Data Leakage Detection Using Dynamic Data Structure and Classification Techniques*

Detección de Fugas de Información Aplicando Estructura de Dinámica de Datos y Técnicas de Clasificación

Research Article - Reception Date: June 20, 2014 - Acceptance Date: December 15, 2014

César Byron Guevara Maldonado
PhD Student in Computer Science Engineering. Master in Computer Science Research Computer Science and Information Engineer. Universidad Complutense de Madrid. Madrid (Spain) cesargue@ucm.es

To cite this paper:
C. B. Guevara Maldonado, "Data Leakage Detection Using Dynamic Data Structure and Classification Techniques", *INGE CUC*, vol. 11, no. 1, pp. 79-84, 2015

***Abstract*— Data leakage is a permanent problem in public and private institutions around the world; particularly, identifying the information leakage efficiently. In order to solve this problem, this paper poses an adaptable data structure based on human behavior using all the activities executed within the computer system. When applying this structure, the normal behavior is modeled for each user, so in this way, detects any abnormal behavior in real time. Moreover, this structure enables the application of several classification techniques such as decision trees (C4.5), UCS, and Naive Bayes, these techniques have proven efficient outcomes in intrusion detection. In the testing of this model, a scenario demonstrating the proposal's effectiveness with real information from a government institution was designed so as to establish future lines of work.**

***KeyWords*— Data Leakage, Data Structure, Decision Tree C4.5, UCS, Naive Bayes**

***Resumen*— La fuga de información es un problema que está presente en instituciones públicas y privadas alrededor del mundo. El principal problema que se presenta es identificar de forma eficiente el filtrado de la información. Para solucionar este problema en el presente trabajo desarrolla una estructura de datos adaptable al comportamiento humano, utilizando como base las actividades ejecutadas dentro del sistema informático. Al aplicar esta estructura se modela un comportamiento NORMAL de cada uno de los usuarios y de esta manera detecta cualquier comportamiento ANÓMALO en tiempo real. Además, permite la aplicación de varias técnicas de clasificación como los árboles de decisión (C4.5), UCS y Naive Bayes las cuales han demostrado un eficiente resultado en la detección de intrusiones. Para probar este modelo se ha diseñado un escenario que sirve para demostrar la validez de la propuesta con información real de una institución gubernamental y para acreditar líneas futuras de trabajo.**

***Palabras Clave*—Fuga de Información, Estructura de Datos,Árbol de decisión C4.5, UCS , Naive Bayes.**

* Research paper deriving from the research project "Detección de accesos fraudulentos en sistemas de información gubernamental". Funded by SENECYT Ecuador and the Universidad Complutense de Madrid. Starting date: February 2013. Ending date: February 2017.

I. Introduction

Data leakage is a major problem concerning most people and public and private companies since information is their most important asset in order to perform their job. Sensitive data of institutions include intellectual, financial, personal, and other sorts of information, depending on the organization or business' aim [1]. A study carried out by InsightExpress in the US, requested by Cisco [2], reveals that employees' behavior represents the major source of data loss threat mainly related to intellectual property or client's sensitive data. Another study performed by InfoWatch [3] reported a 16% increase in confidential data leakage compared to the previous year 2011. These data reveal a number of 2.5 leaks per day and between 75 and 80 per month. Moreover, when compared to 2008, leakage reaches a 40%.

Considering this threat situation, several researches have been implemented to avoid data breaches through data leakage prevention systems, also known as *DLP* –Data Leak Prevention.

DLP systems monitor computer resources to prevent sensitive data leakage from hard discs, databases, and more. To detect data leakage within computer resources, especially inside databases, data mining and machine learning [4] are frequently used to identify abnormalities within users' activities and create a behavior pattern so as to detect future data leaks.

DLP is an active front for development research of dynamic methodologies capable of adapting to the evolution of the more sophisticated and complex information system attacks. DLP strategies are in charge of detecting user's behavior and create a behavior-based profile to benefit from it in the future, and in this way, identify abnormal activities for each user in the system.

DLP user profile processes normal behavior when a user performs certain activities inside the system, any dissimilar conduct is registered as an abnormal activity with a high probability of data leakage identification. The main disadvantage is the great amount of false positives found during the analysis (confusing a normal behavior with an abnormality). Besides, users' behavior is quite difficult to model as a person may change the conduct depending on a situation or need. Another disadvantage is the large quantity of data required to model each user's behavior effectively during the year, for this reason, an ample period of time is necessary to gather information (1 to 2 years) so as to train classification techniques [5]. The present study applies this approach to improve detection efficiency, and in this way, heighten user's identification in information systems.

This paper sets forth a methodology and materials section describing in detail the elements used like the data structure design and the techniques selected for the DLP creation. Afterwards, in section III, the application of the data structure together with the classification techniques and configuration are posed, likewise, the training and testing tools used are also established. Section IV presents the results from the previous section; and finally, section V puts forward the conclusions and projects research lines for the future.

II. Materials and Methods

To begin with the creation of a DLP, the methodology for the construction of the classification model needs to be posed first. The phases involved in this methodology are:

1. Phase 1
   Selection of optimum classification techniques.
2. Phase 2
   Data structure construction using a government's entity database for data leakage detection (BD-ECU).
3. Phase 3
   Training and testing of the selected classification techniques using the data structure.

A. Phase 1: Classification Techniques Assessment.

This phase assesses diverse machine learning techniques with the dataset NSL-KDD [6]. This intrusion detection dataset has been studied and analyzed by McHugh [7] and Tavallaee [8], allowing the comparison of precision results in the classification techniques applied in those works and selecting the best performance methods for this study. Moreover, NSL-KDD contains user's behavior information patterns in information systems networks, from both intrusions and authorized activities, quite similar to those analyzed in this research.

As part of the DLP system development, a prior analysis of a specific number of classification techniques has been carried out to select the most suitable for the problem in question. The techniques applied are:

1. Multilayer Neural Network (MNN) [9].
2. Decision trees C4.5 and ID3 [10, 11, 12].
3. Support Vector Machines (SVM) [13, 14].
4. Bayesian networks (BN) [15].
5. Supervised learning (SL) [16, 17, 18].
6. NaiveBayes (NB) [19].

NSL-KDD dataset has 40 variables or attibutes that provide different type of information about accesses (protocols, timing, kinds of Access, etc.). For this research, 20% of the dataset was used, this means, 25192 records divided in this way: 13499 normal records and 11743 intrusion records; this information was applied for training and assessment, as suggested in [7].

Table I. Classification Techniques Comparison using NDL-Kdd Dataset

				CONFUSION MATRIX			
Method	**Appropriately Classified**	**MAE**	**Precision**	**TP Normal**	**TP Intrusions**	**FP Normal**	**FP Intrusions**
C4.5	99.96	0.04	0.9996	13444	11740	0	12
NB	99.69	0.31	0.9969	13408	11708	32	44
SL	99.23	0.77	0.9923	13346	11654	94	98
SVM	97.32	2.68	0.9732	13261	11258	485	188
MNN	97.08	2.92	0.9708	13064	11394	349	385
BN	89.59	10.41	0.8959	12272	10298	1445	1177

Source: Author

To evaluate the classification techniques, a K-fold cross validation was implemented in which the sample was partitioned into 10 subsamples (k=10) in each of the techniques applied in [17]. This test showed the percentage of appropriately classified cases and the precision of each classification technique.

Table I shows some indicators that help selecting the most suitable and efficient techniques, as well as the percentage of appropriately classified cases. Furthermore, intrusive and normal behavior indicators are also presented. The mean absolute error (MAE) of the classification method allows knowing the difference between 1 minus the precision value of the method. The method's precision enables to know its efficiency. The matrix illustrates the data regarding the tests performed to each technique, delivering information about the true positives (TP) and the false positives (FP).

As shown, the number of correct outcomes is high for all the methods since a large number of examples have been used for training.

Thanks to the test outcomes, the following have been identified as the techniques with the best results: Decision trees C4.5, Naive Bayes, and Supervised learning.

B. Phase 2: Data Structure Construction for DLP Development.

This section poses the phases of the data structure design; for this, the following are described in detail: ECU database attributes, data preparation, structure construction, and finally, database normalization for subsequent classification techniques.

Dataset with Government Information.

Dataset ECU comprises all the activity information from users inside an information system of an Ecuadorian Government Institution collected during 2011 and 2012. In order to conceal personal information, user id has been masked for every employee. The database accounts for 51690 records of 29 users who have carried out transactions inside the system; also, several attributes are considered:

1. *Day*: Day of the week in which the user performed the activity in the system; a round number comprising the period between 1 and 7.
2. *Hour*: The hour of the day in which the user performed the activity in the system; also, a round number (0 to 23).
3. *Minutes*: Minutes within the hour have fuzzy values, so the period comprising the first 15 minutes (0 to 15) has been assigned the round value 1; from 16 to 30, 2; from 31 to 45, 3; and finally, above 45 the value assigned is 4.
4. *Operation*: The type of operation performed by the user in the system. Each operation is encoded, in this way, Insert (1), Update (2), Delete (3), and Find (4).
5. *Table*: Represents the table in which the user performed the activity in the system; in this case, cadastral and real estate information managed by this entity. It is a round number ranging from 1 to 7.
6. *Department*: Department in which the user works; legal, technical, administrative, or financial. It is round number ranging from 1 to 4.
7. *Position*: It is the user's position within the organization, mostly lawyers and cadastral technicians. It is round number ranging from 1 to 4.
8. *City*: City in which the system's activity was performed; currently, Quito, Guayaquil, or Cuenca. It is round number ranging from 1 to 3.
9. *User id*: Identifies each user performing activities inside the information system. It is a round number ranging from 1 to 53.
10. *Type*: Attribute indicating whether the record belongs to an intruder "0" or to an authorized user "1"; also, a round value.
11. The dataset ECU is constituted by 54335 authorized records and 400 intrusion records. Detection was rather difficult since intrusion behavior is quite similar to that from a normal behavior.

Data Processing.

This stage sets forth the creation process of the data structure in three steps:

1) Data Preparation.

This step is necessary to make classification more efficient by reducing any kind of inconsistencies like redundant and noisy data, among others. "The fundamental purpose of data preparation is to manipulate and transform raw data so that the information content enfolded in the dataset can be exposed, or made more easily accessible" [20].

2) Data Normalization.

On the other hand, assessing the behavior of the dataset ECU, establishing the most relevant attributes, and reducing any kind of data noise or inconsistency are vital actions [21]. For this, some filters are used, as explained in [22, 23, 24, and 25], allowing more trustworthiness in subsequent intrusion detection stages. With the reliable data gathered in this step, more effective classification models can be achieved.

3) Database Data Collecting.

Consulting based on SQL statements was required to collect data in order to gather the correct information for subsequent tasks. For this reason, the following statements were made, as shown in Table II.

Table II. Common Data Coding Of Access Database

Mysql Database Consulting
Select day, hour, minutes, table, operation from audit group by day, hour, minutes, table, operation

Source: Author.

After consulting, a unique id is assigned to each of the records in ascending order with the prefix TSK; so in the next stage of behavior pattern building, these may be used more easily, as seen in Table III.

Table III. Outcomes for Common Data Coding of Access Database

CODE	DAY	HOUR	MINUTES	TABLE	OPERATION
TSK1	2	9	3	7	1
TSK2	2	10	1	4	2
TSK3	3	8	40	3	2
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
TSK20	4	14	2	5	3

Source: Author

Structure Creation for User Behavior Pattern.

In order to create the data structure, an activity cycle for a given period needs to be designed; as the user's behavior varies and data leakage occurs in brief periods of time, the cycle is designed taking into account *n* quantity of activities performed between logging in logging out the system. Fig. 1 illustrates the way the data structure was created.

As seen, there are three tables:

1. *Data Table:* Comprehends all the transactions executed by each of the users, both normal and leaks.

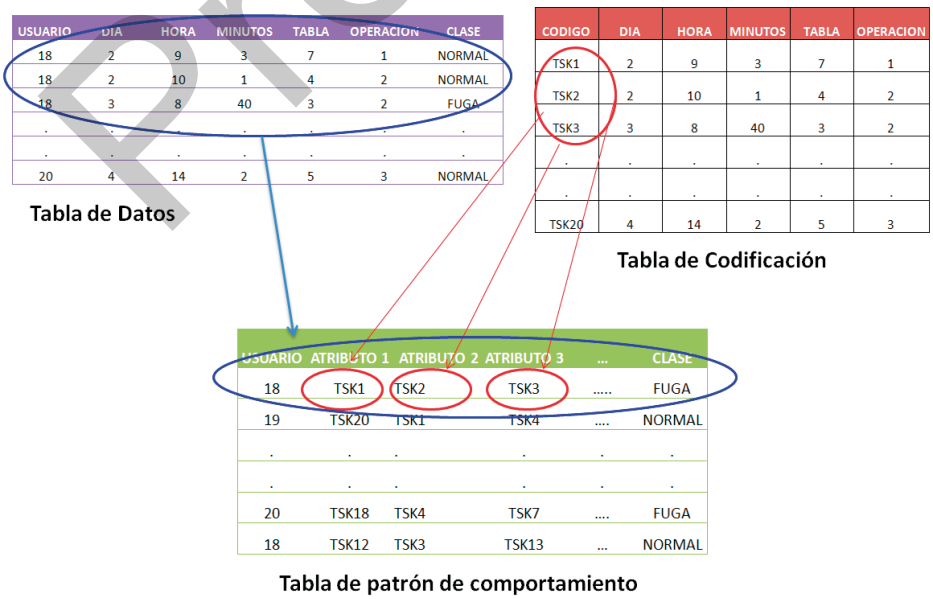

Fig. 1. Data integration for data structure creation
Source: Author.

EDUCOSTA

2. *Coding Table:* Transactions are coded to be reused during other user's activities.
3. *Behavior Pattern Table:* Behavior pattern cycles are created based on the two previous tables. As an example, user 18 has performed 3 activities during the session, 2 of them are normal and 1 is a leak. These activities were created in the *Coding Table* with these codes: TSK1, TSK2, and TSK3. Afterwards, in the Behavior Pattern Table these activities are organized according to their execution, this means, hour and minutes in which the activity was carried out. Finally, as this session encompasses a leakage activity, it is classified as LEAK.

With this data structure a personal profile can be created for every user and their behavior inside an information system, additionally, size and complexity of the data to be analyzed is reduced by 80%. The amount of records to be used for the classification system C4.5, SL, and Naive Bayes is 8533; 423 of them are data leaks and 8110 are normal transactions.

III. Estimation And Application

Phase 3: Training and testing of C4.5 using the data structure so as to apply what it has been explained so far.

A. Configuration

In order to train C4.5, SL, and Naive Bayes techniques with the data structure, some configuration processes are necessary for a correct classification.

Configuration SL
Number of Explores:100000
Pop Size:6400
delta:0.1
nu:10.0
acc_0:0.99
pX:0.8
pM:0.04
theta_ga:50.0
theta_del:50.0
theta_sub:500.0
do GA Subsumption: true
type Of Selection: RWS
tournament Size:0.4
Type of Mutation: free
Type of Crossover: 2PT
r_0:0.6
m_0:0.1
Configuration C4.5
Pruned: TRUE
Confidence: 0.25
Instances per Leaf: 2
Configuration NaiveBayes
NaiveBayes does not have configuration parameters.

B. Training and Testing

Once configured, techniques are trained and subdued to tests using the data structures of user behaviors: 70% of the data are destined for training purposes and 30% for testing, which was carried out with KEEL [26]. This software tool, developed in Spain, allows applying endless classification techniques and data mining tasks, evolutionary algorithms, etc. Fig. 2 presents the training and testing diagram.

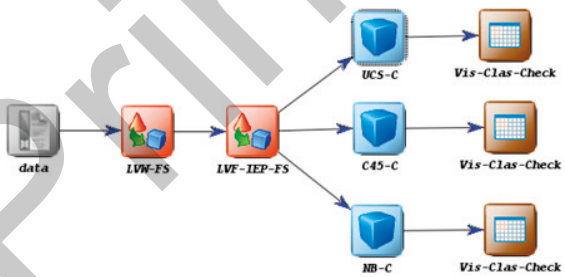

Fig. 2. Diagram of KEEL tool for training and testing of the model proposed. Source: Author.

IV. Results

To begin with, C4.5, SL, and Naive Bayes algorithms were applied to the proposed data structure and results with efficiency indicators were obtained as false positives and true positives. In the application of the data structure pattern an optimum outcome was achieved when using C4.5, reaching a 99.95% precision in true positives (8507 cases), this is, 0.001% of false positives (26 cases), meaning the model performs an efficient classification. SL classification technique achieved a 99% precision and Naive Bayes attained a 98.5%. Outcomes are shown in Table IV.

Table IV. Training and Testing of C4.5 Classification Method

	TRAINING				TESTING			
Algorithms	**Precision**	**Error**	**TP**	**FP**	**Precision**	**Error**	**TP**	**FP**
C4.5	0.995	0.005	8491	42	0.997	0.003	8507	26
Naive Bayes	0.985	0.015	8405	128	0.996	0.004	8499	34
UCS	0.99	0.01	8448	85	0.982	0.018	9379	154

Source: Author.

The information involved in this test is complex and ample since many examples were available to train the classification algorithm.

This study demonstrates that using the proposed data structure, data leakage detection improves the classification's precision and reduces false positives.

V. Conclusions

This study assesses the application of several classification techniques to a well-defined problem. It has been proven that data structures and C4.5, SL, and Naive Bayes algorithms provide an optimum identification percentage. The NSL-KDD dataset was used as a knowledge base to evaluate the classification methods, and the ECU dataset and data structure were applied to train and assess the algorithms for data leakage identification in information systems, and resulting highly efficient for data leakage detection.

Future research lines are mainly two: on the one hand, the application of the data structure proposed in more complex problems with multiple training examples and thousands of noisy data so as to evaluate detection performance; on the other hand, developing adaptive algorithms for human behavior capable of identifying data leakage, as well as intrusion detection, both in information systems and communication networks.

References

[1] A. Kumar, A. Goyal, N. K. Chaudhary, and S. Sowmya Kamath, "Comparative evaluation of algorithms for effective data leakage detection," in Information & Communication Technologies (ICT), IEEE Conference, 2013, pp. 177–182. DOI:10.1109/CICT.2013.6558085

[2] E. Summary, "Data Leakage Worldwide: Common Risks and Mistakes Employees Make," Europe, pp. 1–8, 2008.

[3] InfoWatch Research Center, "Global Data Leakages & Insider Threats Report, 2012". Disponible en: http://tech-titan.com/infowatch/pdf/InfoWatch%20Global%20Data%20Leakages%20and%20Insider%20Threats%20Report%202012.pdf

[4] W. L. W. Lee, S. J. Stolfo, and K. W. Mok, "A data mining framework for building intrusion detection models," IEEE Symp. Secur. Priv., vol. 00, no. c, pp. 120–132, 1999. DOI:10.1109/SECPRI.1999.766909

[5] C. Guevara, M. Santos and J. A. Martín, "Método para la Detección de Intrusos basado en la Sinergia de Técnicas de Inteligencia Artificial," in Proceedings of the IV Congreso Español de Informática CEDI 2013, pp. 963-972.

[6] NSL-KDD. Disponible en: http://nsl.cs.unb.ca/NSL-KDD/

[7] J. McHugh, "Testing Intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory," ACM Transactions on Information and System Security, vol. 3. pp. 262–294, 2000. DOI:10.1145/382912.382923

[8] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in IEEE Symposium on Computational Intelligence for Security and Defense Applications, CISDA 2009, 2009. DOI:10.1109/CISDA.2009.5356528

[9] M. Møller, "A scaled conjugate gradient algorithm for fast supervised learning," Neural networks, vol. 6, pp. 525–533, 1993. DOI:10.1016/S0893-6080(05)80056-5

[10] B. Widrow and M. A. Lehr, "30 years of adaptive neural networks: Perceptron, Madaline, and backpropagation," Proc. IEEE, vol. 78, no. 9, pp. 1415–1442, 1990. DOI:10.1109/5.58323

[11] J. Quinlan, C4.5: Programs for Machine Learning, 240th ed. Londres: Morgan Kaufmann, 1993.

[12] J. R. Quinlan, "Induction of decision trees," Mach. Learn., vol. 1, no. 1, pp. 81–106, 1986. DOI:10.1023/A:1022643204877

[13] J. C. Platt, "Sequential minimal optimization: A fast algorithm for training support vector machines," Adv. Kernel Methods Support Vector Learn., vol. 208, pp. 1–21, 1998.

[14] S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, and K. R. K. Murthy, "Improvements to Platt's SMO Algorithm for SVM Classifier Design," Neural Computation, vol. 13, no. 3. pp. 637–649, 2001. DOI:10.1162/089976601300014493

[15] D. Heckerman, "Bayesian Networks for Data Mining," Data Min. Knowl. Discov., vol. 119, no. 1, pp. 79–119, 1997. DOI:10.1023/A:1009730122752

[16] S. W. Wilson, "Classifier Fitness Based on Accuracy," Evolutionary Computation, vol. 3, no. 2. pp. 149–175, 1995. DOI:10.1162/evco.1995.3.2.149

[17] T. G. Dietterich, "Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms," Neural Computation, vol. 10, no. 7. pp. 1895–1923, 1998. DOI:10.1162/089976698300017197

[18] E. Bernadó-Mansilla and J. M. Garrell-Guiu, "Accuracy-based learning classifier systems: models, analysis and applications to classification tasks," Evol. Comput., vol. 11, no. 3, pp. 209–238, 2003. DOI:10.1162/106365603322365289

[19] P. Domingos and M. Pazzani, "On the Optimality of the Simple Bayesian Classifier under Zero-One Los," Mach. Learn., vol. 29, no. 2–3, pp. 103–130, 1997.

[20] D. Pyle, Data Preparation for Data Mining, 1st ed., vol. 1. San Francisco: Morgan Kaufmann, 1999. DOI:10.1023/A:1007413511361

[21] M. Basu, "Complexity measures of supervised classification problems," IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, no. 3, pp. 289–300, 2002. DOI:10.1109/34.990132

[22] H. Brighton and C. Mellish, "Advances in instance selection for instance-based learning algorithms," Data Mining and Knowledge Discovery, vol. 6, no. 2. pp. 153–172, 2002. DOI:10.1023/A:1014043630878

[23] F. Ceballos, L. E. Muñoz, and J. Moreno, "Selección de perceptrones multicapa usando aprendizaje bayesiano," Sci. Tech., no. 49, pp. 110–115, 2011.

[24] L. Rokach, "Ensemble-based classifiers," Artif. Intell. Rev., vol. 33, no. 1–2, pp. 1–39, 2010. DOI:10.1007/s10462-009-9124-7

[25] H. Liu and R. Setiono, "Feature Selection and Classification: A Probabilistic Wrapper Approach," in Proceedings of the 9th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, 1996, pp. 419–424.

[26] J. Alcala-Fdez, L. Sanchez, S. Garcia, M. J. del Jesus, S. Ventura, J. M. Garrell, J. Otero, C. Romero, J. Bacardit, V. M. Rivas, J. C. Fernandez, and F. Herrera, "KEEL: a software tool to assess evolutionary algorithms for data mining problems," Soft Comput., vol. 13, no. 3, pp. 307–318, 2009. Disponible: //www.keel.es/

EDUCOSTA