented in the Enterprise Resource Planning (ERP) institutional system, which helps the student to receive the accompaniment he requires so as not to interrupt his studies; this effort translates into monitoring about 450 students per period belonging to the deaneries of social sciences, administration, and engineering. In this way, the student monitoring area improves its administrative tasks using the model because it obtains a list in a matter of seconds and avoids consolidating a report of several Excel documents, even from different areas (admissions and school control).

Dropout is a challenge for any educational institution, and with the help of algorithms and technological platforms, it is possible to collaborate in decision making to avoid dropouts or abandonment, manage better tutoring and student support. It allows better management of academic and economic resources of the institution at the institutional level, optimizing its processes and reducing response times.

To continue with the research, we have considered using additional attributes such as payment history, debts, campus access, and similar, as well as implementing stratified sampling by deanery for class balancing in the model's training and dividing data into new entries and re-entry. Also, we intend to use other classification algorithms such as near neighbors, vector support machines, logistic regression, and use a combination of classifiers to generate a more robust solution. Also, we plan to use cloud services such as Machine Learning in AWS, Azure Machine Learning, BigML, IBM Watson, TensorFlow, or some other computer learning solution to improve processing times and, finally, deploy models to classify online mode.

References

[1] V. Tinto. "Definir la deserción: una cuestión de perspectiva". Revista de Educación Superior, vol. 71, no. 18, pp. 1-9, 1989. Available: https://bit.ly/3mptENw.

[2] Organization for Economic Cooperation and Development. "Panorama de la Educación, indicadores de la OECD 2019", Ministerio de Educación y Formación Profesional, 2019. Available: https://bit.ly/3afjDjq.

[3] C. Alonso, E. Blanco, T. Fernández, and P. Solís. "Caminos desiguales: trayectorias educativas y laborales de los jóvenes en la ciudad de México". INEE-El Colegio de México. Primera edición, 2014. Available: https://bit.ly/3oVxVtt.

[4] Organization for Economic Cooperation and Development. "Higher Education in Mexico: Labour Market Relevance and Outcomes", OECD Publishing, Paris, 2019, https://doi.org/10.1787/9789264309432-en.

[5] L. López Villafaña, and A. Beltrán Solache. "La deserción en estudiantes de educación superior: tres percepciones en estudio, alumnos, docentes y padres de familia". Pistas Educativas, no. 126, 2017. Available: https://bit.ly/37qBvWJ.

[6] E. Ramírez, D. Espinosa, and E. Millán. "Estrategia para afrontar la deserción universitaria desde las tecnologías de la información y las comunicaciones". Revista Científica, vol. 24, pp. 52-62, 2016, doi: 10.14483/udistrital.jour.RC.2016.24.a5.

[7] J. R. Casanova, A. Cervero, J. C. Núñez, L. S. Almeida, and A. Bernardo, "Factors that determine the persistence and dropout of university students", Psicothema, vol. 30, no. 4, 2018, pp. 408-414, doi: 10.7334/psicothema2018.155.

[8] Secretaría de Seguridad Pública. "Deserción escolar y conductas de riesgo en adolescentes". Subsecretaría de prevención y participación ciudadana, Dirección general de prevención del delito y participación ciudadana. gobierno federal SSP. Dirección General de Prevención del Delito y Participación Ciudadana. Gobierno Federal SSP, 2011. Available: https://bit.ly/34kOwz1.

[9] Secretaría de Educación Pública. "Principales cifras del sistema educativo nacional 2015-2016, cifras preliminares". Biblioteca de publicaciones oficiales del gobierno de México, 2016. Available: https://bit.ly/3gS8juK.

[10] I. W. Li, D. R. Carroll, "Factors influencing university student satisfaction, dropout, and academic performance: an Australian higher education equity perspective," National Centre for Student Equity in Higher Education, pp. 3-59, 2017. https://doi.o10.1080/1360080x.2019.1649993.

[11] P. Perchinunno, M. Bilancia, D. Vitale, "A Statistical Analysis of Factors Affecting Higher Education Dropouts", Social Indicators Research, 2019, https://doi.org/10.1007/s11205-019-02249-y.

[12] E. Chinkes, "Pronósticos y data mining para la toma de decisiones, pronóstico sobre la deserción de alumnos de una facultad". Cuadernos del CIMBAGE, no. 20, pp. 107-132, 2017. Available: https://bit.ly/2MEyGK0.

[13] M. Solis, T. Moreira, R. Gonzalez, T. Fernandez and M. Hernandez, "Perspectives to Predict Dropout in University Students with Machine Learning," 2018 IEEE International Work Conference on Bioinspired Intelligence (IWOBI), San Carlos, Costa Rica, 2018, pp. 1-6, doi: 10.1109/IWOBI.2018.8464191.

[14] N. Mduma, K. Kalegele, D. Machuve, "Machine learning approach for reducing students dropout rates. International journal of advanced computer research (IJACR)," Vol 9. No. 42, 2019. https://doi.org/10.19101/ijacr.2018.839045.

[15] M. Alban, D. Mauricio, "Neural networks to predict dropout at the universities. International journal of machine learning and computing," Vol. 9. No 2. Pp. 149-153, 2019, doi: 10.18178/ijmlc.2019.9.2.779.

[16] N. Lázaro, Z. Callejas, D. Griol, "Predicting computer engineering students' dropout in Cuban higher education with pre-enrolment and early performance data," Journal of technology and science education. vol. 10 no. 2, 2020, doi:10.3926/jotse.922.

[17] L. Auluck, N. Velagapudi, J. Blumenstock, J. West, "Predicting Student Dropout in Higher Education," 2016 ICML Workshop on #Data4Good: Machine Learning in Social Good Applications, New York, NY, USA, 2017, pp. 16-20.

[18] C. Lacave, A. Molina, J. Cruz-Lemus, "Learning Analytics to identify dropout factors of Computer Science studies through Bayesian networks", Behaviour & Information Technology, vol. 37, 2018, https://doi.org/10.1080/0144929X.2018.1485053.

[19] P. Ramírez, and E. Grandón, "Predicción de la deserción académica en una universidad pública chilena a través de la clasificación basada en árboles de decisión con parámetros optimizados," Formación Universitaria, vol. 11, no. 3, pp. 3-10, 2018

[20] A. B. Urbina-Nájera, J. C. Camino-Hampshire, R. Cruz-Barbosa, "Deserción escolar universitaria: Patrones para prevenirla usando

minería de datos educativa," RELIEVE, vol. 26, no. 1, 2020, http://doi.org/10.7203/relieve.26.1.16061.

[21] D. Sun, Y. Mao, J. Du, P. Xu, Q. Zheng, H. Sun, "Deep learning for dropout prediction in MOOCs," 2019 Eighth International Conference on Educational Innovation through Technology (EITT). Biloxi, MS. USA, 2019, doi: 10.1109/EITT.2019.00025.

[22] J. M. Moine, "Metodologías para el descubrimiento de conocimiento en bases de datos: un estudio comparativo," Universidad Nacional de la Plata. Facultad de Informática. Tesis presentada en la Facultad de Informática de la Universidad Nacional de la Plata. 2013. Available: https://bit.ly/3wbmfH0.

[23] C. Carvajal, J. González, S. Sarzoza, "Variables sociodemográficas y académicas explicativas de la deserción de estudiantes en la facultad de ciencias naturales de la universidad de playa ancha (chile)," Formación Universitaria, vol. 11, no. 2, 2017, pp. 3-12. https://doi.org/10.4067/s0718-50062018000200003.

[24] V. Cendejas, L. Acuña, M. Cortez, J. Bolaños, "El uso de modelo y metodologías de minería de datos para la inteligencia de negocios," Revista de sistemas computacionales y TIC'S. vol.3, no. 8, 2017, pp. 54-63. Available: https://bit.ly/2ThbtQV.

[25] M. Zavala, M. Álvarez, M. Vázquez, M., I. González and A. Bazán, "Factores internos, externos y bilaterales asociados con la deserción en estudiantes universitarios," Interacciones Revista de Avances en Psicología, vol. 4. no.1, 2018, pp. 59-69. https://doi.org/10.24016/2018.v4n1.103.

[26] S. Muñoz-Camacho, T. Gallardo, M. Muñoz-Bravo, C. Muñoz-Bravo, "Probabilidad de deserción estudiantil en cursos de matemáticas básicas en programas profesionales de la Universidad de los Andes Venezuela," Formación Universitaria, vol. 11, no. 4, 2018, pp. 33-42. https://doi.org/10.4067/s0718-50062018000400033.

[27] J. Gallegos, N. Campos, K. Canales, K., E. González, "Factores determinantes en la deserción universitaria. caso facultad de ciencias económicas y administrativas de la universidad católica de la santísima concepción," Formación Universitaria, vol. 11, no. 3, 2018, pp. 11-18. https://doi.org/10.4067/s0718-50062018000300011.

[28] R. Ruiz., J. Aguilar, and J. Riquelme, "Evaluación de Rankings de Atributos para Clasificación," Departamento de Lenguajes y Sistemas Informáticos. Universidad de Sevilla, Sevilla, España, 2002. Available: https://bit.ly/38bscde.

[29] C. L. Villagrá-Arnedo, F. J. Gallego-Durán, F. Llorens-Largo, R. Satorre-Cuerda, P. Compañ-Rosique, and R. Molina-Carmona, "Time-Dependent Performance Prediction System for Early Insight in Learning Trends," International Journal of Interactive Multimedia and Artificial Intelligence. vol. 6, no. 2, 2020. Doi: 10.9781/ijimai.2020.05.006.

[30] I. Witten, E. Frank, M. Hall, and C. Pal, "Data mining: Practical machine learning tools and techniques," Morgan Kaufmann Publishers, Burlington.

[31] C. Romero, and S. Ventura, "Educational data mining: A survey from 1995 to 2005," Expert Systems with Applications, vol. 33, no. 1, pp. 135-146, 2007. https://doi.org/10.1016/j.eswa.2006.04.005.

[32] C. Menes, G. Arcos, P. Moreno, and C. Gallegos C. "Desempeño de algoritmos de minería en indicadores académicos: Árbol de decisión y Regresión Logística," Revista Cubana de Ciencias Informáticas, vol. 9, no. 4, 2015.

[33] T. Mitchell, "Decision Tree Learning," Washington State University, 2000. Available: https://bit.ly/2N1AI32.

[34] N. Sánchez, "Máquinas de soporte vectorial y redes neuronales artificiales en la predicción del movimiento USD/COP spot intradiario," Universidad Externado de Colombia, vol. 9, pp. 113-172, 2016, doi: http://dx.doi.org/10.18601/17941113.n9.04.

[35] D. Chicco, and G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation," BMC Genomics, vol. 21, no. 6, pp. 1-13, 2020. https://doi.org/10.1186/s12864-019-6413-7.

[36] R. Malhotra, and S. Kamal, "An empirical study to investigate oversampling methods for improving software defect prediction using imbalanced data," Neurocomputing, vol.343, no. 28, 2019, pp. 120-140. https://doi.org/10.1016/j.neucom.2018.04.090.

[37] X. Zhang; Z. Shi; X. Liu; X. Li, "A Hybrid Feature Selection Algorithm for Classification Unbalanced Data Processing," IEEE International Conference on Smart Internet of Things (SmartIoT), 17-19 August 2018, doi: 10.1109/SmartIoT.2018.00055.

[38] I. Bolodurina, A. Shukhman, D. Parfenov, A. Zhigalov, and L. Zabrodina, "Investigation of the problem of classifying unbalanced datasets in identifying distributed denial of service attacks," Journal of Physics: Conference Series, vol. 1679, 2020, doi:10.1088/1742-6596/1679/4/042020.

[39] M. Rogalewicz, and R. Sika, "Methodologies of knowledge discovery from data and data mining methods in mechanical engineering," Management and Production Engineering Review, vol. 7, no. 4, pp. 97-108, 2016. https://doi.org/10.1515/mper-2016-0040.

[40] C.R.M. Rosa, M.T.A. Steiner, and P.J. Steiner Neto, "Knowledge Discovery in Data Bases: a Case Study in a Private Institution of Higher Education," IEEE Latin America Transactions, vol. 16, no. 7, pp. 2027-2032, 2018, doi: 10.1109/TLA.2018.8447372.

[41] J. S. Aguilar-Ruiz, and J. Díaz-Díaz, "Selección de atributos relevantes basada en bootstrapping," Departamento de lenguajes y sistemas informáticos. Universidad de Sevilla. ETS ingeniería informática. Actas del III Taller Nacional de Minería de Datos y Aprendizaje, TAMIDA2005, 2005, pp. 21-30.

[42] J. Rodríguez, J. Hernández, J., "La deserción escolar universitaria en México, la experiencia de la universidad autónoma metropolitana campus Iztapalapa," Revista Actualidades Investigativas en Educación, vol. 8, no. 1, 2011, pp. 1-31, doi: 10.15517/AIE.V8I1.9308.

Argelia Berenice Urbina Nájera

She is a full-time research professor at the Popular Autonomous University of the State of Puebla (UPAEP), México. She obtained a PhD from the UPAEP in 2015. She is a member of the National System of Researchers (SNI) of México since 2017. Her research focuses on Educational Data Mining, Learning Analytics, Machine Learning applied to health, business and education; as well as, other knowledge areas related to Education and Business Intelligence.

Luis Andrés Méndez Ortega

Master in Data Science and Business Intelligence from the Popular Autonomous University of the State of Puebla (UPAEP). Degree in Computer Engineering from the Autonomous University of Tlaxcala (UAT). With more than 9 years in the development and implementation of administrative software for different sectors such as automotive, motor transport, medical, commerce and government projects. He has also participated as an analyst in the area of systems for data migration, and user requirements management in the education sector. He is currently a business intelligence consultant in the CRM & BI area at UPAEP and responsible for BI architecture. Main interests in data management, ETL processes, computer learning and software development.

Improving Pipelining Tools for Pre-processing Data

María Novo-Lourés [1,2,3], Yeray Lage[1], Reyes Pavón[1,2,3], Rosalía Laza[1,2,3], David Ruano-Ordás[1,2,3], José Ramón Méndez[1,2,3]*

[1] Department of Computer Science, University of Vigo, ESEI - Escuela Superior de Ingeniería Informática, Edificio Politécnico, Campus Universitario As Lagoas s/n, 32004 Ourense (Spain)
[2] CINBIO, University of Vigo, Research Group SI4, Department of Computer Science, 32004 Ourense (Spain)
[3] SING Research Group, Galicia Sur Health Research Institute (IIS Galicia Sur), SERGAS-UVIGO (Spain)

Received 7 July 2020 | Accepted 22 July 2021 | Published 21 October 2021

unir
LA UNIVERSIDAD
EN INTERNET

Abstract

The last several years have seen the emergence of data mining and its transformation into a powerful tool that adds value to business and research. Data mining makes it possible to explore and find unseen connections between variables and facts observed in different domains, helping us to better understand reality. The programming methods and frameworks used to analyse data have evolved over time. Currently, the use of pipelining schemes is the most reliable way of analysing data and due to this, several important companies are currently offering this kind of services. Moreover, several frameworks compatible with different programming languages are available for the development of computational pipelines and many research studies have addressed the optimization of data processing speed. However, as this study shows, the presence of early error detection techniques and developer support mechanisms is very limited in these frameworks. In this context, this study introduces different improvements, such as the design of different types of constraints for the early detection of errors, the creation of functions to facilitate debugging of concrete tasks included in a pipeline, the invalidation of erroneous instances and/or the introduction of the burst-processing scheme. Adding these functionalities, we developed Big Data Pipelining for Java (BDP4J, https://github.com/sing-group/bdp4j), a fully functional new pipelining framework that shows the potential of these features.

Keywords

Burst Processing, Data Pre-processing, Java, Pipeline Frameworks.

DOI: 10.9781/ijimai.2021.10.004

I. Introduction and Motivation

Data mining techniques emerged as a set of tools for exploiting heterogeneous and often unstructured data compiled from a wide variety of information sources to improve decision support processes in different domains (healthcare, commercial decisions, stocks market predictions,...) [1]. Under this paradigm for addressing decision support, facts could not be explained with simple and isolated variables from the same domain, but as the combination of a large collection of circumstances (variables) that occur in different and heterogeneous domains [2]. Hence, stocks market predictions, for example, should be modelled by compiling information about the moods of the people (maybe from the news or from social networks), company results (profit and loss), customer satisfaction, the company image (for non-customers), etc. These intuitive and simple ideas became more and more popular and originated the current revolution of big-data.

Tools and programming methods to implement the compilation and pre-processing of data have evolved considerably over time. Currently the most advanced tools to address these issues are included in big-data frameworks, which facilitates the processing of enormous volumes of information over large computer clusters. Particularly, MapReduce [3], [4] is a powerful and earlier programming paradigm, mainly popularized by Google and Hadoop Project, which simplifies the processing of data using hundreds of cluster nodes. Several MapReduce implementations are available including Apache Hadoop [5], Amazon Elastic MapReduce [6], Disco MapReduce [7], [8] and Spark [9], among others. However, MapReduce does not provide a complete solution to easily address all the problems of pre-processing data [3], [4], [10]. Particularly, the most relevant limitation of MapReduce is the processing model (batch) which requires all data to be (up) loaded into the cluster to execute its analysis. Consequently: (*i*) this model is not suitable for processing real-time streaming sources (such as twitter streaming data); and (*ii*) the harnessing of the computational capabilities of the cluster is clearly hampered by the need for loading the data. These limitations led to the introduction of other forms of analysing data, such as the pipelining methods [11]. These methods are based on dividing the data analysis process into a set of small and easy to implement tasks; and orchestrating their sequential/parallel execution. After the emergence of pipelining methods, many pipelining frameworks were introduced [12] and several important companies started offering effective pipeline-based data analysis services such as AWS Data Pipeline [13], SnapLogic IIP [14] or Alooma Enterprise Data Pipeline [15]. Due to the popularity of pipeline technology, recent studies have addressed the issue of improving the speed of data processing [16]–[20]. However, there are still many areas for improvement of such frameworks, such as the facilities provided to

* Corresponding author.

E-mail address: moncho.mendez@uvigo.es

developers or the implementation of mechanisms capable of detecting errors at an early stage (constraints, datatype issues, etc.).

During the last years, Java has become popular in the development of enterprise software [21]. By getting in touch with IT –Information Technology– professionals and examining job offers, we found that important companies (e.g., Inditex, PSA, CaixaBank) primarily use Java technology in their developments. Hence the use of Java to define pipelines (*i*) enables the possibility of reusing many software components (especially persistence components to access their information); (*ii*) facilitates the search of qualified developers; and (*iii*) allows for the use of a wide variety of software libraries and services that are currently developed in Java (sometimes exclusively) [22], [23].

In this context, this work focuses specifically on the pipelining schemes used by enterprises and, therefore, primarily considers pipelining frameworks supporting the execution of Java tasks. This study identified some limitations of current software implementing pipeline functions such as: (*i*) the definition or checking of constraints; (*ii*) the invalidation inconsistent/error data; or (*iii*) the data sharing between tasks. Particularly, some mechanisms should be introduced to minimize the errors in the entire process (constraints). Additionally, when invalid data are found they should be marked to discard (instance invalidation) and should not be further processed. Finally, sharing data between tasks cannot be safely implemented through external repositories (databases, files, etc.) because some task execution questions should be previously addressed (i.e. how additional data instances -with regard to the initial ones- affect the process). To this end, we have developed BDP4J [24], a new pipelining framework that successfully address all limitations identified for analysed software. In addition to BDP4J software, this work also contributes different processing schemes for resuming or debugging the operation of pipelines, several constraints that should be checked in pipelines, and methods to implement the identified functionalities.

The rest of our work is structured as follows: Section II presents the state of the art in Java frameworks to define computational pipelines together with their limitations. Section III introduces BDP4J as a framework able to solve the limitations that have been identified in analysed software. Finally, section IV summarizes the main outcomes and future work.

II. State of Art

As stated before, several pipeline-based tools for data processing have been successfully introduced in recent years. In order to facilitate the implementation of pipeline-based data solutions in enterprises and reutilize previously developed software components, the use of language-independent, and/or Java pipelining frameworks would probably be the best solution. In this section, we analyse a set of the most adequate frameworks, demonstrating their strengths and weaknesses, in order to facilitate the definition of (big) data studies by enterprises. We analyse the functionalities of some frameworks that are no longer available (such as COMPSs [25], [26] whose URLs and GitHUB repositories have been removed) or currently obsolete (Conan2 [27], Dockerflow [28], [29], Suro [30] and Swift [31]–[33]). Conan2 can only be built using Java 6 which is obsolete, Dockerflow has been abandoned on 2017 (see readme.md in official GitHUB repository), Suro can only be compiled using obsolete Gradle/JDK versions and Swift requires Java endorsed dirs property which is obsolete (see https://docs.oracle.com/javase/8/docs/technotes/guides/standards/). Moreover, we also analyzed other available software including COMPI [34], Cromwell [35], Drake [36], Mallet [37], [38] and ML –Machine Learning– pipeline [9], [39], [40].

By studying the above-mentioned frameworks, we identified a set of relevant features that are not addressed by all of them. Table I summarizes the analysed features and displays a comparative analysis of their presence in the studied frameworks

As shown in Table I, the vast majority of analysed frameworks do not have any checking strategy while orchestrating tasks inside a pipeline (*i*, *ii*). Keeping this in mind, we estimate the utility of type constraints included in strong typing languages (such as Java) as an effective method to prevent development errors (*i*). Inspired by strong typing languages, we found type constraints appropriate to check whether the type of output information generated by a task is consistent with the input information required for the next task executed in sequence. The type of input information for tasks that are executed in parallel should be the same. We are strongly convinced of the need to add some constraints to establish the right execution order of tasks (*ii*) and prevent inconsistent pre-processing (for instance, stripping HTML tags is mandatory before tokenizing contents). Additionally, in our view, tasks can be classified into different categories (*iii*), thus contributing to help users in their selection (if a GUI –Graphical User Interface– is provided) and in the validation of their orchestration. The goal of some tasks in particular is to simply compute instance properties and not to transform the input data. Therefore, the classification of tasks into different categories would provide the possibility of defining constraints for validating tasks and their orchestration as well as to detect the execution errors.

Additionally, taking advantage of parallel processing schemes (*iv*) and/or distributed execution methods (*v*) could contribute to reduce the data processing time required.

The communication between tasks is often not limited to the output-input stream. As an example, in text mining, a dictionary is created after the tokenizing process, and can be used later to build dense vector representations. However, data sharing (*vi*) between tasks in burst-based operation implies taking additional issues into account. For example, in a large number of problems, the task that generates the shared data should complete the processing of all data instances belonging to the current burst before executing the task that consumes the shared data. Additionally, we should also select the fastest sharing mechanisms from those available. In this sense the use of language-internal mechanisms (for example, a singleton-based object) is preferred to that of external mechanisms (a database). However, if only external mechanisms can be used, they should be carefully selected to improve performance. In this sense, distributed memory object caching systems (such as Memcached or Ehcache) are clearly better than (slower) SQL-based database servers.

Loading task orchestration (*vii*) from a file (XML –eXtensible Markup Language– or YAML –YAML Ain't Markup Language–) allows for providing independence between individual task definitions and their orchestration. Additionally, these orchestration definition languages would allow customizing tasks with specific execution parameters and easily modifying the steps of the pipeline.

The dynamic loading of tasks (*viii*) from files included in a directory (for instance .jar files) enables the user to develop customized tasks and facilitates their development (because the tasks can be defined in separate projects).

Once a great amount of data has been compiled, it must be analysed by using ML strategies. Therefore, data mining developers would greatly appreciate the provision of facilities for the integration of ML frameworks (such as Weka) (*ix*).

The most relevant feature that could be provided from a pipeline framework is probably the ability to transparently resume the execution (*x*) of a particular pipeline when the application crashes. To implement this functionality, we should address the persistence of data instances after being processed by each task and check whether

TABLE I. Relevant Features for Pipeline Frameworks

	COMPSs	**COMPI**	**Conan2**	**Cromwell**	**Dockerflow (+Google Cloud Dataflow API)**	**Drake**	**Mallet**	**ML Pipeline (Apache Spark)**	**Suro**	**Swift**
(*i*) Type check				✓	✓			✓		
(*ii*) Dependency check	✓	✓		✓		✓				✓
(*iii*) Different kinds of tasks							✓			
(*iv*) Parallel processing	✓	✓			✓	✓		✓	✓	✓
(*v*) Distributed	✓				✓				✓	✓
(*vi*) Data sharing	✓		✓	✓	✓		✓ (partially)	✓		
(*vii*) Loading Pipeline from File		✓			✓					
(*viii*) Dynamic loading of tasks						✓			✓	
(*ix*) Integration with ML					Cloud Machine Learning		Own ML API	Spark ML		
(*x*) Resume execution from a particular task		✓		✓	✓	✓		✓	✓	✓
(*xi*) Orchestration GUI			✓		✓				✓	
(*xii*) Open Source	✓	✓	✓	✓	✓		✓	✓	✓	✓
(*xiii*) Language agnosticism	✓	✓			✓	✓				
(*xiv*) Resource manager	✓			✓						
(*xv*) Instance invalidation										
(*xvi*) Last instance notification										
(*xvii*) Developer mode										

the pipeline and data burst are the same when resuming the execution of the pipeline. The storage of instances should be completely transparent for developers and carried out as quickly as possible to reduce the time requirements for data analysis.

Finally, there are minor features that improve the benefit of using a data pipelining framework including, but not limited to, the existence of a GUI application to support the orchestration of tasks (*xi*), its availability as open source (*xii*), language agnosticism (i.e. the possibility of implementing tasks in any programming language) (xiii), or the inclusion of resource management utilities (*xiv*). These utilities (for resource management) could allow assessing the resources (RAM –Random Access Memory–) required for pre-processing data and/or their reservation in cloud environments (as COMPSs do), or simply limit the usage of CPU –Central Processing Unit– /RAM (as Cromwell).

As we can deduct from the current analysis, current software implementing pipelining strategies have important limitations that hamper their application to address the pre-processing of data. This fact suggests the need to develop a pipelining framework and implement all the features analysed in this study. Furthermore, we have also identified several interesting features that have not been raised by any of the analysed frameworks. Particularly, we found the advances in the following directions to be suitable: (*xv*) instance invalidation, (*xvi*) last instance notification, and (*xvii*) developer mode.

The instance invalidation (*xv)* is the capability of discarding a data instance when we detect its invalidity during the pipelining process (for instance when trying to download the contents of a tweet through its Twitter ID that has been previously removed). The invalidation of a data instance could be invoked in any task belonging to the pipeline and implies that it will not be further processed (no other task will be called with the instance). Despite the fact that this functionality seems to be very simple and intuitively required, most pipeline implementations do not incorporate it.

The main advantage of pipeline processing schemes is the processing of streaming data instances (for instance a Twitter stream). As shown in Fig. 1, before the beginning of the pipelining process (b), a set of available data instances are buffered, in the form of burst (a) to be processed when computational resources are available (after processing the last data-instances burst).

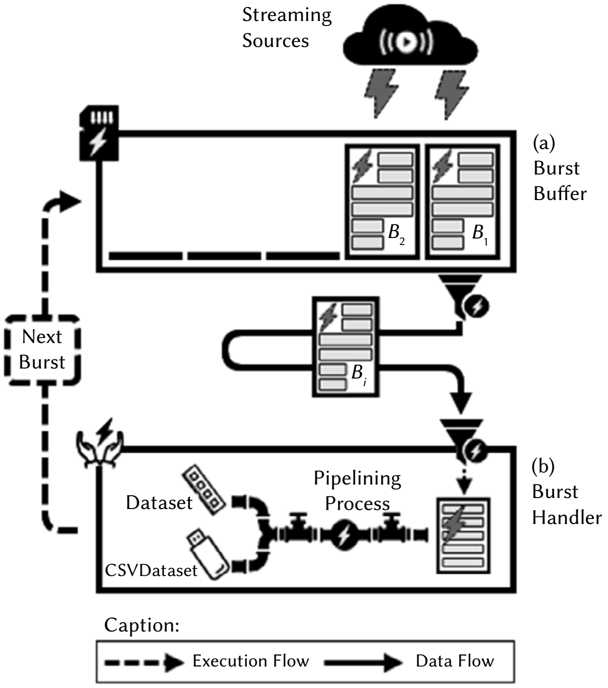


Fig. 1. General burst operation overview.

As result of the pipelining process, a dataset is generated (or grown) and stored in disk or main memory. In a task that saves the data to disk (or any other resource dependent task), the identification of the last data instance of a burst (*xvi*) would help to safely free resources (closing/flushing opened files, database connections, etc.). As a particular example, closing dataset files would make them available to be used for further analysis until the execution of the dataset saving task for the next burst.

As Fig. 1 shows, a framework would also provide facilities to store datasets (disk or memory) and let them grow both in number of rows and columns after processing each burst.

Finally, developers should be able to quickly test the proper operation of their tasks by using a collection of real data. However, to debug a certain task, all previous tasks defined in the orchestration should be executed before debugging the target task. In such a situation, providing mechanisms to achieve more efficiency when testing a new task adds a great value to pipelining frameworks (*xvii*). In particular, if the orchestration and input instances remain invariable for all test executions, the state of instances used as input for a task that is being debugged could be saved to disk (serialized) the first time they are computed, in order to reuse them for later executions.

Given the list of the most adequate features that a pipelining framework should provide, and starting from the simple pipelining support included in Mallet, we designed BDP4J, a new pipelining framework implementing most of the features that have been previously shown. Section III identifies the details of the implementation and use of the framework.

III. Introducing BDP4J

After analysing the pipelining frameworks, which provided a wide list of indispensable features for improving pipelining in enterprises, we developed BDP4J. BDP4J was inspired by the pipelining architecture included in Mallet software. Using this pipeline architecture as a base (and specifically the classes Instance, Pipe and SerialPipes), we transformed the name of the classes to accommodate them to the Java Code Conventions [41] and other non-explicit code rules (i.e. the name of interfaces and abstract classes that usually start with the prefix “Abstract” or “Default”). Specifically, Pipe, SerialPipes and Instance classes from the Mallet framework were transformed into AbstractPipe, SerialPipes and Instance BDP4J classes respectively. Using this architecture as a starting point, we implemented most features compiled in the previous section, achieving a product that is different from the Mallet pipeline implementation.

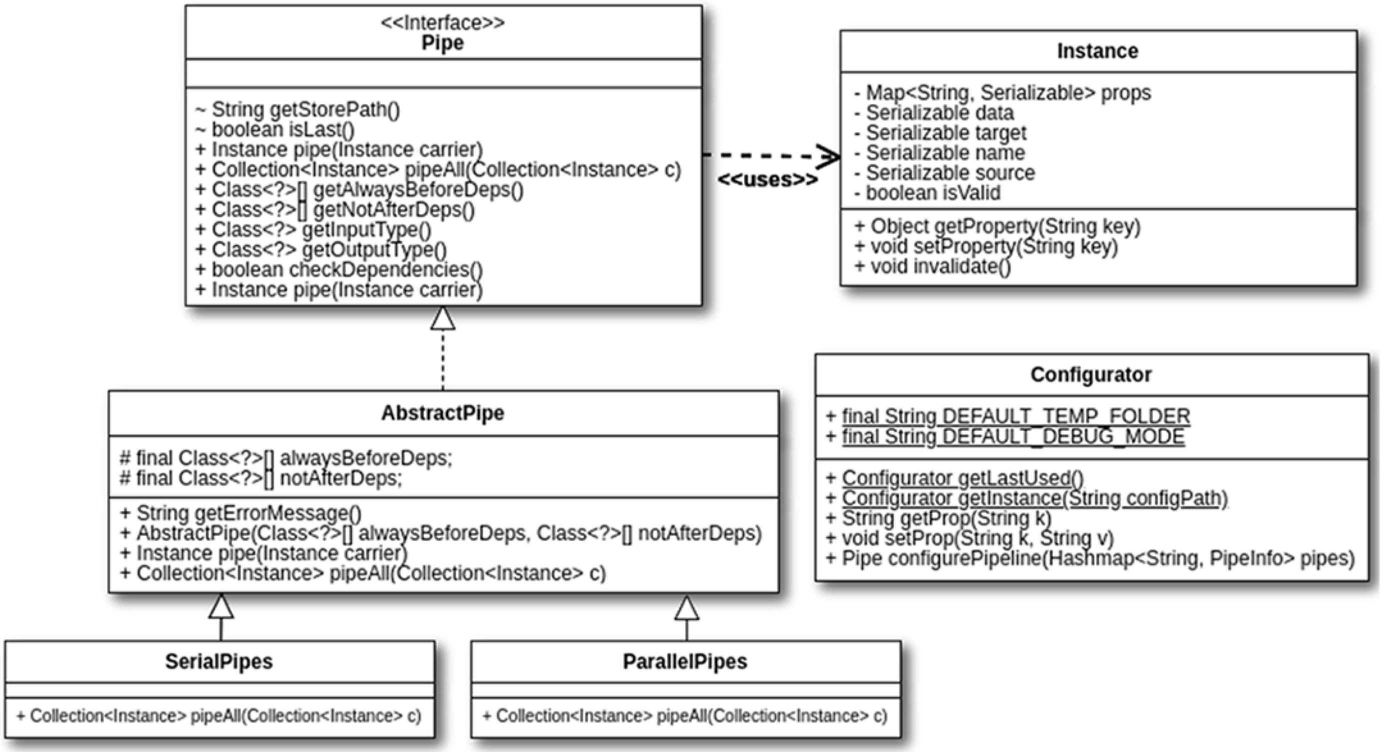
Similar to Mallet, BDP4J uses the Instance class (see Fig. 2) to represent a specific case (including raw data, properties, and the target solution) and solve a certain problem. As an example, in order to define an e-mail classifier, an Instance would represent the information of a specific message. The raw message contents would be included in source/data attributes; interesting features extracted through pipelining process would then be stored in props attribute, and the class of the message would be represented in target. Moreover, data attribute would be transformed through the pipelining process (from raw data to token list, feature vector…).

BDP4J tasks are represented as simple pipes (by implementing the Pipe interface or, even better, by extending AbstractPipe class). Moreover, the orchestration of pipes is defined through classes SerialPipes and ParallelPipes. These details together with the basic design of BDP4J are shown in Fig. 2. To facilitate readability, only relevant methods/attributes have been included in the classes shown in the diagram. Please refer to the Javadoc documentation (generated through the build process) to obtain a complete list.

As we can deduct from Fig. 2, BDP4J Pipe interface was created to allow developers to implement their tasks by extending, if necessary, from other classes. However, the use of AbstractPipe abstract class provides the implementation of all methods of Pipe interface, except for pipe (which stands for the specific work that should be done), to simplify the development of tasks. SerialPipes class keeps the same functionality that was originally provided in Mallet software while slightly changing the instance processing flow. Finally, BDP4J adds the support for parallel tasks execution through the class ParallelPipes. Configurator class is used for loading pipelines from files and is explained below.

Each pipe or task implemented in BDP4J must implement the methods getInputType and getOutputType to indicate the type of data included in the Instance before and after executing it. This information is used by SerialPipes and ParallelPipes to check data types in the orchestration (feature *i*). Additionally, these types should dynamically be checked with the Instance after executing the task.

When creating a task by extending the AbstractPipe class, the developer must call on its constructor (through super) to specify “Always Before” (alwaysBeforeDeps attribute) and “Not After” (notAfterDeps attribute) task execution constraints (feature *ii*). “Always



<<Interface>>
Pipe
~ String getStorePath()
~ boolean isLast()
+ Instance pipe(Instance carrier)
+ Collection<Instance> pipeAll(Collection<Instance> c)
+ Class<?>[] getAlwaysBeforeDeps()
+ Class<?>[] getNotAfterDeps()
+ Class<?> getInputType()
+ Class<?> getOutputType()
+ boolean checkDependencies()
+ Instance pipe(Instance carrier)
<<uses>>
Instance
- Map<String, Serializable> props
- Serializable data
- Serializable target
- Serializable name
- Serializable source
- boolean isValid
+ Object getProperty(String key)
+ void setProperty(String key)
+ void invalidate()
Configurator
+ final String DEFAULT_TEMP_FOLDER
+ final String DEFAULT_DEBUG_MODE
+ Configurator getLastUsed()
+ Configurator getInstance(String configPath)
+ String getProp(String k)
+ void setProp(String k, String v)
+ Pipe configurePipeline(Hashmap<String, PipeInfo> pipes)
AbstractPipe
# final Class<?>[] alwaysBeforeDeps;
# final Class<?>[] notAfterDeps;
+ String getErrorMessage()
+ AbstractPipe(Class<?>[] alwaysBeforeDeps, Class<?>[] notAfterDeps)
+ Instance pipe(Instance carrier)
+ Collection<Instance> pipeAll(Collection<Instance> c)
SerialPipes
+ Collection<Instance> pipeAll(Collection<Instance> c)
ParallelPipes
+ Collection<Instance> pipeAll(Collection<Instance> c)

Fig. 2. Main classes comprising BDP4J.

Before" constraints indicate which tasks must be executed before the current one (for instance, the identification of the text language should probably be done before expanding abbreviations and/or slang terms to select the appropriate dictionary). Moreover, "Not After" dependencies indicate the tasks that cannot be executed after the execution of the current one (for instance, the recognition of acronyms should not be run after changing text to lowercase or removing punctuation marks). checkDependencies and getErrorMessage methods included in AbstractPipe class allow checking these constraints and obtaining information about the specific errors found.

BDP4J has divided the tasks in the following categories (feature *iii*): PropertyComputingPipe used only for computing instance properties; TeePipe used for storing instances when needed by user; TargetAssigningPipe used for assigning the real target class on classification; and TransformationPipe, which transforms the instance data. These task categories allow including additional constraints on tasks. Hence, the input and output data types of a task (getInputType and getOutputType) should be the same for all tasks except for TransformationPipe, which could be different. Moreover, the number of instance properties should be increased after an instance is processed through a PropertyComputingPipe, and the target attribute of an Instance should not be null after the execution of a TargetAssigningPipe. Finally, the number of TargetAssigningPipe included in a pipeline must be zero or one. In the future, the classification of tasks could be used to automatically group the functionalities in GUI pipeline management tools.

Currently, the support of parallel processing (feature *iv*) schemes in BDP4J is limited to the use of ParallelPipes task orchestration class. When several tasks are marked to be executed in parallel, they are performed in separate threads. Although we believe that we can take advantage of load balancing clustering schemes when multiple computers are available (feature *v*), its support has not been implemented yet and should be carefully designed.

Moreover, in order to adequately support burst data processing and data sharing (feature *vi*), the original Mallet processing method for data instances was changed. Mallet instance processing model is based on processing one instance through all tasks included in the pipeline. However sometimes we need all instances included in a data burst to be processed by a certain task in order to fill the shared information (for instance, when a text mining task builds a dictionary that will be used by other tasks to create a dense vector representation or a standard CSV –Comma-Separated Values– file). To cope with these situations, BDP4J executes each task on all available instances of the burst before starting the execution of the next task. This behaviour was implemented in pipeAll methods of orchestration schemes (SerialPipes and ParallelPipes).

BDP4J is able to load pipeline orchestration from XML files (feature *vii*). This functionality is connected with the dynamic loading of tasks from *.jar files (feature *viii*) and cannot be used independently. In order to implement the first functionality, we took advantage of Document Object Model (DOM) Application Programming Interface (API). Configurator class (see Fig. 2) provides the functionality of loading the pipeline configuration using DOM API. The XML file should contain the configuration/general/pluginsFolder parameter, which allows defining the directory where the *.jar files containing task definitions are located. The dynamic loading of tasks from jar files was implemented through the default Java service-provider loading facility (java.util.ServiceLoader<S> included in Java 8). Using this facility, BDP4J searches for Pipe implementations included in the location specified by pluginsFolder parameter. To facilitate the use of standard Java service loader by avoiding the manual creation of file META-INF/services/org.bdp4j.pipe.Pipe, all classes implementing tasks can be annotated with the @AutoService(Pipe.class) [42]. Fig. 3 shows how the orchestration of a pipeline can be easily represented in XML format (Fig. 3a) and loaded for its execution using BDP4J framework (Fig. 3b).

```
<?xml version=”1.0”?>
<configuration>
 <!-- General properties -->
 <general>
   <samplesFolder>./samples</samplesFolder>
   <pluginsFolder>./plugins</pluginsFolder>
   <outputDir>./output</outputDir>
   <tempDir>./temp</tempDir>
 </general>
 <!-- the pipeline orchestration -->
 <pipeline resumable=”yes” debug=”no”>
  <serialPipes>
    <pipe>
     <name>File2TargetAssignPipe</name>
    </pipe>
    <pipe>
     <name>File2StringPipe</name>
    </pipe>
    <pipe>
     <name>String2TokenArray</name>
    </pipe>
    <pipe>
     <name>TokenArray2FeatureVector</name>
    </pipe>
    <pipe>
     <name>
        GenerateFeatureVectorOutputPipe
     </name>
     <params>
      <pipeParameter>
         <name>outFile</name>
         <value>out.csv</value>
      </pipeParameter>
     </params>
    </pipe>
  </serialPipes>
 </pipeline>
</configuration>
```

(a)

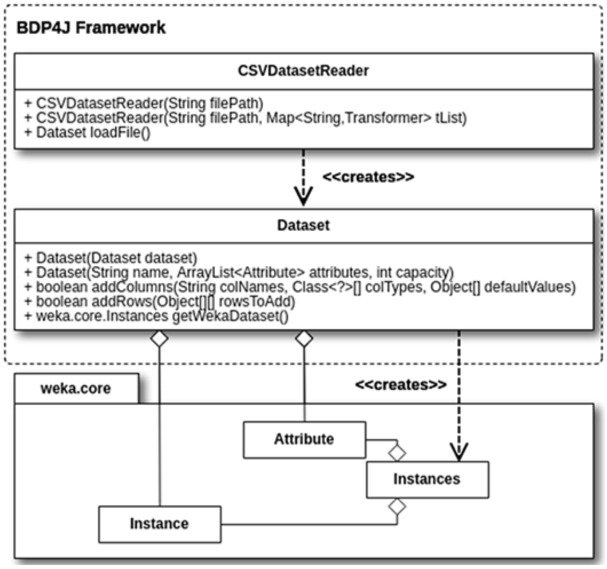
```
/* Load XML configuration */
Configurator cfg =
  Configurator.getInstance(“configuration.xml”);

/*Load tasks from jar files*/
PipeProvider pipeProvider =
new PipeProvider(
     cfg.getProp(Configurator.PLUGINS_FOLDER)
   );
HashMap<String, PipeInfo> pipes =
   pipeProvider.getPipes();

/*Load the pipeline orchestration*/
Pipe p =
  Configurator.configurePipeline(pipes);
logger.info(“orchest:” + p.toString() + “\n”);

/*Check orchestration dependencyes*/
if (!p.checkDependencies()) {
       logger.fatal(
           “[CHECK DEPENDENCIES] “+
            AbstractPipe.getErrorMessage()
       );
       System.exit(-1);
}

/*Load and pipe the current burst*/
ArrayList<Instance> burst = …
p.pipeAll(burst);
```

(b)

Fig. 3. XML orchestrating facilities included in BDP4J: a) XML structure used to define task orchestration (configuration.xml) and b) BDP4J source to load task orchestration.

The source code included in Fig. 3b (configurePipeline method) makes it possible to automatically instantiate as many SerialPipes and ParallelPipes as necessary to load the orchestration and configure tasks (because task configuration parameters are also included in XML, Fig. 3a).

A typical BDP4J pre-processing pipeline would contain one or more TeePipes. As stated before, the mission of TeePipes is to bring together the information of instances to generate datasets in memory (through using Dataset class shown in Fig. 4a) or in disk (through using CSVDatasetWriter class) and continue the execution of the rest of the pipeline. This mission fits with the dataset generation issue shown in Fig. 1. One of the most important features added to CSVDatasetWriter and Dataset classes is the possibility of dynamically growing the number of columns and rows. After processing some data bursts, they can be analysed by using external ML APIs (feature *ix*). Due to the popularity of Weka ML library, we have implemented a feature to transform a BDP4J dataset stored in memory (Dataset class) to Weka dataset (weka.core.Instances) through the getWekaDataset method. Additionally, the CSVDatasetReader class implements the loading of a CSV file to instantiate a BDP4J Dataset (Fig. 4b).


BDP4J Framework
CSVDatasetReader
+ CSVDatasetReader(String filePath)
+ CSVDatasetReader(String filePath, Map<String,Transformer> tList)
+ Dataset loadFile()
<<creates>>
Dataset
+ Dataset(Dataset dataset)
+ Dataset(String name, ArrayList<Attribute> attributes, int capacity)
+ boolean addColumns(String colNames, Class<?>[] colTypes, Object[] defaultValues)
+ boolean addRows(Object[][] rowsToAdd)
+ weka.core.Instances getWekaDataset()
weka.core
<<creates>>
Attribute
Instances
Instance

(a)

```
CSVDatasetReader csvdr =
 new CSVDatasetReader(“out.csv”);
Dataset ds =  csvdr.loadFile();
Instances wekaDS =
 ds.getWekaDataset();

wekaDS.deleteStringAttributes();
wekaDS.setClassIndex(
 wekaDS.numAttributes() - 1
);
int num = wekaDS.numInstances();
int start = (num * 80) / 100;
int end = num - start;

Instances trn =
 new Instances(wekaDS, 0, start);
Instances tst =
 new Instances(wekaDS, start, end);

try {
 Evaluation rfEval = new Evaluation(tst);
 RandomForest rf = new RandomForest();
 rf.buildClassifier(trn);
 rfEval.evaluateModel(rf, tst);
} catch (Exception ex) {}
```

(b)

Fig. 4. Weka integration facilities included in BDP4J: a) BDP4J and Weka interaction architecture and b) BDP4J and Weka interaction snippet.

As we can see from Fig. 4b, source lines highlighted in bold (first three sentences) allow the loading of Weka dataset, while the remaining lines show how we can take advantage of Weka API to run a Random Forest [43] classifier.

Furthermore, following the same orchestration behaviour of SerialPipes and ParallelPipes classes, we implemented ResumableSerialPipes and ResumableParallelPipes classes respectively. These classes support the resumption of the execution of a pipeline (feature *x*) that has been stopped for any reason (an application failure, accidental power down of computer...). The inner operation of the mentioned classes includes saving the state of instances after executing each task to allow resuming the pipeline from the last successfully executed task. One of the most relevant challenges to implement the resuming functionality was its compatibility with the data sharing between tasks. To this end, pipes can be marked by implementing SharedDataConsumer and/or SharedDataProducer interfaces. These interfaces include the methods readFromDisk and writeToDisk, respectively, to allow the programmer defining how to save and read the shared information to make it available when resuming the execution of a pipeline. These details are included in Fig. 5.

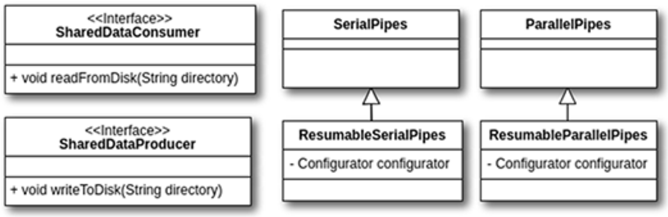


Fig. 5. BDP4J architecture details to support resuming function.

A resumable pipeline could be easily instantiated by defining the orchestration in source code (Fig. 4b) and replacing the use of SerialPipes and ParallelPipes by ResumableSerialPipes and ResumableParallelPipes, respectively. Additionally, the resuming behavior can be achieved with an XML file (Fig. 4a) using resumable and debug modifiers included in pipeline tag. When resumable is set to “yes” (“true” or “1”), the pipeline can be resumed. The debug modifier included in pipeline could be set to “no” (“false” or “0”) only if the last complete result of a task remains stored on disk, or “yes” if all partial results are kept on disk (useful to manually drop steps and repeat tasks and only applicable to ResumableSerialPipes).

Finally a GUI application (feature *xi*) to orchestrate tasks was created using JGraphX [44] framework. The GUI is launched when using “gui” as the first parameter for the execution of the main class org.bdp4j.Main and allows visually defining, executing and saving a pipeline orchestration.

BDP4J is released as open source (feature *xii*). Its tasks can only be developed using Java technology (non-Java programming languages are not supported) (feature *xiii*). Moreover, BDP4J does not provide resource management utilities (feature *xiv*) and does not allow optimising the use of resources. In this sense, the limitation of RAM could be easily done by using the -Xmx parameter of Java Virtual Machine (JVM) and the number of cores via the GNU/Linux taskset utility. Therefore, the development of additional resource management functionalities (mainly computing RAM requirements) will be addressed as future works.

The instance invalidation (feature *xv*) was implemented by adding the invalidate method in the Instance class (which can be called from task definition) and the management of invalidated instances in the BDP4J orchestration subsystem (SerialPipes/ResumableSerialPipes and ParallelPipes/ResumableParallelPipes). Methods pipe and pipeAll implemented in the orchestration classes skip the processing of invalidated instances.

Additionally, the detection of the last data instance (feature *xvi*) is provided in the AbstractPipe class by providing a default implementation for isLast method. This implementation marks an instance *i* as last when the instance is processed alone through a call to pipe method or when *i* is the last valid instance of a collection of data (data burst) that is processed by calling the pipeAll method.

Given the support of resuming a pipeline from a certain position, we find it appropriate to take advantage of this feature in order to implement the debugging mode function (feature *xvii*). The debugging mode function allows developers of a task to avoid the processing of all previous required tasks (when they were previously executed) in order to reduce the time required to test whether the new task is operating properly. The code snippets (XML and Java) included in Fig. 6 provide a detailed description of how to take advantage of this functionality when the orchestration is defined in XML (Fig. 6a) or in Java (Fig. 6b). We highlighted in bold the instructions used to select the task that is being debugged.

As we can see from Fig. 6, the orchestrations defined in both columns are exactly the same. Hence, a task can be marked for debug when the orchestration is defined in XML or in Java, providing great flexibility for developers. In the case of using the XML to define the orchestration, the task (pipe) that is being debugged should include a debug tag, and the entire pipeline should be executed in resumable mode (resumable=”yes”). Additionally, for source code orchestrations, the debug mode implies the use of ResumableSerialPipes and ResumableParallelPipes classes.

As shown in this section, BDP4J covers important limitations found in current pipelining software. The implemented features provide flexibility to developers, allowing them to analyse and pre-process data obtained from different sources in order to solve different kinds of problems. Subsection A shows a case study of our framework to address the pre-processing and classification of SMS –Short Message Service– spam messages.

A. Using BDP4J

In order to show the simplicity of using the BDP4J framework, we developed a case study to apply different text pre-processing techniques over SMS spam messages included in the SMS Spam Corpus v.0.1 [45]. The project has been publicly shared through GitHub (https://github.com/sing-group/bdp4j_sample). This project contains a collection of eight text pre-processing tasks (some of them trivial) implemented in the Java package org.bdp4j.sample.pipe.impl. The implemented pipes (sorted by the right execution order) make it possible to: create a property with the size of the text file (FileSizePipe); load the target attribute from the file (File2TargetAssigningPipe); load SMS text from file (File2StringPipe); create a property with the length of the SMS text (MeasureLengthPipe); add all information generated up to this step to a CSV file (GenerateStringOutputPipe); tokenize the text of each instance (String2TokenArray); create a feature vector for each instance (TokenArray2FeatureVector); and add all information to a CSV File (GenerateFeatureVectorOutputPipe).

The class org.bdp4j.sample.Main (main) creates instances from the files included in the samples directory. In detail during this process, data and source attributes are initialized with an instance of java.io.File representing the disk file containing SMS data. Instances are divided into three groups to simulate a burst operation. Moreover, this class also orchestrates a pipeline with all tasks defined in the project and use it to process the three bursts. After processing the instances, a NaïveBayes [46] classifier from Weka is executed over the generated data and the confusion matrix results (true positives, true negatives, false positives, and false negatives) are printed via system standard

```
<?xml version="1.0”?>
<configuration>
 <!-- General properties -->
 <general>
  <samplesFolder>./samples</samplesFolder>
  <pluginsFolder>./plugins</pluginsFolder>
  <outputDir>./output</outputDir>
  <tempDir>./temp</tempDir>
 </general>
 <!-- the pipeline orchestration -->
 <pipeline resumable=”yes” debug=”yes”>
  <serialPipes>
   <pipe>
    <name>File2TargetAssignPipe</name>
   </pipe>
   <pipe>
    <name>File2StringPipe</name>
   </pipe>
   <pipe>
    <name>String2TokenArray</name>
    <debug/>
   </pipe>
   <pipe>
    <name>
      TokenArray2FeatureVector
    </name>
   </pipe>
   <pipe>
    <name>
      GenerateFeatureVectorOutputPipe
    </name>
    <params>
     <pipeParameter>
      <name>outFile</name>
      <value>out.csv</value>
     </pipeParameter>
    </params>
   </pipe>
  </serialPipes>
 </pipeline>
</configuration>
```

(a)

```
/* Debug String2TokenArray pipe */
String2TokenArray s2ta =
     new String2TokenArray();
s2ta.setDebugging(true);

/* Create the processing pipe */
AbstractPipe p =  new ResumableSerialPipes(
  new AbstractPipe[]{
   new File2TargetAssignPipe(),
   new File2StringPipe(),
   s2ta,
   new TokenArray2FeatureVector(),
   new GenerateFeatureVectorOutputPipe()
  }
 );

logger.info(“orchest:” + p.toString() + “\n”);

/*Check orchestration dependencies*/
if (!p.checkDependencies()) {
  logger.fatal(
   “[CHECK DEPENDENCIES] “+
   AbstractPipe.getErrorMessage()
  );
  System.exit(-1);
}

/*Load and pipe the current burst*/
ArrayList<Instance> burst = …
p.pipeAll(burst);
```

(b)

Fig. 6. BDP4J debug mode snippet: a) Enabling the debugging for a task in a XML orchestration and b) Enabling the debugging for a task in source code.

output. Additionally, two CSV files are generated on a disk (one per each tee pipe called output.csv and output2.csv) that contains the pre-processing results.

As can be easily found in the source code, two PropertyComputingPipe (FileSizePipe and MeasureLengthPipe), one TargetAssigningPipe (File2TargetAssignPipe), two TeePipe (GenerateFeatureVectorOutputPipe and GenerateStringOutputPipe) and three TransformationPipe (File2StringPipe, String2TokenArray, TokenArray2FeatureVector) tasks are provided as examples. Moreover, TokenArray2FeatureVector and GenerateFeatureVectorOutputPipe share a dictionary (org.bdp4j.sample.types.Dictionary) and hence, they implement the interfaces SharedDataProducer and SharedDataConsumer respectively to allow resuming and debugging functions.

Since Main class is provided and task implementations have been annotated with @AutoService annotation, the example could be executed by launching the generated jar file or by generating an alternative orchestration with BDP4J GUI.

The creation of the bdp4j_sample project provides a simple form for showing the use of the BDP4J pipelining framework to other researchers and developers. During the last months, we have been using BDP4J for pre-processing data in the research activities of our research group. This work has given us some conclusions for usage, summarized in subsection B.

B. Performance Evaluation

We carried out a performance evaluation of BDP4J and other available frameworks introduced in Section II. In order to accurately evaluate the performance of the analysed frameworks we use void tasks (those doing nothing). Fig. 7 represents the benchmarking protocol designed for the evaluation of the frameworks.

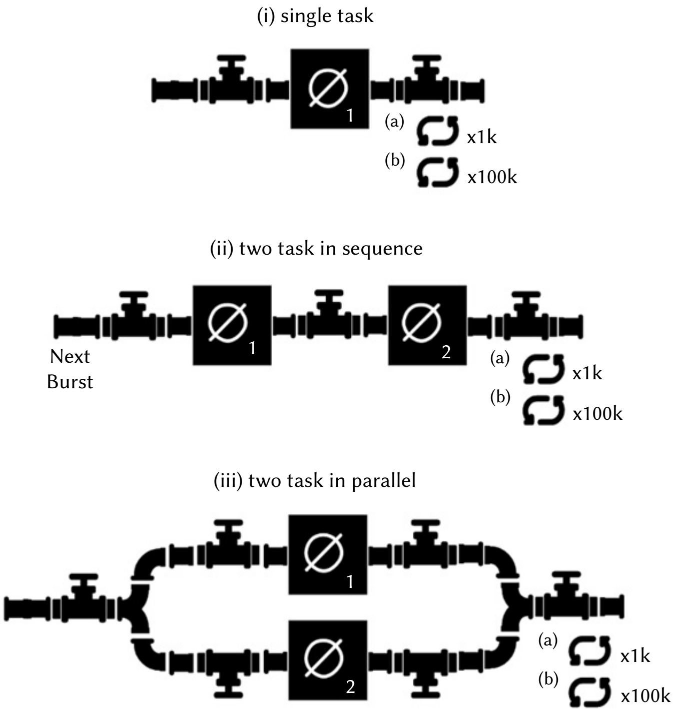
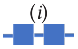
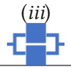


Fig. 7. Benchmarking design.

As shown in Fig. 7 our experimental protocol comprises three simple computational pipelines: (*i*) a pipeline composed exactly by one task, (*ii*) a pipeline that comprises the execution of two tasks in sequence and (*iii*) a pipeline that executes two parallel tasks. These pipelines were executed 1,000 and 100,000 times in order to compare the impact of the checks made during the execution in the performance of the framework. The source code used for experimentation is available online [47]. The benchmark was executed in a computer with 64 gigabytes of RAM and an Intel i7-6700 processor with four cores (eight threads). Table II shows the evaluation results.

TABLE II. Performance Comparison of Analyzed Frameworks

Pipeline	(*i*)	(*i*) 	(*iii*) 
x 1k			
COMPI	2s, 554ms	3s, 266ms	3s, 149ms
Cromwell	50s, 076ms	1m, 24s, 113ms	1m, 28s, 917s
Drake	31s, 133ms	30s, 946ms	95s 90ms
Mallet	1.05ms	1.09ms	unsupported
MLPipeline	2s, 227ms	2s, 321ms	unsupported
BDP4J	1.03 ms	4.42ms	56.34ms
x 100k			
COMPI	48s 928ms	1m 37s 83ms	1m, 33s, 436ms
Cromwell	unsupported	unsupported	unsupported
Drake	51m, 53s	51m, 34s	2h, 38m 29s
Mallet	10 ms	9 ms	unsupported
MLPipeline	2s, 475ms	2s, 494ms	unsupported
BDP4J	12.93 ms	15.24 ms	801.77 ms

As shown in Table II, Drake (which is written in R but allows executing tasks written in Java) fairly achieves the worst performance while Mallet is the most efficient framework. However, the performances achieved in Mallet and BDP4J when executing pipelines (*i*) and (*ii*) are very similar (a few milliseconds). The increase of time required by BDP4J for the execution of pipelines containing tasks executed in parallel is caused by the need to clone the instances in order to alleviate the programmer of the issues arising from concurrent programming. The impact on performance of this decision is significative, however the time required for the development of tasks is considerably reduced. Additionally, Cromwell has some issues with memory when processing large pipelines (repeating 100000 times the pipeline is not possible with 64GB of RAM). Keeping in mind the benefits of our proposal, we really believe BDP4J is a reliable solution for analysing data using Java technology.

C. Learned Lessons and Main Outcomes

Our use of BDP4J for the development and execution of different pre-processing tasks over the last several months have led us to important findings. One of the most important findings is the great amount of human error that occurs during the development of applications. A significant number of these errors could be detected by using constraints, type checks, and dependencies implemented by BDP4J. People often wrongly think that constraints, type checks, and dependencies included in different programming models are merely time-consuming issues that negatively affect development because any IT professional can develop software with no errors. However, our experience has shown us that the existence of these elements really helps developers to avoid software errors and cause a low impact on the time required for defining pre-processing tasks.

Additionally, the use of debug mode has significantly sped up the development and debugging of tasks, given that the whole execution process is executed only once. In fact, when a pipeline is executed a second time, the task marked to be debugged will be the first executed. As long as the tasks that will be debugged can be quickly defined, the feature is very useful for developers.

Moreover, we have developed an extremely simple mechanism for defining tasks. When extending AbstractPipe class, only the methods getInputType, getOutputType and pipe (execute the task) should be defined. The BDP4J framework adds many sources to check constraints, types, and operation with no effort on the part of the developer. Most developers who tried BDP4J really appreciate the effort to minimize the source they should write for developing tasks.

Finally, we would like to highlight the use of Apache Maven as a software management and building tool that facilitates compilation, packaging, and execution tasks. This finding, together with those previous mentioned, allow us to understand that we are working in the right direction to define the BDP4J framework. The next section provides detailed conclusions and future work to improve this framework.

IV. Conclusions and Future Work

During the last years, the exploitation of data has been increasingly used as a method to solve many problems and understand the inner details of real life. The main challenge to address in data exploitation is the existence of effective programming models able to take advantage of available data. In the current context, popular developing paradigms such as MapReduce are being abandoned while pipelining schemes are becoming more and more popular. Our intention in this work was to design and provide new and interesting functionalities for pipelining software. Particularly, pipelining frameworks should provide simple ways for defining tasks and incorporate mechanisms for early detecting errors.

We designed functionalities to avoid the processing limitations found in MapReduce schemes and other pipelining frameworks. Particularly, we introduce the burst-processing scheme that can be easily applied to the pre-processing of streaming sources of data (such as Twitter streams), while still allowing for the old batch processing schemes. Burst processing combined with the detection of the last data instance in any task (for flushing/closing files) makes it possible to analyse data after the execution of some bursts, thus avoiding the need for processing all input data before they are analysed.

Finally, it should be noted that a great amount of technology and new processing techniques have been made available to data mining developers. As main contributions of this work (processing techniques), we would highlight the number of techniques used to detect errors (constraints, type checks and task dependencies), the resuming (and debug) capabilities of pipelines, and the instance invalidation (never seen before). These technologies can also be applied to improve other frameworks (especially those developed in other programming languages).

Future work comprises the design of execution schemes to distribute computational requirements in a cluster of computers. To this end, we believe that a careful examination of available load balancing schemes would provide effective load distribution mechanisms to implement this functionality. Additionally, this feature will be complemented with the assessment of the amount of RAM necessary to execute a pipeline over a burst with certain size (resource management).

Acknowledgment

D. Ruano-Ordás was supported by a post-doctoral fellowship from Xunta de Galicia (ED481D-2021/024). Additionally, this work was funded by the project Semantic Knowledge Integration for Content-Based Spam Filtering [grant number TIN2017-84658-C2-1-R] from the Spanish Ministry of Economy, Industry and Competitiveness (SMEIC), State Research Agency (SRA) and the European Regional Development Fund (ERDF); and Consellería de Educación, Universidades e Formación Profesional (Xunta de Galicia) under the scope of the strategic funding of Competitive Reference Group [grant number ED431C2018/55-GRC].

SING group thanks CITI (Centro de Investigación, Transferencia e Innovación) from University of Vigo for hosting its IT infrastructure.

References

[1] I. M. Dunham, "Big Data: A Revolution That Will Transform How We Live, Work, and Think",*The AAG Review of Books*,vol. 3, no. 1,pp. 19–21,Jan. 2015.

[2] Q. Qi, F. Tao, "Digital Twin and Big Data Towards Smart Manufacturing and Industry 4.0: 360 Degree Comparison",*IEEE Access*,vol. 6,pp. 3585–3593,2018.

[3] V. Kalavri, V. Vlassov, "MapReduce: Limitations, Optimizations and Open Issues," in *2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, 2013, pp. 1031–1038.

[4] D. Miner, A. Shook, *Mapreduce Design Patterns Building Effective Algorithms and Analytics for Hadoop and Other Systems.* Oreilly & Associates Inc, 2012.

[5] Apache Software Foundation, "Apache Hadoop." 2018.

[6] Amazon, "Amazon Elastic MapReduce." 2019.

[7] Disco Project, "DisCo MapReduce." 2014.

[8] S. Papadimitriou, J. Sun, "DisCo: Distributed Co-Clustering with Map-Reduce: A Case Study towards Petabyte-Scale End-to-End Mining," in *2008 Eighth IEEE International Conference on Data Mining*, 2008, pp. 512–521.

[9] Apache Software Foundation, "Apache Spark - Unified Analytics Engine for Big Data." 2018.

[10] J. Zeng, B. Plale, "Data Pipeline in MapReduce," in *2013 IEEE 9th International Conference on e-Science*, 2013, pp. 164–171.

[11] P. O'Donovan, K. Leahy, K. Bruton, D. T. J. O'Sullivan, "An Industrial Big Data Pipeline for Data-Driven Analytics Maintenance Applications in Large-Scale Smart Manufacturing Facilities",*Journal of Big Data*,vol. 2, no. 1,p. 25,Dec. 2015.

[12] P. Di Tommaso, "Awesome Pipeline: A Curated List of Awesome Pipeline Toolkits." 2018.

[13] Amazon, "AWS Data Pipeline." 2019.

[14] Snaplogic, "SnapLogic Intelligent Integration Platform," 2019. [Online]. Available: https://www.snaplogic.com/products/intelligent-integration-platform. [Accessed: 21-Jun-2020].

[15] Alooma, "Alooma Enterprise Data Pipeline." 2019.

[16] S. G. Ahmad, C. S. Liew, M. M. Rafique, E. U. Munir, "Optimization of Data-Intensive Workflows in Stream-Based Data Processing Models",*The Journal of Supercomputing*,vol. 73, no. 9,pp. 3901–3923,Sep. 2017.

[17] G. Kougka, A. Gounaris, A. Simitsis, "The Many Faces of Data-Centric Workflow Optimization: A Survey",*International Journal of Data Science and Analytics*,vol. 6, no. 2,pp. 81–107,Sep. 2018.

[18] J. Leipzig, "A Review of Bioinformatic Pipeline Frameworks",*Briefings in Bioinformatics*,p. bbw020,Mar. 2016.

[19] P. A. Ewels *et al.*, "The Nf-Core Framework for Community-Curated Bioinformatics Pipelines",*Nature Biotechnology*,vol. 38, no. 3,pp. 276–278,Mar. 2020.

[20] M. Bourgey *et al.*, "GenPipes: An Open-Source Framework for Distributed and Scalable Genomic Analyses",*GigaScience*,vol. 8, no. 6,Jun. 2019.

[21] D. Swersky, "Top 43 Programming Languages: When and How to Use Them," 2018. [Online]. Available: https://raygun.com/blog/programming-languages/. [Accessed: 21-Jun-2020].

[22] E. Frank, M. A. Hall, I. H. Witte, *The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques,"* Fourth Edi. Morgan Kaufmann Publishers Inc., 2016.

[23] A. Moro, R. Navigli, "Babelfy." 2014.

[24] Y. Lage, J. R. Méndez, M. Novo-Lourés, "Big Data Pre-Processing For Java (BDP4J)." 2018.

[25] F. Lordan *et al.*, "ServiceSs: An Interoperable Programming Framework for the Cloud",*Journal of Grid Computing*,vol. 12, no. 1,pp. 67–91,Mar. 2014.

[26] R. M. Badia *et al.*, "COMP Superscalar: An Interoperable Programming Framework",*SoftwareX*,vol. 3–4,pp. 32–36,Dec. 2015.

[27] T. Burdett, N. Kurbatova, D. Hastings, Emma Faulconbridge, Adam Mapleson, R. Davey, "Conan2 Lightweight Workflow Manager." 2019.

[28] J. Bingham, S. Davis, N. Deflaux, "Dockerflow: A Workflow Runner That Uses Dataflow to Run a Series of Tasks in Docker with the Pipelines API," 2017. [Online]. Available: https://github.com/googlegenomics/dockerflow.

[29] Google Inc, "Cloud Dataflow Documentation," 2019. [Online]. Available: https://cloud.google.com/dataflow/docs/?hl=es-419. [Accessed: 21-Jun-2020].

[30] Netflix, "Suro: Netflix Distributed Data Pipeline." 2012.

[31] J. M. Wozniak, M. Wilde, I. T. Foster, "Language Features for Scalable Distributed-Memory Dataflow Computing," in *Fourth Workshop on Data-Flow Execution Models for Extreme Scale Computing*, 2014, pp. 50–53.

[32] J. M. Wozniak, M. Wilde, I. T. Foster, "Swift Tutorial for Running on Localhost," 2014. [Online]. Available: http://swift-lang.org/tutorials/localhost/tutorial.html. [Accessed: 21-Jun-2019].

[33] M. Hategan *et al.*, "Swift-Lang, Swift-K," 2019. [Online]. Available: https://github.com/swift-lang/swift-k. [Accessed: 21-Jun-2019].

[34] H. López-Fernández, O. Graña-Castro, A. Nogueira-Rodríguez, M. Reboiro-Jato, D. Glez-Peña, "Compi: A Framework for Portable and Reproducible Pipelines",*PeerJ Computer Science*,vol. 7,p. e593,Jun. 2021.

[35] Broad Institute, "Cromwell: Workflow Management System Geared towards Scientific Workflows." 2019.

[36] A. Malloy *et al.*, "Drake." 2015.

[37] S. Fong, Y. Zhuang, J. Li, R. Khoury, "Sentiment Analysis of Online News Using MALLET," in *2013 International Symposium on Computational and Business Intelligence*, 2013, pp. 301–304.

[38] A. K. McCallum, "MALLET: A Machine Learning for Language Toolkit." 2002.

[39] Apache Software Foundation, "Apache Spark: ML Pipelines," 2018. [Online]. Available: https://spark.apache.org/docs/latest/ml-pipeline.html. [Accessed: 21-Jun-2020].

[40] A. Liu, *Apache Spark Machine Learning Blueprints*, First. Birmingham, UK: PACKT Publishing Ltd., 2016.

[41] D. S. F. Long, D. Mohindra, R.C. Seacord, D.F. Sutherland, "Svoboda, Java Coding Guidelines: 75 Recommendations for Reliable and Secure Programs",*Addison-Wesley*,2013.

[42] Google LLC, "AutoService: A Collection of Source Code Generatos for Java." 2013.

[43] L. Breiman, "Random Forests",*Machine Learning*,vol. 45, no. 1,pp. 5–32,2001.

[44] M. R. G. Alder, D. Benson, "Jgraph/Jgraphx." 2014.

[45] E. P. S. J.M. Gómez Hidalgo, "SMS Spam Corpus v.0.1," 2011.

[46] A. Pérez, P. Larrañaga, I. Inza, "Bayesian Classifiers Based on Kernel Density Estimation: Flexible Classifiers", *International Journal of Approximate Reasoning*, vol. 50, no. 2,pp. 341–362,Feb. 2009.

[47] M. Novo-Lourés, Y. Lage, R. Pavón, R. Laza, D. Ruano-Ordás, J. R. Mendez, "Benchmarking Code for Pipeline-Based Frameworks." 2021.

María Novo

She was born in Galicia (Spain) in 1983. She graduated from the University of Vigo on Computer Science Engineering and she holds an MSc in E-Commerce from the University of Salamanca. At present, she is a researcher on the SING research group, where she is developing her PhD in spam filtering using Big Data and Machine Learning techniques.

Yeray Lage

He was born in Galicia (Spain) in 1995. He recently finished his Computer He was born in Galicia (Spain) in 1995. He recently finished his Computer Science Degree making his end-of-degree project collaborating with SING group. Currently, he is working as Full-Stack Developer in a company specialized in Web Development.

Reyes Pavón

She is a PhD from the University of Vigo and a member of SING research group. Currently, she is Associate Professor in the Computer Science Department of the University of Vigo. Her main research interests are Artificial Intelligence, classification methods and spam filtering. She is joint author of several articles published in international prestigious journals.

Rosalía Laza

She received a PhD in Computer Science from the University of Vigo in 2003. At present, she is member of SING research group and she is Associate Professor in the Computer Science Department of the University of Vigo. She is co-author of several books, book chapters, and articles published in international prestigious journals; most of these present practical and theoretical achievements of case-based reasoning systems, intelligent artificial paradigm, classification methods and spam filtering.

David Ruano-Ordás

He was born in Galicia (Spain) in 1985 and received his PhD in Computer Science from the University of Vigo (Spain) in 2015. He is computer science engineer with high experience on Linux administration and software development under the ANSI/C standard. He collaborates as researcher with the SING group belonging to the University of Vigo. Regarding to the research experience he is mainly focused in the Artificial Intelligence area (automatic learning, or evolutionary algorithms) applied to spam filtering and drugs-discovery domain. Finally, he has participated in some regional research projects and has been co-author of several articles published in journals belonging recognized editorials such as Hindawi, Springer or Elsevier. (http://www.drordas.info/).

José R. Méndez

He was born in Galicia (Spain) in 1977. Currently, he works at the computer science department of University of Vigo. He worked as a system administrator, software developer, and IT (Information Technology) consultant in civil services and industry during 10 years. He is an active researcher belonging to SING group and, although collaborates in different applications machine learning, his main interests are the development and improvement of anti-spam filters. (http://sing-group.org/).

CompareML: A Novel Approach to Supporting Preliminary Data Analysis Decision Making

Antonio Jesús Fernández-García[1]*, Juan Carlos Preciado[2], Alvaro E. Prieto[2], Fernando Sánchez-Figueroa[2], Juan D. Gutiérrez[2]

[1] Universidad Internacional de La Rioja (Spain)
[2] University of Extremadura (Spain)

Received 8 February 2021 | Accepted 23 July 2021 | Published 2 August 2021

unir
LA UNIVERSIDAD
EN INTERNET

Abstract

There are a large number of machine learning algorithms as well as a wide range of libraries and services that allow one to create predictive models. With machine learning and artificial intelligence playing a major role in dealing with engineering problems, practising engineers often come to the machine learning field so overwhelmed with the multitude of possibilities that they find themselves needing to address difficulties before actually starting on carrying out any work. Datasets have intrinsic properties that make it hard to select the algorithm that is best suited to some specific objective, and the ever-increasing number of providers together make this selection even harder. These were the reasons underlying the design of CompareML, an approach to supporting the evaluation and comparison of machine learning libraries and services without deep machine learning knowledge. CompareML makes it easy to compare the performance of different models by using well-known classification and regression algorithms already made available by some of the most widely used providers. It facilitates the practical application of methods and techniques of artificial intelligence that let a practising engineer decide whether they might be used to resolve hitherto intractable problems. Thus, researchers and engineering practitioners can uncover the potential of their datasets for the inference of new knowledge by selecting the most appropriate machine learning algorithm and determining the provider best suited to their data.

Keywords

Classification, Decision Support System, Knowledge Elicitation, Machine Learning, Regression, Software.

DOI: 10.9781/ijimai.2021.08.001

I. Introduction

The evolution of computing capacities and the democratization of access to cloud computing at reasonable and affordable prices have fostered the appearance of a great number of machine learning applications for different fields. The engineering field in particular is not unaware of this trend. Engineering researchers and practitioners in a variety of areas have been drawn into the use of machine learning with the aim of inferring knowledge from their data. In particular, there are many engineering problems that can be solved using machine learning and artificial intelligence, and this, together with the Internet of Things and Big Data, inter alia, is one of the foundations of the fourth industrial revolution.

In most cases, the lack of knowledge about machine learning hinders engineers' and researchers' ability to analyse the potential that their data may have. Even with such knowledge, they will have to choose to use any from among an ever increasing number of alternatives with respect to algorithms, frameworks, tools, and providers. Given this context, it is hard to properly assess which will be the best way to address a specific engineering problem involving some particular data.

Moreover, the problem is compounded because, although well-known machine learning algorithms (such as Linear Regression, Decision Tree, Logistic Regression, Support Vector Machine, or Naive Bayes) are supposed to perform exactly the same, their implementations in third-party libraries and services can yield different results depending on the particular dataset.

Thus, any thorough evaluation of the different algorithms and their different implementations will be a time-consuming task. This ultimately results in most practitioner engineers and researchers in the field usually relying on their own experience to select a particular platform and algorithm, so that they will probably miss others that might well better reveal the whole potential of their datasets.

In this context, given this variety of options, and before taking into account such other factors as affordability, availability, complexity, or expertise, a preliminary study to find the optimal choice for a particular dataset might be helpful even for engineers, researchers, or scientists with no great depth of machine learning knowledge. While there are general-purpose machine learning tools that allow different algorithms to be applied to the same dataset, to the best of our knowledge, there is no solution available that is able to quickly and easily compare algorithm implementations from different providers.

All of this led us to posit the following research questions:

- RQ1: *Has a dataset the potential, i.e., is it possible to infer knowledge from it, to be used in a real-world engineering application or to solve a real-world engineering problem?*

* Corresponding author.

E-mail address: antoniojesus.fernandez@unir.net

TABLE I. Software Comparison

Software	Regression	Classification	Clustering	Deep Learning	Comparison	Visualization	Multiple Providers	Knowledge
Weka	✓	✓	✓	✓	✓			✓
Orange	✓	✓				✓		✓
RapidMiner	✓	✓	✓	✓		✓		✓
Knime	✓	✓	✓			✓		✓
CompareML	✓	✓			✓		✓	

(Regression: Support for Regression algorithms, Classification: Support for Classification algorithms, Clustering: Support for Classification algorithms, Deep Learning: Support for Deep Learning algorithms, Comparison: Model Comparison, Visualization: Data Visualization, Multiple Providers: Implementation of algorithms from Multiple Providers, Knowledge: Machine Learning Knowledge required.)

- RQ2: *Is it possible to determine, a priori, which are the best algorithms to reveal the potential of a dataset and construct the most appropriate predictive model from it?*
- RQ3: *Is it possible to select from among the wide range of libraries and services that allow the creation of predictive models those which are, a priori, the best to work with a specific dataset and problem?*

With the aim of answering these questions, we present *CompareML*, an approach that allows practitioner or research engineers to make a preliminary analysis of their data by testing different machine learning algorithms from different providers. The main goal of this approach is to allow such users to select the most suitable environment for their datasets without requiring any in-depth knowledge about machine learning. Thus, using *CompareML*, engineers can quickly and effortlessly compare the performance of different algorithms and providers applied to their specific datasets before deciding which to employ in their machine learning models.

The resulting software supporting the *CompareML* approach is licensed under an MIT permissive free software licence for it to be useful, reusable, and customizable in machine learning research areas. It is readily accessible at the following URL: https://compareml.io/. The software is already very intuitive, but nevertheless there is a User Manual[1] available for other researchers to answer any doubts they may have when using *CompareML*. Finally, scripts are available to enable full automated portability of the software.

The rest of this paper is organized as follows. Section II summarizes other approaches, and lists the algorithms and providers supported by *CompareML*. Section III describes the approach. Section IV details the implementation of the approach and the software architecture. Section V provides an illustrative example using *CompareML*. Finally, Section VI draws some conclusions and describes future work.

II. Related Work and Background

This section reviews some related work, describes the fundamentals of the machine learning algorithms compared by *CompareML*, and presents the state-of-the-art of the libraries and services available on the market, describing those supported by *CompareML*.

A. Related Work

There are general-purpose machine learning tools that have dealt with similar problems:

- Weka [1], a machine learning software package ideally suited for teaching and research which allows, inter alia, the comparison of algorithms for a dataset. It has recently been enriched with WekaLearning4j [2], a deep-learning package based on Deeplearning4j. The software is free under GNU GPL 3 for non-commercial purposes.
- Orange [3], a machine learning software package that allows data analysis and data visualization to be performed on datasets. It can be applied by means of Python scripts or through a visual programming interface in which users can make use of its functionalities. The software is developed by the University of Ljubljana, and is open-source released under a GPL licence.
- RapidMiner [4], [5], a data science software platform that implements data mining algorithms as operators that users can visually drag and drop to create customized dataflows. It is proprietary software developed by the RapidMiner firm, although previous versions were open source.
- Knime [4], a data mining tool integrated in Eclipse (the Java Integrated Development Environment) with a visual interface in which users make use of blocks, connecting them to create dataflows that visualize, deploy, and manage machine learning models. It is open source software developed by the University of Konstanz.

Table I presents a comparison of the work mentioned in this section in such aspects as:

a) Support for Regression algorithms;
b) Support for Classification algorithms;
c) Support for Clustering algorithms;
d) Support for Deep Learning algorithms;
e) Model Comparison;
f) Data Visualization;
g) Implementation of Multiple Providers' algorithms;
h) Machine Learning Knowledge required.

A more detailed and specific comparison of these software packages and the Scikit-Learn library and R Programming Language can be found in [6], where the authors compare software and services that are beyond the scope of this present work.

Our approach follows AutoML or Automated Machine Learning principles. AutoML [7] is an idea that consists of automating the entire pipeline or a part of a machine learning project. This is a hot topic of interest in both industry and academia, and the evaluation of its results [8] has led to several AutoML approaches and tools emerging. Some of those most widely used are:

- Automated Machine Learning in PowerBI [9] was launched by Microsoft in 2019 to allow business analysts, economists, and PowerBI users in general to build machine learning models to solve business problems without their needing to have a strong background in machine learning.
- PyCaret [10] is an open-source machine learning library in Python developed by Moez Ali and launched in 2019. Its main characteristic is its low-code orientation, making it simple and easy to use.
- Cloud AutoML [11] is the AutoML platform launched by Google in 2018 that trains custom machine learning models for image, video, and tabular data, as well as making use of pretrained models to create natural language processing, image classification, video recognition, or structured data discovery applications.

[1] User Manual available at http://shorturl.at/yCLX4

In addition to the software analysed, one can find in the literature related work worthy of mention with several tools and libraries, such as:

- LEAC [12], an efficient library for clustering with evolutionary algorithms in the neural networks field.
- Ruta [13], which implements autoencoders, neural networks that perform feature learning on data.
- AutoML-Zero [14], a specific-purpose tool that simultaneously searches for all aspects of a machine learning algorithm, using basic maths operations, with the objective of reducing human bias in the search space.

However, as noted above, to the best of our knowledge, there is no solution that is able to compare algorithm implementations from different providers while requiring no depth of machine learning knowledge on the part of the user.

B. Background

CompareML provides researchers with the possibility of creating models of their data using the following well-known algorithms:

- Linear Regression. This assumes a linear relationship between the input variables and a single output variable. The model learns by estimating the values of the coefficients used in the representation from the data available. The linear regression can be expressed as $y = ax + b$, where a and b are the aforementioned coefficients.
- Decision Tree. Decision tree algorithms build a tree-like structure in which each node represents a question concerning an attribute. The responses to that question create new branches, expanding the structure until the end of the tree is reached, with the leaf node being the one that indicates the predicted class.
- Boosted Decision Tree. This is a general method, not limited to decision trees, which consists of applying a boosting method to combine many classifiers into a new and stabler one with a smaller error. In the boosting, the predictors are made sequentially rather than independently, applying the rationale that the subsequent predictors learn from the mistakes of the previous ones.
- Random Forest. This algorithm is an improvement that creates several decision trees, using bagging or some other technique, and votes for the most popular output that the trees yield. Usually, most implementations do not count the outputs directly, but sum their normalized frequencies to get the label with greatest likelihood.
- Logistic Regression. This algorithm uses a more complex cost function – the 'sigmoid' or 'logistic' function – than the Linear Regression model. Input values are combined linearly using weights or coefficients to predict an output value. A key difference with Linear Regression is that the output being modeled is dichotomous (a 0 or 1) rather than numerical.
- Support-Vector Machine. The data are mapped onto a high-dimensional feature space so that the data points can be categorized even when the data are not otherwise linearly separable. Then, a separator is estimated for the data. The data should be transformed in such a way that a separator can be drawn as a hyperplane. As there are many possible hyperplanes, the Support Vector Machine algorithm finds a hyperplane that represents the largest separation, or margin, between classes.

Of particular importance are the libraries and services from different machine learning providers that *CompareML* supports:

- Turi Create [15], a Python™ [16] package that allows programmers to perform end-to-end large-scale data analysis and data product development. It is a distributed computation framework written in C++, developed at Carnegie Mellon University, and acquired in 2016 by Apple Inc.
- Scikit-Learn [17], one of the most popular machine learning libraries. It is largely written in Python with some core algorithms written in Cython to improve performance. It is supported by several institutional and private grants.
- R [18], a programming language that is an environment for statistical computing software written in C, Fortran, and R itself. It is widely used in machine learning tasks. It is developed by the *R Core Team. CompareML* runs R code embedded in Python through the access provided by the *rpy2* library.

III. Approach

A. The Approach in a Nutshell

The approach consists of a software architecture that supports machine learning experiments being carried out very easily, even for practitioners with no in-depth knowledge or skills in machine learning and artificial intelligence. Specifically, the approach allows classification and regression models to be implemented, and evaluates them so that their performance can be very easily compared using metrics that are in line with the models' types. Additionally, none of the machine learning algorithms are constrained to a single library or provider. Instead, several implementations of the algorithms from different libraries, tools, or providers can be built and evaluated, so that, for every scenario (i.e., dataset and problem to solve), it is possible to select the most suitable algorithm and the most appropriate implementation library or provider.

B. Data Analysis Decision Making

After the foregoing description of the main concepts of the approach's architecture, this section will present a discussion of the proposed pipeline on which *CompareML* is built. Engineers and practitioners will follow this pipeline (Fig. 1) to decide whether there is hidden knowledge within their data, and, if so, which are the best algorithms and providers for them to use to build accurate models.

The circumstances begin when engineers are faced with trying to solve some problem, and they think that machine learning presumably can help in looking for a solution. After collecting the data, reasonable doubt arises about the data's suitability and the possible hidden knowledge they contain so that the problem can be solved properly.

In the interest of clarifying the data's potential, researchers or engineers upload their data to an implementation of the *CompareML* approach, indicating the attribute to predict (the label). They then choose the machine learning libraries and services they can work with, and select those machine learning algorithms they want to use to create predictive models. In response, after evaluating and comparing the models generated, *CompareML* provides performance metrics about the models. It is quite common to do that as can be seen in work such as [19]–[22] where authors create several models to find out the one that gets better results. By analysing these performance metrics, engineers can decide whether their data has the potential to build accurate models with which to solve their problem, and, if so, the algorithm and provider combination most likely to produce a reliable solution.

C. Inputs and Outputs

There are certain classes of data that need to be sent to *CompareML*. These input data are listed in Table II.

In version 1.0, the validation software does not automatically recognize the selected label's data type, so that the experiments will be run regardless of the appropriateness of the label data type and the type of algorithms selected. In such cases, the experiments' results will be given according to how each one of the libraries, services, and tools selected handles these situations.

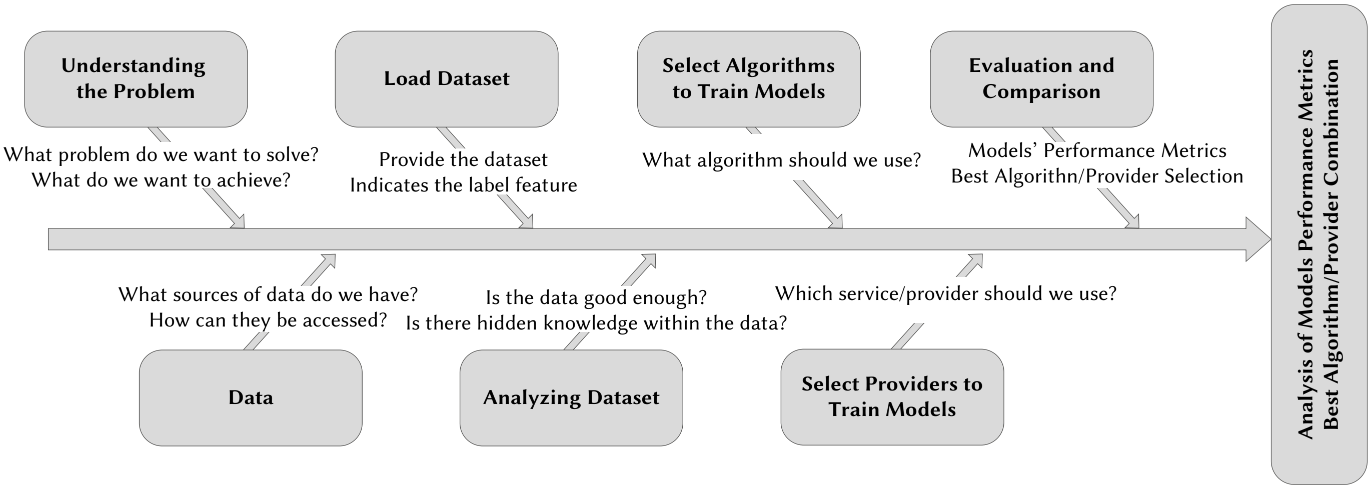

Understanding the Problem
What problem do we want to solve?
What do we want to achieve?
Load Dataset
Provide the dataset
Indicates the label feature
Select Algorithms to Train Models
What algorithm should we use?
Evaluation and Comparison
Models' Performance Metrics
Best Algorithm/Provider Selection
What sources of data do we have?
How can they be accessed?
Data
Is the data good enough?
Is there hidden knowledge within the data?
Analyzing Dataset
Which service/provider should we use?
Select Providers to Train Models
Analysis of Models Performance Metrics
Best Algorithm/Provider Combination

Fig. 1. Pipeline of the approach.

TABLE II. CompareML Input Data

Input Name	Description	Observations
`Dataset`	The dataset must be in CSV (comma-separated values) format, and it must contain a header row.	Notice that, since the comma is used as a separator, *CompareML* cannot handle field data containing commas or embedded line breaks. Also, it may not handle other unconventional characters.
`Label`	Feature that models will predict.	After uploading a dataset, users must select the label from among the dataset's existing features.
`Providers`	Machine learning libraries and services available to build models.	Users must select at least one of them. The validation software of the approach supports Turi Create, Scikit- Learn, and R.
`Algorithms`	Regression and classification algorithms available to build models.	Users must select at least one of them. The validation software of the approach allows users to choose between the Linear, Decision Tree, and Boosted Decision Tree regression algorithms, and the Random Forest, Logistic Regression, and Support Vector Machine classification algorithms.

The outputs of *CompareML* are a set of metrices that allow the appropriateness of the models created to be studied using different algorithms from different machine learning tools and services. Those outputs vary depending on the user's algorithm selection.

The outputs for *Regression* algorithms are:

- **RMSE**. This is a measure of the differences between the values predicted by a model and the observed values.

  RMSE can be defined as:

  $$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{n}\left(\hat{y}_i - y_i\right)^2} \tag{1}$$

  where

  - N is the number of instances;
  - $\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n$ are the values predicted by the model; and
  - $y_1, y_2, \ldots, y_n$ are the observed values
- **R²**. Also known as the Coefficient of Determination, this is a metric that helps to explain the relationship between two variables. It ranges from 0 to 1, with the closer to 1, the better the model being analysed. Although a useful metric, it should be noted that, as more independent variables are added, R^2 will always rise.

  R^2 can be defined as:

  $$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{2}$$

  where

  - N is the number of instances,
  - $\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n$ are the values predicted by the model,
  - $y_1, y_2, \ldots, y_n$ are the observed values, and
  - $\bar{y}$ is the mean of the n observed values.
- **Max-Error**. The Max-Error metric is the worst case error between a predicted value and a true value.

  Max-Error can be defined as:

  $$Max\text{-}Error(y, \hat{y}) = \max\left(|y_i - \hat{y}_i|\right) \tag{3}$$

  where

  - $\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n$ are the values predicted by the model, and
  - $y_1, y_2, \ldots, y_n$ are the observed values.
- **Raw Data**. Information yielded directly by the provider.

The outputs for *Classification* algorithms are:

- **Accuracy**. An accuracy value, indicating the correctly predicted instances, is given for each algorithm selected. It can be defined as:

  $$Accuracy = \frac{Number\ of\ correctly\ predicted\ observations}{Total\ number\ of\ observations} \tag{4}$$
- **Confusion Matrix**. A table layout where each row represents the number of instances of each class and each column represents the class that has been predicted by the model. A confusion matrix is created for each algorithm selected [23].
- **Precision**. This indicates the proportion of predicted positives. A precision value is given for each algorithm selected. Using the confusion matrix, it can be calculated as:

  $$Precision = \frac{TP}{TP + FP} \tag{5}$$

  where

  - TP = true positives,
  - FP = false positives.

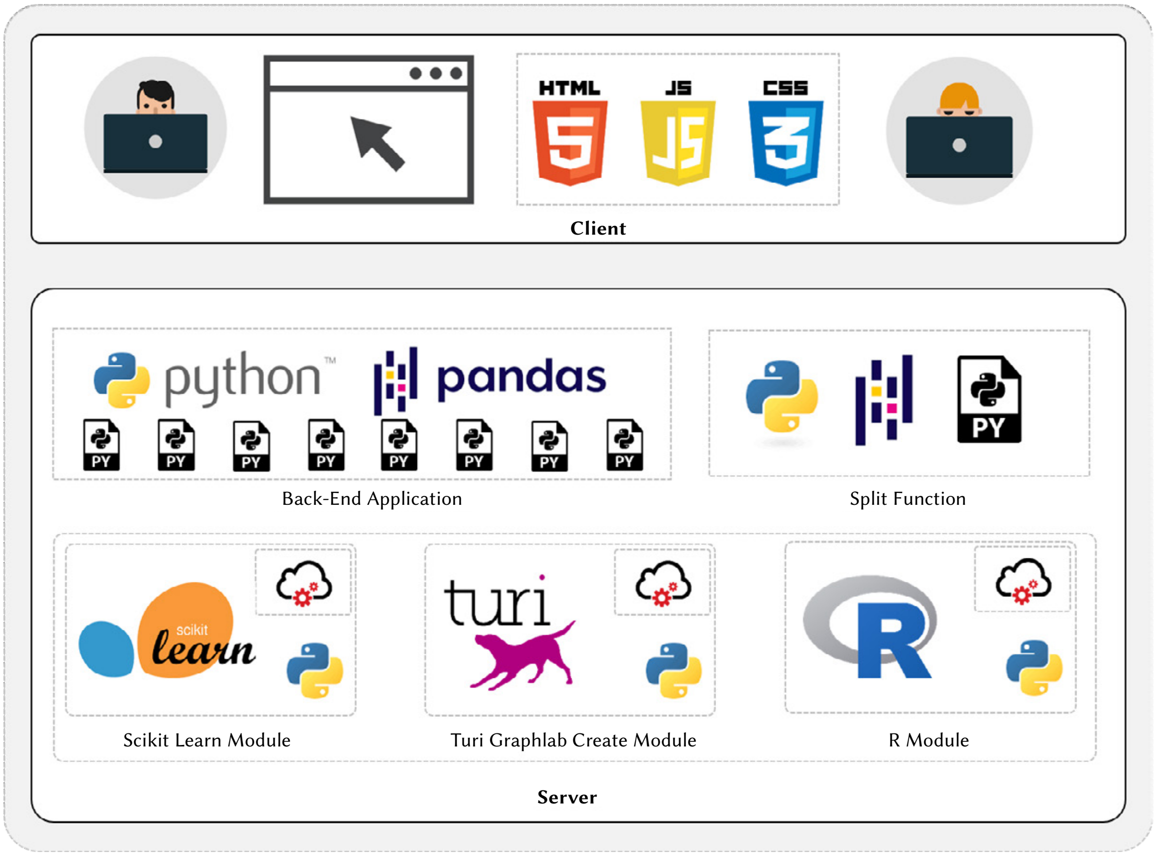


Fig. 2. CompareML Software Architecture.

- **Recall (Sensitivity)**. This indicates the proportion of positives predicted as positives. A recall value is given for each algorithm selected. Using the confusion matrix, it can be calculated as: (number of true positives / (number of true positives + number of false negatives).

  $$Recall = \frac{TP}{TP + FN} \tag{6}$$

  where

  - TP = true positives,
  - FP = false positives.
- **Raw Data**. Information yielded directly by the provider.

IV. Implementation

A. Software Architecture

CompareML has been implemented following a classical client-server software architecture with an MVC (Model View Controller) approach [24] – a suitable option for this solution.

The server hosts the resources that manage the creation of models and deliver the results to the client. The client interacts with researchers through its user interface, and initiates requests to the server. This architecture is illustrated graphically in Fig. 2, and consists of the following modules:

- **Client**. This module handles the presentation layer, and is responsible for interacting with users. In it, users upload the dataset and set the configuration of the experiments that are sent to the server side. When the experiments are carried out, the results are sent back to this module to be shown to users in a friendly manner. The user interface has been designed to maximize usability, being simple, consistent, and offering cross-browser compatibility. This module was developed using widely used technologies such as HTML5, CSS3, and JavaScript.
- **Back-End Main Application**. This is the core of *CompareML*. It is responsible for coordinating and controlling the software's operational processes. It receives from the user interface the conditions under which the experiments must be carried out, and calls the Turi Create, Scikit-Learn, and R modules required, sending them the conditions of the experiments that affect them (algorithm selection, training dataset, test dataset, ...). When the execution of those modules ends, it receives the results and sends them back to the user interface. This module was developed in Python.
- **Back-End Split Function**. The split function is a special module in the back end that deals with the problem of splitting the dataset uploaded by researchers into two subsets: the training dataset containing 80% of the instances of the total dataset, and the test dataset containing the remaining 20% of the instances. This task is carried out in this module because it is necessary to ensure that the experiment's results are as objective as possible. If each provider module were to divide the dataset itself randomly, the random seeds would be different, and this would have potentially negative implications for any objective comparison of the models created through each of those modules. The split function module was developed using Python and Pandas [25], a powerful open source data analysis and manipulation tool built on top of the Python programming language.
- **Provider Modules**. These contain the implementations of the Random Forest, Logistic Regression, and Support Vector Machine classification algorithms and the Linear, Decision Tree, and Boosted Decision Tree regression algorithms using Scikit-Learn. It makes use of the *sklearn* library to build and evaluate the models

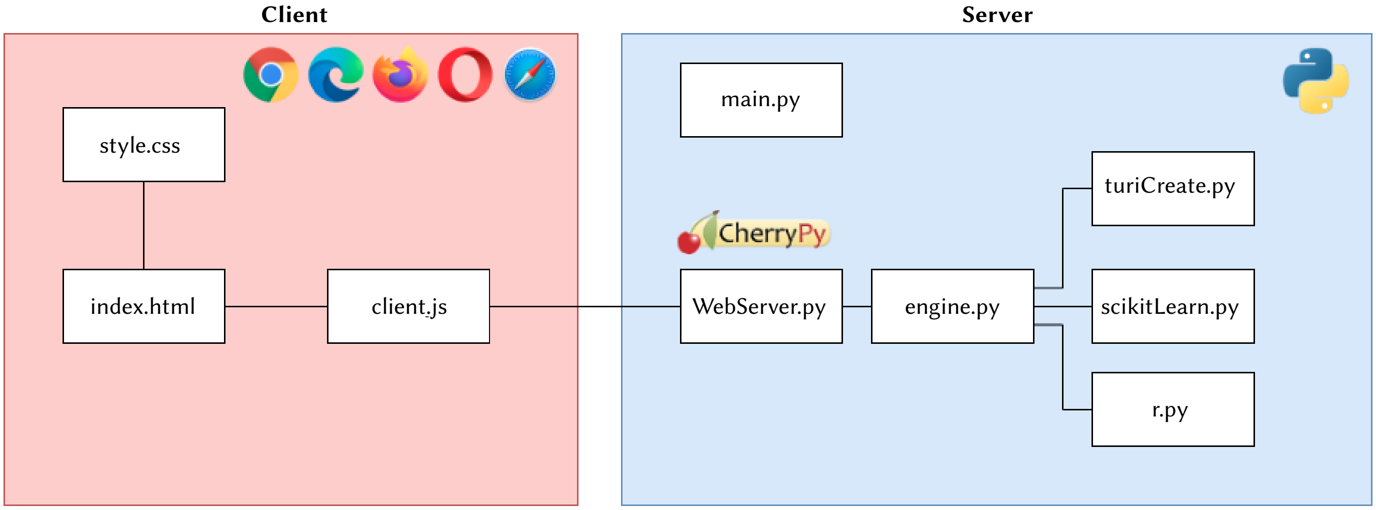


Fig. 3. *CompareML* file structure.

and the *pandas* library to manipulate data using its *DataFrame* data structure and functions. These modules receive as inputs the training and evaluation datasets, the algorithms that need to be used to build models, and the providers. When the experiments are carried out, the results are sent to the Back-End Main Application module. The functionalities of the machine learning providers, i.e., the experiments carried out using each provider's libraries, data structures, and functions, are isolated within the server side so as to facilitate their development.

B. File Structure and Software Business Process

The file structure is illustrated graphically in Fig. 3. The following is a description of the server files:

- `main.py`. This file contains the Web server configuration, defines the resources shown, and links with WebServer.py. It is implemented within the CherryPy framework [26].
- `WebServer.py`. This file receives the input from the client side, and processes it for subsequent handling.
- `engine.py`. This file is the core of *CompareML*. It contains the "split function" module which divides the dataset into training and evaluation, and communicates with the provider modules that need to be called to fulfill the requirements of the experiments defined by the researchers.
- `turiCreate.py`, `scikitLearn.py` and `R.py`. These files create the Turi Create, Scikit-Learn, and R models, respectively.

Fig. 4 is a BPMN (Business Process Model and Notation) diagram of the business processes of the modules in *CompareML* aimed at better illustrating the module relationships and the software framework.

C. Reproducibility and Collaboration

The guidelines on which the approach is based have been described in depth in the preceding subsections. However, in order to facilitate replication of the work and to encourage its implementation, in this subsection additional material is provided so that engineers can easily make use of this approach to perform preliminary data analysis.

The source code of an implementation of the approach is freely available in a GitHub repository under an MIT permissive free software licence[2]. A Developer Manual and a User Manual are provided. The Developer Manual, integrated into the Readme.md[3] file of the GitHub repository, describes clearly how the software is structured and how to make use of the code. It is possible to find Vagrant and Ansible scripts that simplify the software's portability, configuration, and deployment by any party interested in employing their own infrastructure. The User Manual[4] helps engineers without any in-depth knowledge of machine learning or data science to make use of the approach and to get answers to any doubts they may have about how it operates. Moreover, as was noted above, an implementation of the approach has been deployed on the Web[5] so that engineers and researchers can use it without the need to configure it themselves.

Table III provides information relevant for the deployment of software based on the proposed approach.

TABLE III. Information for the Deployment of Software Based on the Proposed Approach

Executable	https://compareml.io/
Licence	MIT licence (MIT)
Platforms	Windows, Linux, MacOS
Installation requirements	The approach can be deployed as a Web application. It is not necessary to install additional software, just a Web browser.
User manual	https://raw.githubusercontent.com/i3uex/CompareML/master/CompareML%20User%20Manual.pdf
Developer manual	https://github.com/i3uex/CompareML/blob/master/README.md
Software used	CherryPy, Python
Compilation requirements	Python dependencies: CherryPy, pandas, 14 sklearn, tensorflow, turicreate. R dependencies: 15 optparse, testthat, ggplot2, randomForest, caret, 16 e1071.

However, it might be useful to illustrate how potential research users could, in their *CompareML* solution, add new algorithms to those already implemented. The *CompareML* software architecture has a simple and elegant design to encourage potential research users to aid in the growth or development of the solution by adding new algorithms or providers to those already implemented. In order to do so, researchers or developers just have to access the files *scikit_learn.py*, *turi_create.py*, or the *R* folders under the providers directory to add the implementation of an algorithm. It is possible to create a new file in that directory to include a new provider. Once the implementations of the new algorithms are done, the *engine.py* file must be updated with

[2] Link to the Source Code: https://github.com/i3uex/*CompareML*/

[3] Link to the Developer Manual: https://github.com/i3uex/*CompareML*/ blob/master/README.md

[4] Link to the User Manual: http://shorturl.at/yCLX4

[5] *CompareML* deployed on the Web: https://compareml.io/

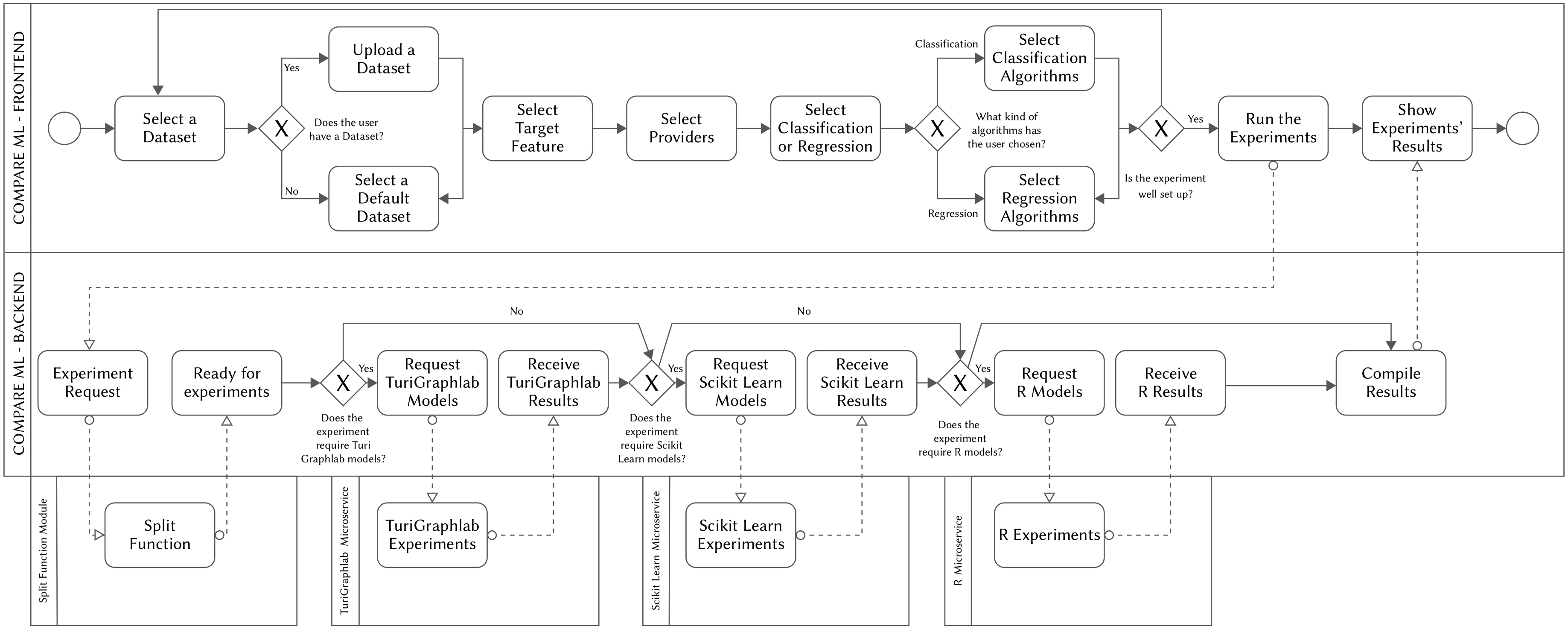

COMPARE ML - FRONTEND
Select a Dataset
Does the user have a Dataset?
Yes
No
Upload a Dataset
Select a Default Dataset
Select Target Feature
Select Providers
Select Classification or Regression
What kind of algorithms has the user chosen?
Classification
Regression
Select Classification Algorithms
Select Regression Algorithms
Is the experiment well set up?
Yes
Run the Experiments
Show Experiments' Results
COMPARE ML - BACKEND
Experiment Request
Ready for experiments
Does the experiment require Turi Graphlab models?
Yes
No
Request TuriGraphlab Models
Receive TuriGraphlab Results
Does the experiment require Scikit Learn models?
Yes
No
Request Scikit Learn Models
Receive Scikit Learn Results
Does the experiment require R models?
Yes
Request R Models
Receive R Results
Compile Results
Split Function Module
Split Function
TuriGraphlab Microservice
TuriGraphlab Experiments
Scikit Learn Microservice
Scikit Learn Experiments
R Microservice
R Experiments

Fig. 4. *CompareML* BPMN diagram.

the list of algorithms and providers as shown in Listing 1. Lastly, the call of the newly implemented methods must be done in the same file.

Listing 1. Code from *engine.py* showing the providers and algorithms currently supported in *CompareML*.

```
PROVIDERS   =   {
   c.TURI_CREATE:   turi_create,
   c.SCIKIT_LEARN:   scikitLearn ,
   c.R:   r,
}

ALGORITHMS   =   {
   'classification':
        [ c.RANDOM_FOREST,
        c.LOGISTIC_REGRESSION,
        c.SUPPORT_VECTOR_MACHINES
   ],
   'regression':
        [ c.LINEAR_REGRESSION,
        c.BOOSTED_ DECISION_TREES,
        c.DECISION_TREE
   ]
}
```

V. Illustrative Example & Results

In this section, as illustrative examples, we make use of *CompareML* to solve two problems – a classification problem and a regression problem. These case studies are good representations of the types of problem for which the approach is useful. The first case study focuses on training classification models, and provides the output metrics of each trained algorithm to allow their evaluation by users. The second case study focuses on training regression models to the same end.

The datasets in the two case studies are in a tabular CSV format, as is supported by*CompareML*. It is important to stress that the current implementation does not allow data preparation operations or the application of feature engineering techniques, so data transformation must be done before loading the dataset into the *CompareML* implementation. Nevertheless, it is easy to rerun the experiments after performing such data transformation operations, and then compare the differences.

A. Classification Example

In this example, the aim is to find the provider which, a priori, is best suited to dealing with the CarEvaluation dataset [27] obtained from the UCI Machine Learning Repository [28]. This dataset comprises labeled data obtained from 1728 cars where the goal is to evaluate car conditions based on certain characteristics related to price and comfort.

The dataset consists of 1726 instances (observations) together with 7 features (variables) which are detailed in Table IV. In contrast with the original dataset, where there are 4 classes of the label feature Acceptability {unacceptable, acceptable, good, verygood}, just 2 categories have been adopted {yes, no} so that we can make use of two-class classification algorithms.

TABLE IV. Set of Features of the CarEvaluation Dataset

Feature Name	Description	Type
`Price`	Buying price	Categorical
`Maintenance`	Price of maintenance	Categorical
`Doors`	Number of doors	Numerical
`Seats`	Capacity in terms of persons to carry	Numerical
`Boot`	Size of luggage boot	Categorical
`Safety`	Estimated safety of the car	Categorical
`Acceptability`	Car acceptability	Categorical (*Label*)

The steps that need to be followed to carry out the experiment in this implementation of the *CompareML* approach are:

1. Provide the dataset to *CompareML*. To this end, the option of selecting a default dataset is ignored, and the CarEvaluation dataset is directly uploaded from our computer.
2. Select the label feature that we are interested in predicting. After uploading the dataset, the drop-down menu of this section is filled with the names of all the variables. In our case, we select the Acceptability variable, which indicates the level of acceptability of a car according to its characteristics, i.e., the field we wish to predict.
3. Choose the machine learning libraries and services in which we want to run the experiment (Scikit-Learn, Turi Create, and R). At least one of them must be selected. In our example, we are interested in comparing all three providers.
4. Select whether we are going to carry out a *Regression* or a *Classification* experiment. Depending on the type of algorithm selected, we can choose from a variety of algorithms. We are interested in predicting a categorical value, and, for that reason, we mark the *Classification* algorithms checkbox. After selecting this option, we must select at least one algorithm. In our example, we want to perform an experiment including all the algorithms available (Random Forest, Logistic Regression, and Support Vector Machine).

If the experiment is set up properly, we can run it by pressing the "Start" button. If not, an error message will be shown describing how to resolve the problem.

When the experiment has been performed, *CompareML* shows the results below the "Start" button. Fig. 5 is a screenshot of the *CompareML* user interface and the output produced after the classification experiment has been carried out. The results of the experiment are

TABLE V. Results of the Classification Case Study Experiment

		Turi Create	Scikit Learn	R
Ramdom Forest	**Accuracy**	0.9480	0.9220	0.9855
	Precision	0.9091	0.4610	0.9876
	Recall	0.3704	0.5000	0.9969
Logistic Regression	**Accuracy**	0.9855	0.9769	0.9884
	Precision	0.8667	0.9322	0.9968
	Recall	0.9630	0.9027	0.9906
Support Vector Machine	**Accuracy**	0.9855	0.9769	0.9827
	Precision	0.8667	0.9197	0.9906
	Recall	0.9630	0.9197	0.9906

presented in Table V. They include the *Accuracy*, *Precision*, and *Recall* evaluation metrics. *CompareML* also provides the *Confusion Matrix* and raw data with information yielded directly by the provider.

As expected, building a model using the same algorithm and the same data produces similar but slightly different results. This implies that, even when the algorithms are the same, their implementations by different providers impacts the performance of the models built using them. Although in most cases those differences are small, in some cases they are significant. Let us focus for example on the *Accuracy* metric. In this case study, the accuracy of the *Logistic Regression* model with *R* is 6.64 percentage points greater than that of the *Random Forest* model with *ScikitLearn*, which is a considerable difference. In general however, the differences are smaller. With the exception of *Random Forest*, the differences between the models built with the same algorithms from different providers do not reach 1 percentage point. Nevertheless, that percentage point can make a difference.

Looking at the outcomes of the experiment, one can see that, especially with *TuriCreate* and *ScikitLearn*, the accuracy of the *Random Forest* models, regardless of the provider, is poorer than that of the *Logistic Regression* and *Support Vector Machine* models. This may be an indication that the models created with these algorithms are better suited to the CarEvaluation dataset. The greatest accuracy (98.84 percentage points) corresponds to the *Logistic Regression* model built with *R*.

The experimental results of this illustrative example can be reproduced using the CarEvaluation dataset which is preloaded in *CompareML*, by selecting acceptability as target feature (label).

B. Regression Example

In the regression example, we use the Heating dataset [29] obtained from the UCI machine learning Repository [28]. This contains data obtained from energy analyses applied to 12 different building shapes. The dataset consists of 768 instances (observations) together with 9 features (variables), which are detailed in Table VI. The label feature HeatingLoad is a continuous numerical feature that is a measure of the amount of heat energy that would need to be added to a space to maintain the temperature within an acceptable range. The remaining features of the dataset are {Relative Compactness, Surface Area, Wall Area, Roof Area, Overall Height, Orientation, Glazing Area, Glazing Area Distribution, Heating Load}, with this last being the label in our case study.

The steps needed to carry out the experiment are the same as in the previous experiment. The algorithms available for this kind of experiment are Linear Regression, Boosted Decision Tree, and Decision Tree.

Fig. 6 is a screenshot of the *CompareML* user interface and the output of the regression experiment. For readability, the results are listed in Table VII for the *RMSE* and *Max-error* evaluation metrics.

As was the case for the classification example, the experimental results differed slightly depending on the algorithms and providers used to create the model. Overall, the *Decision Tree* models gave uniformly good results, but the best result was with *Scikit-Learn* creating the model using the *Boosted Decision Tree* algorithm.

TABLE VI. Set of Features of the Heating Dataset

Feature Name	Type
`Relative Compactness`	Categorical
`Surface Area`	Numerical
`Wall Area`	Numerical
`Roof Area`	Numerical
`Overall Height`	Numerical
`Orientation`	Numerical
`Glazing Area`	Numerical
`Glazing Area Distribution`	Numerical
`Heating Load`	Numerical (*Label*)

Analysing these results, one observes that on occasions the same algorithm can yield the best and the worst results depending on the provider that implements and deploys the algorithms. Although it is not often the case, it is still of interest that, given a specific problem and a particular dataset, the choice of provider can still have a major impact. For this experiment in particular, the difference in RMSE between the best and the worst models is 5.7872, and these are with the same algorithm − *Boosted Decision Tree.*

For the *Linear Regression* algorithm, the evaluation metrics (for both RMSE and Max-Error) differ little from each other. This suggests that the simpler the algorithm, the greater the similarity of the outcomes.

The experimental results of this illustrative example can be reproduced using the Heating dataset which is pre-loaded in *CompareML*, by selecting HeatingLoad as target feature (label).

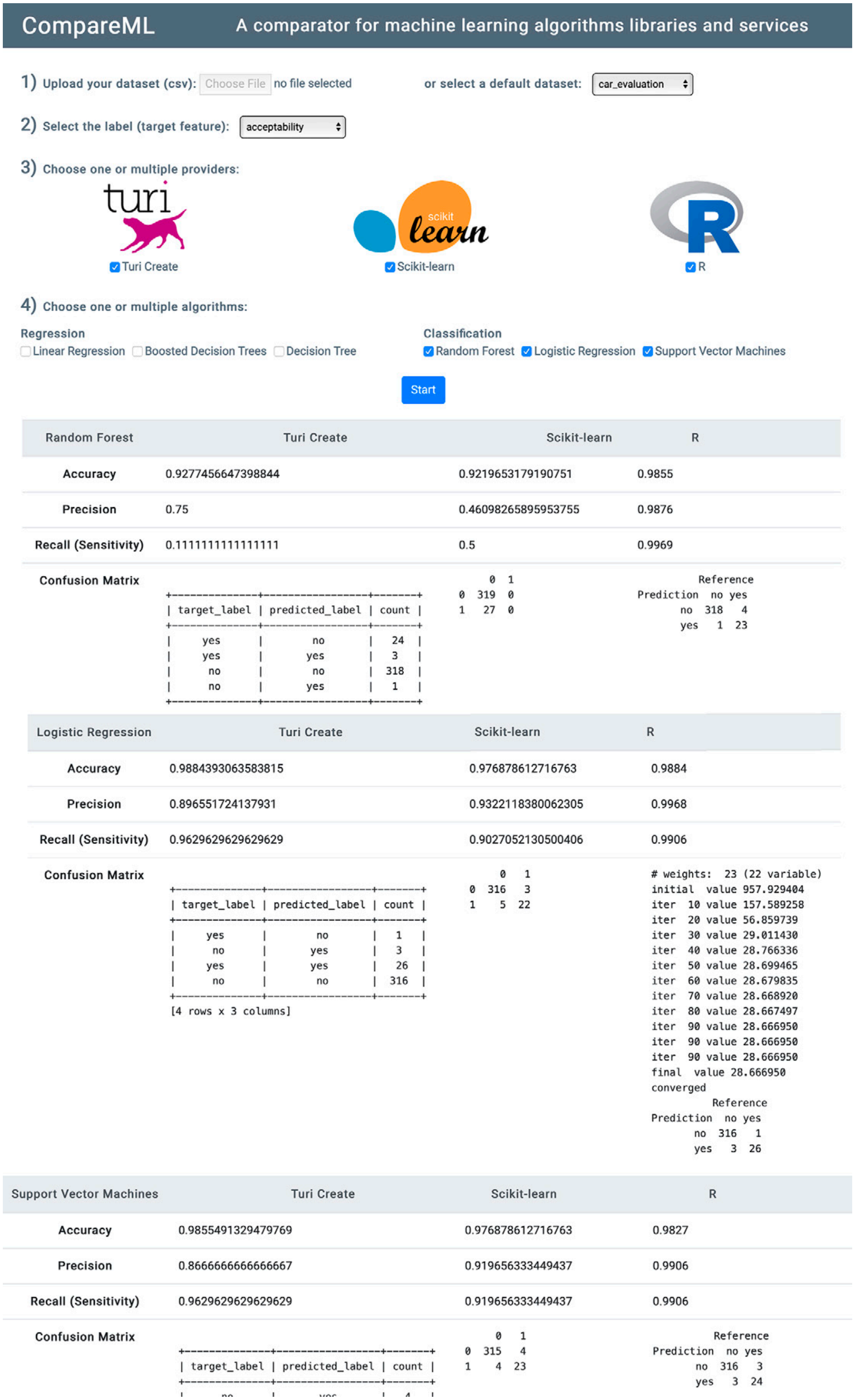
C. Application in Education

As can be seen with the illustrative case studies, *CompareML* is very easy to use and the results are provided straightforwardly. This makes it a potentially interesting tool for teaching in that it can facilitate students' interpretation of the results of applying these algorithms to a given problem with specific data. As examples of aspects in which the use of *CompareML* in teaching can generate meaningful knowledge for students, we would emphasize the following:

- One can see how certain algorithms perform better than others with certain data due to the nature of that data and the hidden knowledge it may contain. For instance, there are problems that can be solved with algorithms whose focus is on similarity, others that can be solved with tree-based algorithms which apply decisions based on the values of features, and others that can be modeled through refining.
- The effect that *feature engineering* techniques may have in transforming certain features according to different criteria can be analysed by monitoring the metrics resulting from each *CompareML* execution corresponding to each data transformation.
- One can study the impact of *feature selection* methods that apply *CompareML* to subsets of the original dataset, or, similarly, study the impact on the models' accuracy of adding new features to the dataset.

TABLE VII. Results of the Regression Case Study Experiment

		Turi Create	Scikit Learn	R
Linear Regression	**RMSE**	3.1837	3.1847	3.1778
	Max-Error	8.7520	8.8801	8.8788
Boosted Decision Tree	**RMSE**	6.2619	0.4747	2.1887
	Max-Error	13.2556	3.0332	6.3126
Decision Tree	**RMSE**	2.2976	1.6007	2.7292
	Max-Error	6.4962	4.3373	9.0515



CompareML   A comparator for machine learning algorithms libraries and services

1) Upload your dataset (csv): Choose File no file selected   or select a default dataset: car_evaluation

2) Select the label (target feature): acceptability

3) Choose one or multiple providers:

☑ Turi Create   ☑ Scikit-learn   ☑ R

4) Choose one or multiple algorithms:

Regression
□ Linear Regression □ Boosted Decision Trees □ Decision Tree

Classification
☑ Random Forest ☑ Logistic Regression ☑ Support Vector Machines

Random Forest	Turi Create	Scikit-learn	R
Accuracy	0.9277456647398844	0.9219653179190751	0.9855
Precision	0.75	0.46098265895953755	0.9876
Recall (Sensitivity)	0.1111111111111111	0.5	0.9969

Confusion Matrix

```
+--------------+-----------------+-------+
| target_label | predicted_label | count |
+--------------+-----------------+-------+
|     yes      |        no       |   24  |
|     yes      |       yes       |   3   |
|      no      |        no       |  318  |
|      no      |       yes       |   1   |
+--------------+-----------------+-------+
```

```
     0  1
0  319  0
1   27  0
```

```
          Reference
Prediction  no yes
       no  318   4
       yes   1  23
```

Logistic Regression	Turi Create	Scikit-learn	R
Accuracy	0.9884393063583815	0.976878612716763	0.9884
Precision	0.896551724137931	0.9322118380062305	0.9968
Recall (Sensitivity)	0.9629629629629629	0.9027052130500406	0.9906

Confusion Matrix

```
+--------------+-----------------+-------+
| target_label | predicted_label | count |
+--------------+-----------------+-------+
|     yes      |        no       |   1   |
|      no      |       yes       |   3   |
|     yes      |       yes       |   26  |
|      no      |        no       |  316  |
+--------------+-----------------+-------+
[4 rows x 3 columns]
```

```
     0   1
0  316   3
1    5  22
```

```
# weights:  23 (22 variable)
initial  value 957.929404
iter  10 value 157.589258
iter  20 value 56.859739
iter  30 value 29.011430
iter  40 value 28.766336
iter  50 value 28.699465
iter  60 value 28.679835
iter  70 value 28.668920
iter  80 value 28.667497
iter  90 value 28.666950
iter  90 value 28.666950
iter  90 value 28.666950
final  value 28.666950
converged
          Reference
Prediction  no yes
       no  316   1
       yes   3  26
```

Support Vector Machines	Turi Create	Scikit-learn	R
Accuracy	0.9855491329479769	0.976878612716763	0.9827
Precision	0.8666666666666667	0.919656333449437	0.9906
Recall (Sensitivity)	0.9629629629629629	0.919656333449437	0.9906

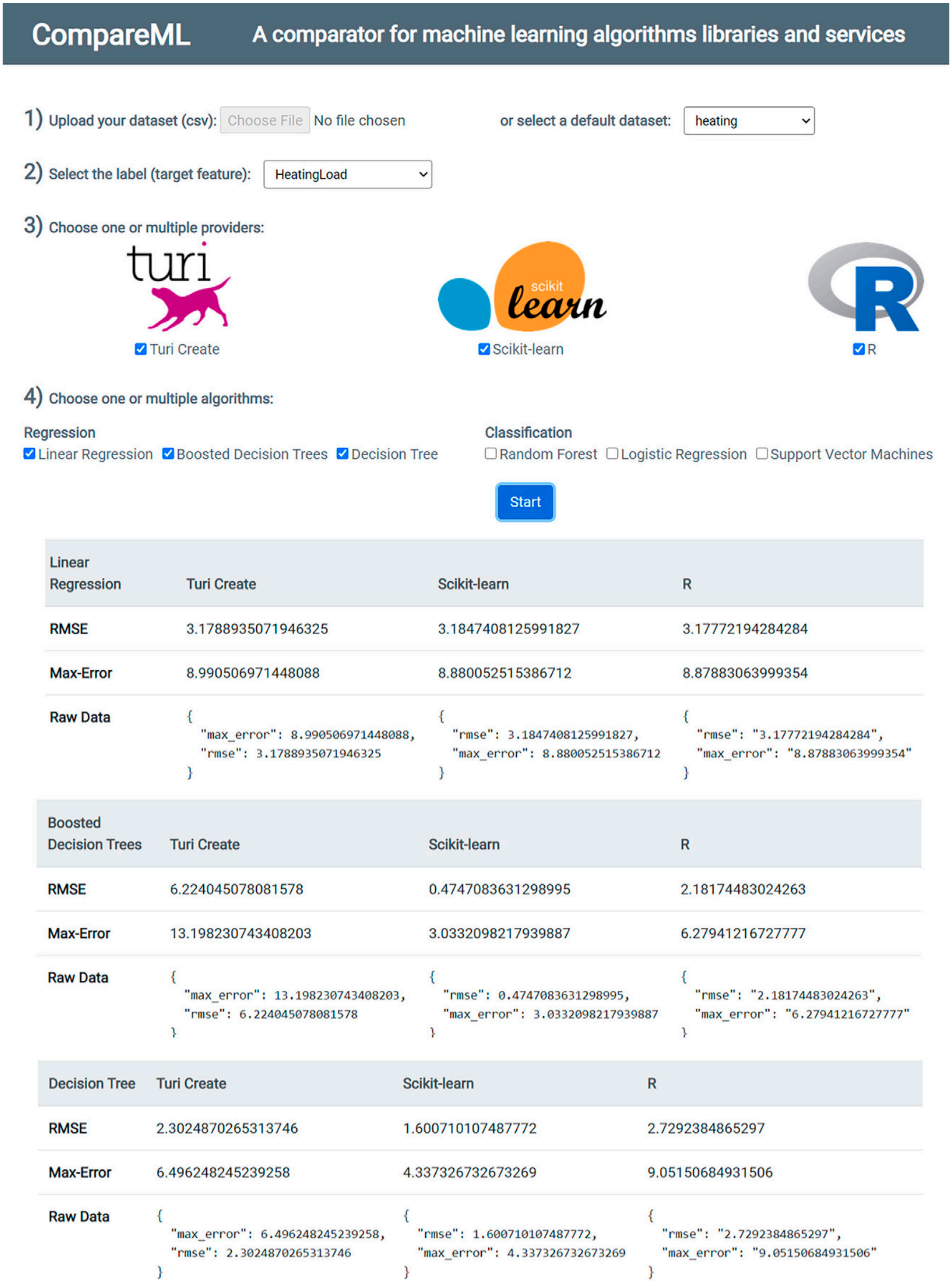
Confusion Matrix

```
+--------------+-----------------+-------+
| target_label | predicted_label | count |
+--------------+-----------------+-------+
|      no      |       yes       |   4   |
```

```
     0   1
0  315   4
1    4  23
```

```
          Reference
Prediction  no yes
       no  316   3
       yes   3  24
```

Fig. 5. *CompareML* User interface providing the output of the Classification Case Study.



CompareML A comparator for machine learning algorithms libraries and services

1) Upload your dataset (csv): Choose File No file chosen or select a default dataset: heating

2) Select the label (target feature): HeatingLoad

3) Choose one or multiple providers:

☑ Turi Create ☑ Scikit-learn ☑ R

4) Choose one or multiple algorithms:

Regression
☑ Linear Regression ☑ Boosted Decision Trees ☑ Decision Tree

Classification
☐ Random Forest ☐ Logistic Regression ☐ Support Vector Machines

Start

Linear Regression	Turi Create	Scikit-learn	R
RMSE	3.1788935071946325	3.1847408125991827	3.17772194284284
Max-Error	8.990506971448088	8.880052515386712	8.87883063999354

Raw Data	{
	"max_error": 8.990506971448088,
	"rmse": 3.1788935071946325
	}
		{
		"rmse": 3.1847408125991827,
		"max_error": 8.880052515386712
		}
			{
			"rmse": "3.17772194284284",
			"max_error": "8.87883063999354"
			}

Boosted Decision Trees	Turi Create	Scikit-learn	R
RMSE	6.224045078081578	0.4747083631298995	2.18174483024263
Max-Error	13.198230743408203	3.0332098217939887	6.27941216727777

Raw Data	{
	"max_error": 13.198230743408203,
	"rmse": 6.224045078081578
	}
		{
		"rmse": 0.4747083631298995,
		"max_error": 3.0332098217939887
		}
			{
			"rmse": "2.18174483024263",
			"max_error": "6.27941216727777"
			}

Decision Tree	Turi Create	Scikit-learn	R
RMSE	2.3024870265313746	1.600710107487772	2.7292384865297
Max-Error	6.496248245239258	4.337326732673269	9.05150684931506

Raw Data	{
	"max_error": 6.496248245239258,
	"rmse": 2.3024870265313746
	}
		{
		"rmse": 1.600710107487772,
		"max_error": 4.337326732673269
		}
			{
			"rmse": "2.7292384865297",
			"max_error": "9.05150684931506"
			}

Fig. 6. *CompareML* User interface providing the output of the Regression Case Study.

TABLE VIII. Comparison With Other Approaches

Weka	**Execution environment**	Run locally
	Difficulty	Medium - low
	Algorithms	Wide range of algorithms available. A single implementation per algorithm
	Classification metrics	Accuracy, confusion matrix, precision, recall, F1-score, ROC
	Regression metrics	MAE, RMSE, correlation coefficient
PyCaret	**Execution environment**	Commonly works in a Jupyter environment that can run locally or in the cloud
	Difficulty	Low
	Algorithms	Wide range of algorithms available. A single implementation per algorithm
	Classification metrics	Accuracy, AUC, precision, recall, F1-score, kappa, MCC
	Regression metrics	MAE, MSE, RMSE, R^2, RMSLE, MAPE
PowerBI	**Execution environment**	Runs in the cloud
	Difficulty	Medium. ML concepts are easy to follow, but Power Platform knowledge is required
	Algorithms	Wide range of algorithms available. A single implementation per algorithm
	Classification metrics	AUC, confusion matrix, precision, recall, cost-benefit analysis
	Regression metrics	Model performance explanation, Average Residual Error
CompareML	**Execution environment**	Runs in the cloud
	Difficulty	Low
	Algorithms	Three each for classification and regression. Three implementations per algorithm
	Classification metrics	Accuracy, confusion matrix, precision, recall
	Regression metrics	RMSE, R^2, Max-Error

D. Comparison With Other Approaches

In this subsection, a comparative evaluation is made of *CompareML* with the other approaches discussed in the Related Works section, especially those that, considering their specific features, admit a direct comparison with *CompareML*. These are Weka, PyCaret, and PowerBI AutoML.

Weka needs to be downloaded and installed to run locally. It offers a large number of functionalities. The user interface is built in Java, but, even though it is not hard to understand and use, it is not up-to-date, which is an obstacle to its use. There are a large number of algorithms to choose from, all of them with a single implementation. Experiments provide the key metrics with which to evaluate the models, and it has a CLI interface.

PyCaret is a machine learning library in Python commonly used in the Jupyter Notebooks Environment. It is simple and easy to use due to its low-code orientation. A large number of algorithms are automatically selected to perform experiments, and it is easy to tune the hyperparameters if necessary. The main classification and regression metrics are generated directly after training the models with just one line of code.

PowerBI incorporates the creation of machine learning models with a focus on their explicability. A little background working with PowerBI dataflows is required, but the AutoML process is straightforward and is finely integrated with the PowerBI online app. Data processing is straightforward in the environment, and *Feature Selection* techniques are applied before the models are trained. The main classification and regression metrics are attractively presented in the PowerBI reports.

CompareML differs from the other approaches in that no configuration is required, and it can be used directly. Also, the fact that more than one implementation is available for each algorithm constitutes its principal differentiating element. In particular, since there are three implementations for each algorithm, this allows users to analyse the differences between those implementations.

Table VIII summarizes this comparison in terms of the type of execution environment, of the difficulty in setting up and using the tool, of the metrics that are provided for an evaluation of regression and classification models, and of the number and type of algorithms supported.

VI. Conclusions

In this paper, we have presented a novel approach to supporting preliminary data analysis in the engineering field that enables engineers and researchers to quickly and easily analyse the potential for inferring knowledge that may lie hidden in their data. Similarly, it assists them in comparing machine learning models using different implementations from different providers of well-known algorithms, without their needing prior knowledge about how to create those models with each provider.

To that end, the most widely used machine learning libraries and services and some of the best-known classification and regression machine learning algorithms can be compared by performing a series of experiments. After the experiments have been completed and the models created, the commonest evaluation metrics are presented so that researchers and engineers can properly evaluate and compare the possibilities they have available, giving them all the information they need to make a decision on how to proceed with their work.

As far as we know, there has been no previous intelligent programming environment with the characteristics and objectives of *CompareML*, i.e., using machine learning techniques to construct software with a modular and scalable architecture, and aimed at providing practising engineers with a decision support system that can help them solve hitherto intractable problems by eliciting knowledge from their data, even though they have no in-depth machine learning skills.

From the experiments carried out with *CompareML*, affirmative answers can be given to the three research questions posited. With regard to RQ1, it is possible to analyse the potential for inferring knowledge hidden in a dataset obtained from a real-world engineering application. With regard to RQ2 and RQ3, it is possible to obtain, a priori, the best combination of algorithm and provider with which to construct a predictive model for that specific engineering application. These affirmative answers have their origin in the following contributions of the *CompareML* approach:

- It allows engineers and researchers who have no extensive prior skills in machine learning to generate their own models with which to evaluate the potential knowledge that can be inferred from their data.
- It finds the machine learning algorithm that would build the most appropriate model for a specific problem involving some specific data.
- It determines the best framework, tools, and providers for addressing a specific problem involving some specific data.

In future work, it will be interesting to improve the approach with the following practices:

- Increase the number of libraries and services supported, as well as the number of regression and classification algorithms.
- Support algorithms to create clustering or recommender system models.
- Implement a microservices-based architecture that allows the functionalities of the machine learning providers to be isolated in those microservices [30], thus making it easier to update the algorithms, maintain the code, and add or delete providers.
- Implement new classification and regression model evaluation metrics.

Acknowledgments

This work was developed with the support of (i) Ministerio de Ciencia, Innovación y Universidades (MCIU), Agencia Estatal de Investigación (AEI), and European Regional Development Fund (ERDF): project RTI2018-098652-B-I00, and (ii) European Regional Development Fund (ERDF) and Junta de Extremadura: projects IB16055, IB18034, and GR18112.

References

[1] I. H. Witten, E. Frank, M. A. Hall, “Introduction to weka,” in *Data Mining: Practical Machine Learning Tools and Techniques (Third Edition)*, The Morgan Kaufmann Series in Data Management Systems, Boston: Morgan Kaufmann, 2011, pp. 403 – 406, third edition ed., doi: https://doi.org/10.1016/B978-0-12-374856-0.00010-9.

[2] S. Lang, F. Bravo-Marquez, C. Beckham, M. Hall, E. Frank, “Wekadeeplearning4j: A deep learning package for weka based on deeplearning4j,” *Knowledge-Based Systems*, vol. 178, pp. 48 – 50, 2019, doi: https://doi.org/10.1016/j.knosys.2019.04.013.

[3] J. Demšar, T. Curk, A. Erjavec, Č. Gorup, T. Hočevar, M. Milutinovič, M. Možina, M. Polajnar, M. Toplak, A. Starič, M. Štajdohar, L. Umek, L. Žagar, J. Žbontar, M. Žitnik, B. Zupan, “Orange: Data mining toolbox in python,” *Journal of Machine Learning Research*, vol. 14, pp. 2349–2353, 2013.

[4] M. R. Berthold, N. Cebron, F. Dill, T. R. Gabriel, T. Kötter, T. Meinl, P. Ohl, K. Thiel, B. Wiswedel, “Knime - the konstanz information miner: Version 2.0 and beyond,” *SIGKDD Explor. Newsl.*, vol. 11, p. 26–31, Nov. 2009, doi: 10.1145/1656274.1656280.

[5] I. Mierswa, M. Wurst, R. Klinkenberg, M. Scholz, T. Euler, “Yale: Rapid prototyping for complex data mining tasks,” in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’06, New York, NY, USA, 2006, p. 935–940, Association for Computing Machinery.

[6] A. Jovic, K. Brkic, N. Bogunovic, “An overview of free software tools for general data mining,” in *2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2014, pp. 1112–1117.

[7] X. He, K. Zhao, X. Chu, “Automl: A survey of the state-of- the-art,” *Knowledge-Based Systems*, vol. 212, p. 106622, 2021, doi: https://doi.org/10.1016/j.knosys.2020.106622.

[8] H. Song, P. Flach, “Efficient and robust model benchmarks with item response theory and adaptive testing,” *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 6, pp. 110–118, 2021, doi: https://doi.org/10.9781/ijimai.2021.02.009.

[9] Microsoft, “Powerbi automated machine learning.” https://docs.microsoft.com/en-us/power-bi/transform-model/dataflows/dataflows-machine-learning-integration. Online; last accessed 2 April 2021.

[10] M. Ali, *PyCaret: An open source, low-code machine learning library in Python*, July 2020. PyCaret version 2.3.

[11] Google, “Cloud automl.” https://cloud.google.com/automl. Online; last accessed 2 April 2021.

[12] H. Robles-Berumen, A. Zafra, H. M. Fardoun, S. Ventura, “Leac: An efficient library for clustering with evolutionary algorithms,” *Knowledge-Based Systems*, vol. 179, pp. 117 – 119, 2019, doi: https://doi.org/10.1016/j.knosys.2019.05.008.

[13] D. Charte, F. Herrera, F. Charte, “Ruta: Implementations of neural autoencoders in r,” *Knowledge-Based Systems*, vol. 174, pp. 4 – 8, 2019, doi: https://doi.org/10.1016/j.knosys.2019.01.014.

[14] E. Real, C. Liang, D. R. So, Q. V. Le, “Automl-zero: Evolving machine learning algorithms from scratch,” 2020.

[15] C. M. University, “Turi graphlab create.” https://turi.com/. Online; last accessed 2 April 2021.

[16] G. van Rossum, the Python Software Foundation, “Python programming language.” https://www.python.org/. Online; last accessed 2 April 2021.

[17] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[18] R Core Team, *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2013.

[19] A. Gupta, K. Ghanshala, R. C. Joshi, “Machine learning classifier approach with gaussian process, ensemble boosted trees, svm, and linear regression for 5g signal coverage mapping,” *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 6, pp. 156–163, 2021, doi: https://doi.org/10.9781/ijimai.2021.03.004.

[20] A. J. Fernández-García, L. Iribarne, A. Corral, J. Criado, J. Z. Wang, “A recommender system for component- based applications using machine learning techniques,” *Knowledge-Based Systems*, vol. 164, pp. 68–84, 2019, doi: https://doi.org/10.1016/j.knosys.2018.10.019.

[21] A. J. Fernández-García, R. Rodríguez-Echeverría, J. C. Preciado, J. M. C. Manzano, F. Sánchez-Figueroa, “Creating a recommender system to support higher education students in the subject enrollment decision,” *IEEE Access*, vol. 8, pp. 189069–189088, 2020, doi: 10.1109/ACCESS.2020.3031572.

[22] T. H.-Y. Chiu, C. Wu, R. C. C.-H. Chen, “A generalized wine quality prediction framework by evolutionary algorithms,” *International Journal of Interactive Multimedia and Artificial Intelligence*, doi: https://doi.org/10.9781/ijimai.2021.04.006.

[23] K. M. Ting, *Confusion Matrix*, pp. 260–260. Boston, MA: Springer US, 2017.

[24] A. Leff, J. T. Rayfield, “Web-application development using the model/view/controller design pattern,” in *Proceedings Fifth IEEE International Enterprise Distributed Object Computing Conference*, Sep. 2001, pp. 118–127.

[25] W. McKinney, “pandas: a foundational python library for data analysis and statistics,” *Python for High Performance and Scientific Computing*, vol. 14, 2011.

[26] S. Hellegouarch, *CherryPy Essentials: Rapid Python Web Application Development Design, Develop, Test, and Deploy Your Python Web Applications Easily*. Packt Publishing, 2007.

[27] M. Bohanec, V. Rajkovič, “Knowledge acquisition and explanation for multi-attribute decision,” in *8th International Workshop Expert Systems and Their Applications*, 1988.

[28] D. Dua, C. Graff, “UCI machine learning repository,” 2017. [Online]. Available: http://archive.ics.uci.edu/ml.

[29] A. Tsanas, A. Xifara, “Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools,” *Energy and Buildings*, vol. 49, pp. 560 – 567, 2012, doi: https://doi.org/10.1016/j.enbuild.2012.03.003.

[30] J. Lewis, M. Fowler, “Microservices: a definition of this new architectural term.” http://martinfowler.com/articles/microservices.html, 2014.

Antonio Jesús Fernández-García

He received his PhD in Computer Science from the Universidad de Almería in 2019. He is a Researcher and Member of the Applied Computing Group at the University of Almería (UAL) and the Quercus Software Engineering Group at the University of Extremadura (UEX). He has published more than 15 scientific publications in journals and international conferences, and has participated in more than 6 research projects. His research areas include Recommender Systems, Machine Learning, Artificial Intelligence, Data Mining, Data Engineering, and Software Engineering. At present, he is an Associate Professor in the Escuela Superior de Ingeniería y Tecnología at the Universidad Internacional de la Rioja (UNIR).

Juan Carlos Preciado

He is a Professor and member of the Quercus Software Engineering Group in the Department of Computer Science at the University of Extremadura (UEX). He was vice-rector of that university for several years. His research areas include Model-Driven Development, Web and Data Engineering, in which he has published around 100 papers in the software engineering field. He received a PhD in computer science from UEX in 2008.

Alvaro E. Prieto

He is an Assistant Professor of Computer Languages and Systems at the University of Extremadura (Spain). He is a member of the Quercus Software Engineering Group. He received his BSc in Computer Science from the University of Extremadura in 2000 and a PhD in Computer Science in 2013. His research interests include Linked Open Data, Predictive Analytics, and Business Intelligence. He is currently involved in various R&D&I projects.

Fernando Sánchez-Figueroa

He is a Professor in the Department of Computer Science at UEX. His research focuses on Web engineering, big data visualization, and MDD. He holds a PhD in Computer Science from UEX and is co-author of more than 100 publications related to software engineering.

Juan D. Gutiérrez

He is a Computer Engineer for the University of Extremadura. He combines a research staff job at this university with working towards a PhD in visible LED-based light indoor positioning systems (IPS). His skills include different programming languages, system administration, application design, databases, and the Internet. He has also written more than twenty computer science books and translated another ten books from English to Spanish.

unir
LA UNIVERSIDAD
EN INTERNET

Rectorado
Avenida de la Paz, 137
26006 Logroño (La Rioja)
t (+34) 941 21 02 11

www.unir.net