QUANTIZATION AND APPLICATION OF LOW-RANK TENSOR DECOMPOSITION BASED ON THE DEEP LEARNING MODEL

Jia Zhao*

1.- School of Tourism, Shanghai Normal University, Shanghai, 200234, China.

2.- School of Hospitality and Culinary Arts Management, Shanghai Institute of Tourism, Shanghai, 201418, China.

vivienne_yangyu@163.com

Reception: 20/11/2022 **Acceptance**: 13/01/2023 **Publication**: 27/02/2023

Suggested citation:

Z., Jia. (2023). **Quantization and application of low-rank tensor decomposition based on the deep learning model**. *3C TIC. Cuadernos de desarrollo aplicados a las TIC, 12(1)*, 330-350. https://doi.org/10.17993/3ctic.2023.121.330-350

https://doi.org/10.17993/3ctic.2023.121.330-350

ABSTRACT

Watching the presentation of a large-scale network is very important for network state tracking, performance optimization, traffic engineering, anomaly detection, fault analysis, etc. In this paper, we try to develop deep learning technology to solve the defect problem of tensor filling based on inner product interaction. To solve the limitations of the existing tensor-filling algorithms, a new neural tensor-filling (NTC) model is proposed. NTC model can effectively type the third-order communication between data landscapes through outer creation operation. It creates the third-order interaction mapping tensor. On this basis, the interaction between local features of the 3D neural network is studied. In this paper, another fusion neural tensor filling (Fu NTC) model is proposed to solve the problem that the NTC model can only extract the nonlinear complex structural information between potential feature dimensions. In the framework of the neural network, the NTC model and tensor decomposition model share the same potential feature embedding. It can effectively extract nonlinear feature information and linear feature information at the same time. It achieves higher precision data recovery.

KEYWORDS

Tensor filling; Sparse network monitoring; Deep learning; matrix; modeling

https://doi.org/10.17993/3ctic.2023.121.330-350

PAPER INDEX

https://doi.org/10.17993/3ctic.2023.121.330-350

1. INTRODUCTION

The rapid development and wide application of modern multimedia technology and information science. We need a higher level of technology to achieve data collection, storage, processing, and analysis. Massive data brings convenience to our life. Its scale is increasing. Its structure is becoming more and more complex. The data we collected may be incomplete. In addition, in the recommendation system, we also need to use the known data to infer the unknown data[1]. It can also be transformed into the problem of missing data recovery.

The estimation of missing values from the very limited information of an unknown matrix has attracted extensive attention in many fields. In practical applications, the target matrix is usually low rank or approximately low rank. For example, natural image data has a low-rank structure [2]. Therefore, a hypothesis is often used in the matrix filling: the matrix to be restored as a low-rank or near-low-rank structure [3-5]. The above problem is called the rank minimization problem [6]. Because the rank function is nonconvex and discontinuous. The solution to the above problem is the NPHard problem [7]. The original algorithm which can calculate the lowest rank solution of all instances needs at least an exponential time of matrix dimension [8]. In reference [9], Wei et al. Proposed two heuristic algorithms for approximate RMP based on convex optimization. And it proved that the kernel norm (i.e., the sum of singular values). The heuristic method is optimal in the sense of minimizing the convex envelope of rank functions. Subsequently, a series of theoretical studies were carried out to prove that. The kernel norm is a good convex proxy for the minimization of rank functions [10]. Yang et al. [11] proved that the kernel norm is the most compact convex lower bound of the rank function. And the relationship between kernel norm and matrix rank is similar to that between the L1 norm and l0 norm of the vector. Therefore, many scholars have implemented the rank kernel function as a surrogate matrix. The fixed-point extension and approximate singular value decomposition algorithm for solving the rank minimization problem of large-scale matrices [12] proposed by Qiao et al. Ahn et al. Provide a boundary for the number of elements needed to reconstruct a low-rank matrix. It is optimal in the range of a small numerical constant and a logarithmic factor [13-15]. In addition, some studies have shown that, under certain constraints, the minimum kernel norm can be filled by partial observation elements of the matrix [16].

The kernel norm minimization problem was first proposed. One of the most advanced semi-definite programming algorithms, to solve the problem. The algorithm is based on the interior point method. It needs to solve a large number of linear equations to calculate Newton's direction. When the matrix scale is large. The algorithm is not suitable. Ai proposed the singular value threshold algorithm (SVT) for the approximate solution of the kernel norm minimization problem in 2010. It proved that the method is suitable for large-scale matrix-filling problems [17].In addition, Saeedi et al. Proposed a fast approximate gradient descent method for solving kernel norm regularized linear least squares problems [18].In practical applications, the above-mentioned kernel norm algorithm and some other kernel norm algorithms may only obtain suboptimal solutions. A large number of iterative converge is required [19].

https://doi.org/10.17993/3ctic.2023.121.330-350

An important reason for this is that the kernel norm cannot approximate the rank function well in practical applications. Specifically, in the rank function, all non-zero singular values are treated equally. The kernel norm is to add all singular values. It leads to different singular values in the process of optimization. In addition, the theoretical requirements of heuristic kernel norms are difficult to meet in practical applications [20]. Therefore, some kernel norm extension algorithms have been proposed. Zhang et al. Used joint minimization of Schatten PNorm and LPNorm is used to recover the incomplete noise matrix. And it is applied to actual scenarios such as collaborative filtering and social network link prediction in the recommendation system [21]. CHEN et al. Proposed to use truncated kernel norm to better approximate matrix rank function. The truncated kernel norm is obtained by subtracting the sum of the largest singular values from the kernel norm [22]. At the same time, a new matrix-filling algorithm is proposed to minimize the truncated kernel norm of the matrix. At the same time, three effective iterative algorithms are developed to solve the truncated kernel norm minimization model: TNNRADMM、TNNRAPGL、TNNRADMMAP. The first type (TNNRADMM is an alternative multiplier method (ADMM). Firstly, according to a new ADMM update rule, the adaptive penalty parameter is used to accelerate the convergence speed. Among them, the ADMM method has also been widely used in image processing, such as non-local low-rank regularization [23] compressed sensing and matrix decomposition based on dual kernel norm for occluded image restoration [24], imprecise low rank and structural sparse decomposition [25]. Subsequently, the TNN algorithm has been successfully applied in many fields by many scholars. Liu et al. Proposed a matrix-filling TNN algorithm based on weighted residuals [26]. Liu proposed a more efficient TNN algorithm to deal with the matrix filling of high dynamic range images [27]. In addition, the reference research [28] further shows that TNN can be used as a better alternative to the rank function.

In the field of computer vision and signal processing, a large number of multidimensional data need to be analyzed and processed. In particular, multidimensional arrays (i.e., tensors) provide a natural representation of these data. Tensor is considered a generalization of a multivariate linear matrix or vector. Since tensor is a mathematical model which can establish multi-dimensional data structure. The study and research of tensors have attracted great attention. In recent years, more and more people use it in computer vision [29], machine learning [30], signal processing [31], pattern recognition [32], and other fields. However, due to technical limitations, the tensors we observe in real life are usually incomplete. It makes the application of tensors a challenging problem.

Because of previous studies, the goal of this paper is to recover the original data from partially lost observation matrix and tensor data. And apply them to practical applications, such as spectral data recovery, image/video data, text analysis, multitask learning, recommendation system, etc.

https://doi.org/10.17993/3ctic.2023.121.330-350

2. NEURAL TENSOR FILLING MODEL FOR PRECISE NETWORK MONITORING

In the modern data center, there are tens of thousands of servers. And the network scale is constantly expanding. Network monitoring, monitoring equipment, and communication costs will be a huge challenge. To effectively reduce the measurement cost. It measures the network performance data of some nodes and paths. It reconstructs other unmeasured networks. Therefore, in the case of sampling. How to reconstruct and infer the unmeasured network performance data. And how to ensure the push. The accuracy of measurement has become a prominent problem in network monitoring.

2.1. DESCRIPTION OF THE NTC MODEL

To solve the limitations of the existing tensor filling algorithms. A new neural tensor filling (NTC) model is proposed. The multi-layer architecture is adopted. And the output of the previous layer is used as the input of the next layer to model the source node target node time interaction. **Figure.1** is the overall framework of the NTC model. From bottom to top, there are five main functional modules: input sheet, inserting layer, interactive mapping sheet, feature abstraction layer, and prediction layer.

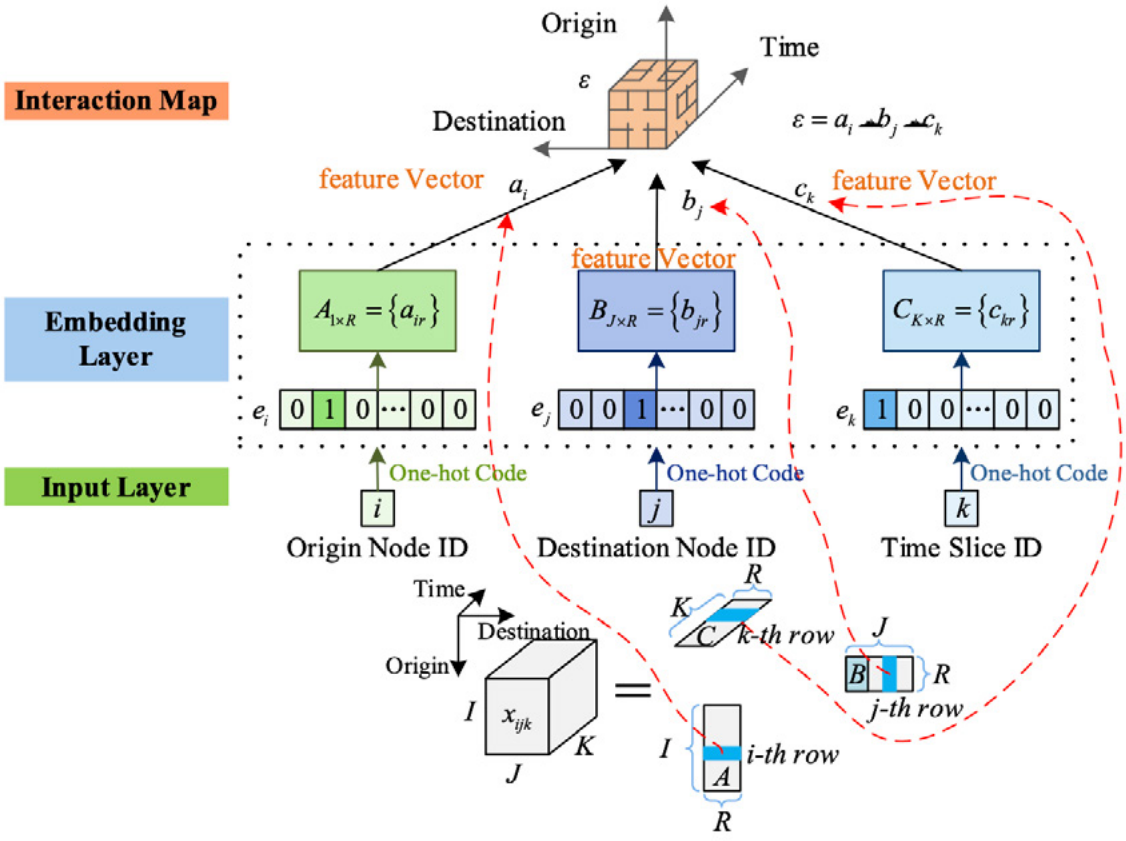


Figure 1. The overall framework of the NTC model

https://doi.org/10.17993/3ctic.2023.121.330-350

At the lowest input sheet, there are three indicator vectors e_i, e_j, e_k. The source node i, target node j, and moment hole k are described separately. Next, the embedded sheet is fully associated. It designs the source node, target node, and time hole onto a feature direction respectively.

$$\min_{\mathbf{A},\mathbf{B},\mathbf{C},\Theta_f^C,\Theta_f^M} \sum_{i,j,k\in\Omega} \left(x_{ijk} - f\left(\mathbf{e}_i\mathbf{A}, \mathbf{e}_j\mathbf{B}, \mathbf{e}_k\mathbf{C}\Theta_f^C, \Theta_f^M\right)\right)^2 + \alpha\|\mathbf{A}\| + \beta\|\mathbf{B}\| + \gamma\|\mathbf{C}\| \quad (1)$$

Among them, $A \in R^{I\times R}$, $B \in R^{J\times R}$, $C \in R^{K\times R}$. The potential characteristic matrix represents the source node, target node, and time respectively. e_i, e_j and e_k is a unique hot code vector indicating the source node i, the target node j, and the time slot k, respectively. α, β and γ is the regularization coefficient.

2.2. DETAIL OF THE NTC MODEL

Inspired by this, the input and embedding layers of a neural network are designed to obtain a potential eigenvector of the source node, target node, and time slot. Given the ID of the source node i, target node j, and slot k, their embedding vectors can be obtained by the formula a_i, b_j and c_k:

$$\mathbf{a}_i = \mathbf{e}_i\mathbf{A}, \mathbf{b}_j = \mathbf{e}_j\mathbf{B}, \mathbf{c}_k = \mathbf{e}_k\mathbf{C} \quad (2)$$

Among them, $e_i \in R^{1\times I}$, $e_j \in R^{1\times J}$ and $e_k \in R^{1\times K}$The unique hot ID indication vectors of the source node i, target node j, and slot k are represented respectively. On top of the embedded layer, 3D interactive mapping is recommended to represent the interaction between potential features.

$$\varepsilon = \mathbf{a}_i \; o \; \mathbf{b}_j \; o \; \mathbf{c}_k \quad (3)$$

Among them, ε was an $R \times R \times R$. In the 3D measurement of R, each of its values (x, y, z) could be calculated. This was the main project of the NTC model to safeguard the authenticity of the NTC model's data speculation. The main compensations of using this 3D communication map to express the communication between features were:

1. Compared with the traditional inner product operation, the outer product of the 3D interactive mapping tensor can capture the interaction between dimensions more effectively. Because the embedded dimensions of the inner product are independent of each other.
2. The 3D interaction and map structure were good for high-level relevant models. And they could be regarded as a kind of tornado in the depth. He could use the powerful 3D CNN to learn the complicated structure concealed in the nursing data. So that he could restore the missing data more accurately.

To extract the potential information from the net checking data, a simple key was to use the MLP net. Although MLP was hypothetically certain to have a robust aptitude to express information. It had the shortcoming of a large number of arguments that could not be ignored.

https://doi.org/10.17993/3ctic.2023.121.330-350

To solve the shortcomings of MLP, it is suggested to use 3D CNN to extract hidden features on a 3D interactive map.3D CNN stacks layers in the way of local connection and limits distribution. It uses much fewer limits than the MLP network. It makes it possible to build a hidden 3D CNN model than an MLP network [33-34]. The hidden 3D CNN typically can better learn the high-order correlation between embedded dimensions, thus bringing higher accuracy for the recovery of missing data.

In the design of 3D CNN, there were L roll-up layers and a full connect sheet. The input of 3D CNN was the 3D communication map $\varepsilon \in R^{R\times R\times R}$. It seemed that he was not going to give up. In each layer, there were T-packing cores (or filters). On each floor, the three-dimensional operation of the crumple core and the Re LU was performed first. And then a deviation was added. Using the re Lu as the activation function, the three-dimensional transformation was carried out. And a new one was obtained. The structure design of 3D CNN was to cutting potential topographies from the 3D communication map. It was achieved through the three-dimensional operation between the purification cores. The various layers of the input Hebe. Because one of the two kinds of cores could only excerpt one kind of topographies from the input Hebe. To excerpt more kinds of topographies. T (t) and 1 (more than one) of the 3D CNN were used. Therefore, there were T feature maps on each layer of the tornado. And each of them stored the data extracted by a core of the tornado. In the simulation test, how the number of the spiral core T would affect the performance was displayed. And the data was set according to the test.

In each convolution sheet, the input of the first sheet can be expressed as a ε^{l-1}. When I = 1, its contribution is a 3D interactive mapping tensor ε^{l-1}. After convolution, the i-th feature chart extracted from the L-th convolution sheet can be expressed as follows:

$$\mathrm{S}_i^l = \mathrm{ReLU}\left(G_i^l \otimes \mathrm{E}^{l-1} + b_l\right) \tag{4}$$

Among them, S_i^l represents the i-th chin record extracted on the L-th complication sheet. $\otimes$ represents the convolution operation, b_l represents deviation. The (x, y, z) term of the i-th typical diagram of the layer l can be obtained by the following formula:

$$\begin{cases} e_{xyz}^{li} = \mathrm{ReLU}\left(b_{li} + \sum_{p=0}^{l_R-1}\sum_{q=0}^{l_R-1 l_{R-1}}\sum_{r=0}^{li} g_{pqr}^{li} e_{(x+p)(y+q)(z+r)}^{0}\right), l = 1 \\ e_{xyz}^{li} = \mathrm{ReLU}\left(b_{li} + \sum_{m=0}^{T-1}\sum_{p=0}^{l_R-1}\sum_{q=0}^{l_R-1 l_R-1}\sum_{r=0}^{lim} g_{pqr} e_{(x+p)(y+q)(z+r)}^{(l-1)m}\right), l > 1 \end{cases}$$

Among them, e_{xyz}^{0} means the (x, y, z)item of the 3D communication chart. b_{li} means the deviation term of the i-feature map of level l. l_R was the scope of the spiral core beside the source node, the target node. And the time direction. g_{pqr}^{li} and g_{pqr}^{lim} is the (p, q, r) term of the i layer of the l sheet.

The latter sheet of the NTC model was the prediction sheet. It accepted the production of the previous chin-extracting sheet. Then the final deduction data was generated. On this floor, he first inputs into a single-sheet observation device. And then used the Sigmoid function σ (·) to calculate $\hat{x}_{ijk}$:

https://doi.org/10.17993/3ctic.2023.121.330-350

$$\hat{x}_{ijk} = \sigma\left(\mathbf{w}_{out}^{T}\mathbf{s} + b\right) \tag{6}$$

Among them, $W_{out}^{R} \in R^{R}$ a vector is a weight. $s \in R^{R}$ is the expansion vector of 3D CNN output. B is a deviation term.

2.3. THEORETICAL ANALYSIS

The NTC model is assumed to have a half-barbican structure. The convolution kernel is 2×2×2.

$$k_1 = 3(l+1) \tag{7}$$

The theorem is proved by mathematical induction.

The input layer maps the tensor (, each of its entries) for 3D interactively e_{xyz}. It encodes the third-order interaction between embedded dimensions $[x; Y; Z]$. In this case, $l = 0,\ K_0 = 3$.

When $l = 1$, the size of the convolution kernel is $2 \times two \times 2$. convolution layer L captures 2 of the input 3D interactive mapping tensor. It can capture the sixth-order features of the input layer. The entries in the first feature map of the first convolution layer depend on the eight elements of the previous layer, namely, the 3D interactive mapping tensor.

When $Kt = 1$, the feature can be captured. When $Kt = 1$, the convolution layer $T+1$ captures 2 of the convolution layers $t \times two \times 2$. The feature of the local region, and the term of the i-th feature map of the $(T+1)$ layer $e_{xyz}^{(t+1)i}$. The interaction between embedded dimensions $[x; X+1; Y; y+1; Z; Z+1]$ can be captured. Where the interaction between embedded dimensions $[x+1; y+1; Z+1]$ is new to the $(T+1)$ level of convolution. And $[x; Y; Z]$ can be inherited from the previous level $t._{t+1} re, K(M203) = 3 + KT = 3 + 3(T+1) = 3((T+1)+1)$.

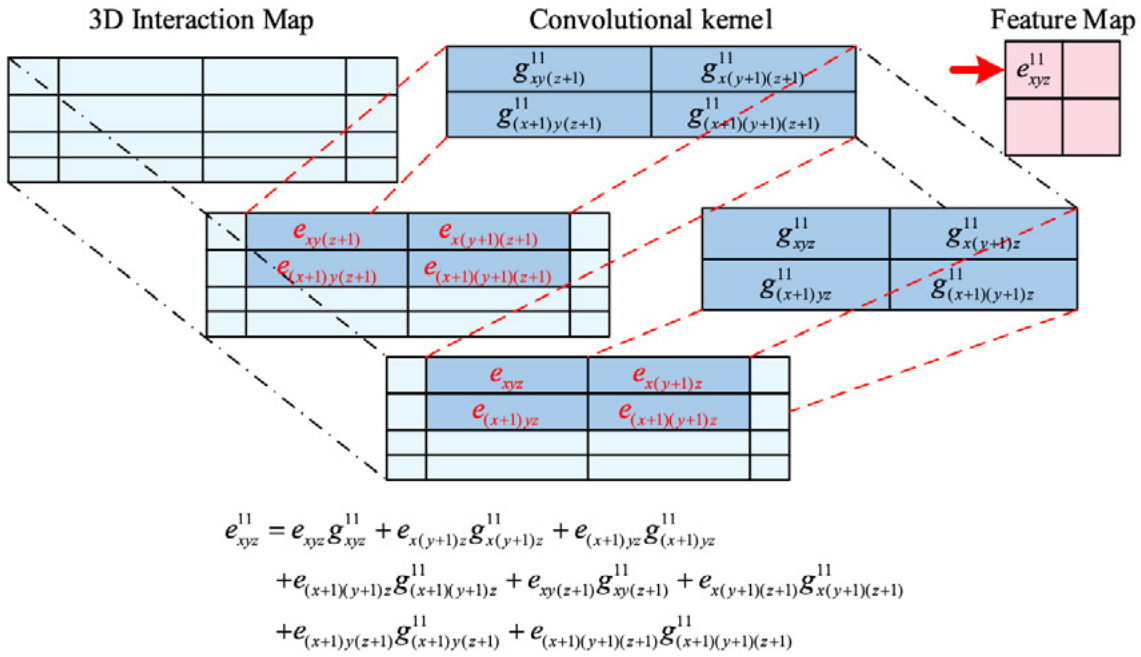


Figure 2. Example of high-order feature extraction

https://doi.org/10.17993/3ctic.2023.121.330-350

CP disintegration was the most common way to solve the problem. It was widely studied. It showed how to interpret the CP division as an exception to the NTC model. The internal product function was used as the interaction function in the CP disintegration. The internal product of a_i, b_j, and CK were the elements on the diagonally opposite side of the a_i, b_j, and CK. If one wanted to express the standard CP decomposed model with the NTC model, he needed to do the following:

1. By deleting the feature extracting layer based on 3D CNN, the three-dimensional interaction map layer in NTC was connected with the prediction layer directly.
2. The diagonally opposite three-dimensional interaction map was unfolded, forming the input of the prediction layer, and then projected the input to the prediction layer.

$$\hat{x}_{ijk} = a_{out}\left(\mathbf{h}^T\mathbf{h}\right) \tag{8}$$

Among them, V was the unfolded arrow, a_{out} and h respectively showed the activation functions and parameters of the prediction layer. To put it bluntly, if a_{out} constant functions of out and h were set to 1. So the revised NTC has filled in the model based on the CP disintegration.

2.4. 2.4 EXPERIMENTAL SIMULATION OF THE REAL DATA SET

Represent raw data as $\chi \in R^{I\times J\times K}$ about x_{ijk}. To process the data more effectively, we use the inverse function of σ. They represent matrix and variance respectively

$$x_{ijk} = \mu + \sqrt{2\sigma erfinv}\left(2x_{ijk} - 1\right) \tag{9}$$

Performance index: in the experimental simulation, four evaluation indexes are used to evaluate the performance of the NTC model: training error ratio (SER).

$$SER = \frac{\sqrt{\sum_{i,j,k\in\Omega}\left(x_{ijk} - \hat{x}_{ijk}\right)^2}}{\sqrt{\sum_{i,j,k\in\Omega} x_{ijk}^2}} \tag{10}$$

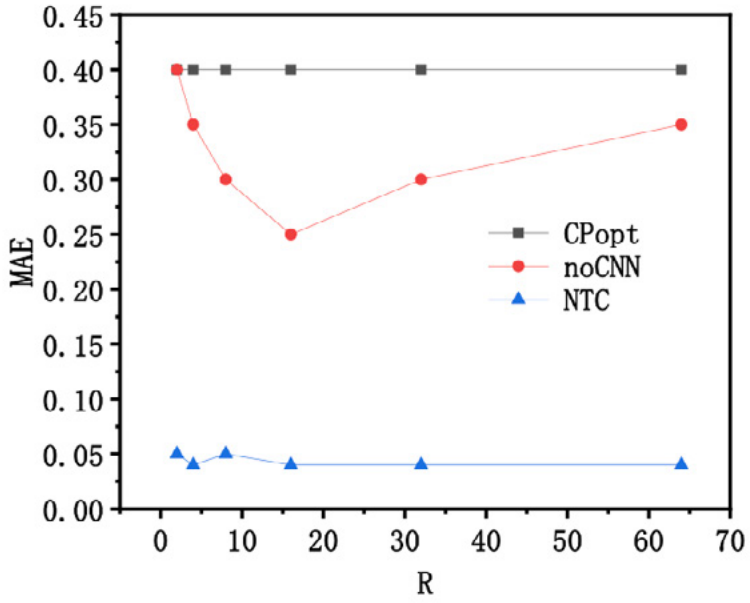
$$TER = \frac{\sqrt{\sum_{i,j,k\in\bar{\Omega}}\left(x_{ijk} - \hat{x}_{ijk}\right)^2}}{\sqrt{\sum_{i,j,k\in\bar{\Omega}} x_{ijk}^2}} \tag{11}$$

$$MAE = \frac{1}{m}\left(\sum_{i,j,k\in(\Omega+\bar{\Omega})}\left|x_{ijk} - \hat{x}_{ijk}\right|\right) \tag{12}$$

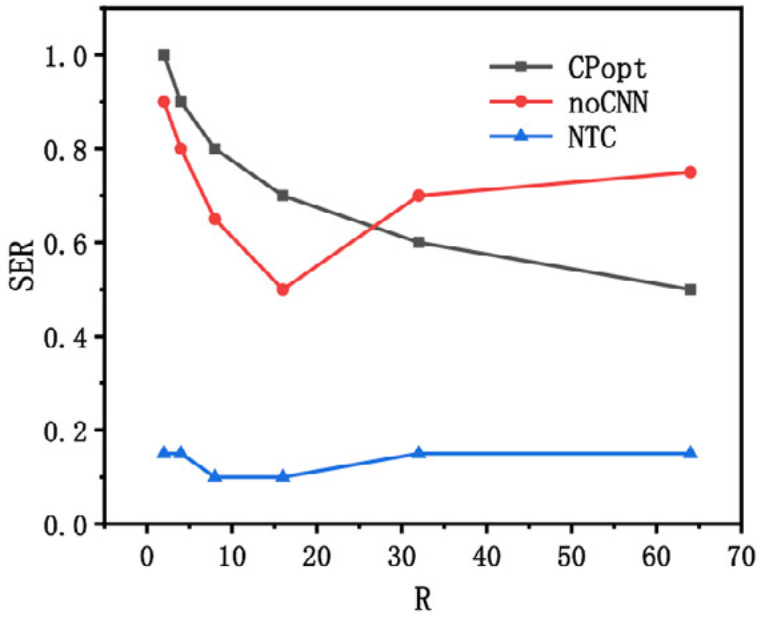
$$RMSE = \sqrt{\frac{1}{m}\sum_{i,j,k\in(\Omega+\bar{\Omega})}\left(x_{ijk} - \hat{x}_{ijk}\right)^2} \tag{13}$$

https://doi.org/10.17993/3ctic.2023.121.330-350

x_{ijk} and $\hat{x}_{ijk}$ represent the monitoring data tensors respectively χ. The (I, J, K) original monitoring data and predictive monitoring data. Only the training error ratio SER was calculated for the sampled data set. And the test error ratio TER was calculated for the non-sampled data set. The mean absolute error MAE and root mean square error $RMSE$ was calculated for all data items.



(a)



(b)

Figure 3. The influence of spatial dimension (R) on potential features

Effects of CNN mode. In this comparative experiment, in addition to 3D CNN, 2dcnn was implemented to extract potential features of missing data. As expected, 3D CNNs have much better recovery performance than 2D CNNs. The convolution of 3D

https://doi.org/10.17993/3ctic.2023.121.330-350

CNNs in the depth direction explicitly captures higher-order features of network monitoring data in time. 2D CNNs cannot achieve in such an unambiguous way.

Table 1. Recovery performance of Abilene

Model	TER						MAE					
	1 %	3 %	5 %	7 %	9 %	11 %	1 %	3 %	5 %	7 %	9 %	11 %
CP-als	0.99	0.98	0.96	0.95	0.94	0.93	0.55	0.55	0.54	0.53	0.52	0.52
CP-nmu	0.96	0.98	0.96	0.95	0.93	0.93	0.5	0.5	0.54	0.53	0.52	0.51
CP-opt	0.99	0.98	0.96	0.95	0.94	0.93	0.55	0.55	0.54	0.53	0.52	0.52
No-CNN	1.99	1.12	0.11	0.07	0.06	0.05	0.89	0.51	0.54	0.53	0.52	0.52
No-OUT	0.05	0.04	0.03	0.03	0.03	0.03	0.03	0.02	0.02	0.01	0.01	0.01
NTC	0.01	0.03	0.03	0.02	0.02	0.02	0.02	0.01	0.01	0.01	0.01	0.01
Advance	21	27	31	32	33	33	26	36	40	43	43	44

Table 2. Recovery performance of ws-dream

Model	TER						MAE					
	1 %	3 %	5 %	7 %	9 %	11 %	1 %	3 %	5 %	7 %	9 %	11 %
CP-als	0.99	0.97	0.95	0.93	0.91	0.90	0.52	0.51	0.50	0.49	0.48	0.48
CP-nmu	0.99	0.97	0.95	0.93	0.91	0.90	0.52	0.51	0.50	0.49	0.48	0.48
CP-opt	0.99	0.97	0.95	0.93	0.91	0.90	0.52	0.51	0.50	0.49	0.48	0.48
No-CNN	2.85	0.95	0.72	0.60	0.40	0.25	1.20	0.85	0.44	0.33	0.23	0.21
No-OUT	0.15	0.14	0.14	0.11	0.11	0.10	0.06	0.06	0.05	0.04	0.04	0.03
NTC	0.12	0.11	0.11	0.10	0.10	0.02	0.06	0.04	0.04	0.04	0.03	0.03
Advance	8	8	9	9	9	9	9	11	12	12	13	14

2.5. REAL NETWORK PLATFORM EXPERIMENT

To more effectively verify the accuracy of the proposed NTC model for missing data recovery of a network monitoring system, the NTC model will be applied to the network monitoring center in the later stage of the experiment. The algorithm of the NTC model will be integrated into the network monitoring platform. Large-scale actual network deployment will be carried out. Then, experiments are carried out in a real network environment to evaluate, verify and improve the proposed algorithm.

It is proposed to adopt the software. The hardware-integrated platform of high-performance data packet collection and intelligent analysis. It can realize high-speed packet capture and obtain the latest data sets of the backbone network and service

https://doi.org/10.17993/3ctic.2023.121.330-350

requests of operators. It includes the following aspects: one-way delay, round-trip delay (QoS), round-trip delay (QoS), packet loss rate (QoS), packet loss rate (QoS), packet loss rate (QoS), packet loss rate (QoS), packet loss rate.

In the actual deployment of the NTC model, a small and medium-sized LAN is proposed to collect monitoring data every 10 minutes to continuously measure the network performance data for 7 days, mainly measuring network delay and bandwidth. The measured data of this week were taken as the total sample of NTC model training. In the subsequent validation of the NTC model, the first task is to train the parameters of the NTC model in this real network monitoring platform, including the dimension of latent feature space (R) and the number of convolution kernels (T).In the previous real data set simulation experiments, we can see that there are generally a better R-value and t-value in the data, not that the larger the R-value and t-value, the better. After training the optimal R-value and t-value, to verify that the NTC model can also have good recovery accuracy in the case of a large data loss rate, a random collection of 1% to 10% of the total sample data for model training and missing data recovery. Finally, the performance of the proposed NTC model is evaluated and verified by comparing it with the real data.

Because the deployment of the NTC model in a real network monitoring platform is not enough to evaluate the performance of the NTC model. It is planned to implement all the comparison algorithms on the platform. To evaluate the effectiveness of the NTC model through comparative analysis.

3. FUSION NEURAL TENSOR FILLING MODEL FOR MORE COMPREHENSIVE FEATURE EXTRACTION

One model based on tensor filling uses a linear kernel to extract potential features from data. And the other model based on a neural network uses a nonlinear kernel to learn interaction functions in data. However, the actual network monitoring data may contain both linear and nonlinear features. Therefore, it is not comprehensive to adopt a single linear kernel or a nonlinear kernel. Then, to better improve the recovery performance of network monitoring data unmeasured data, the next problem is how to integrate the model based on tensor filling. And the model is based on a neural network. So that they can enhance each other. To better model the complex data interaction. It improves the information extraction of data.

3.1. SOLUTION OVERVIEW

To fuse linear and nonlinear frameworks, a simple solution is to take one of the models as the main one and express the other model under its framework. To fuse them. The NTC model is a typical non-linear model. It can be expressed and extended by other algorithms in its framework. It is proved that the tensor filling model based on CP decomposition is a special case of NTC. Therefore, a simple solution is to use a neural network to represent CP and let NTC and CP share the same embedding layer. That is a potential eigenvector. And then combine the output of their interaction

https://doi.org/10.17993/3ctic.2023.121.330-350

function to predict the recovery data. This method is similar to the famous neural tensor network (NTN).

The model combines the inner product and outer product of potential features to realize the fusion of linear and nonlinear latent features. Therefore, the model is defined as a fusion neural tensor completion (Fu NTC) model. The structure of the Fu NTC model is similar to that of the NTC model. It also adopts multi-layer hierarchical architecture.

3.2. DETAIL OF FU NTC MODEL

The essential purpose of input and inserting layer operations is to represent the potential characteristics of source node i, target node j, and time k by the ID of source node i, target node j, and time k, respectively

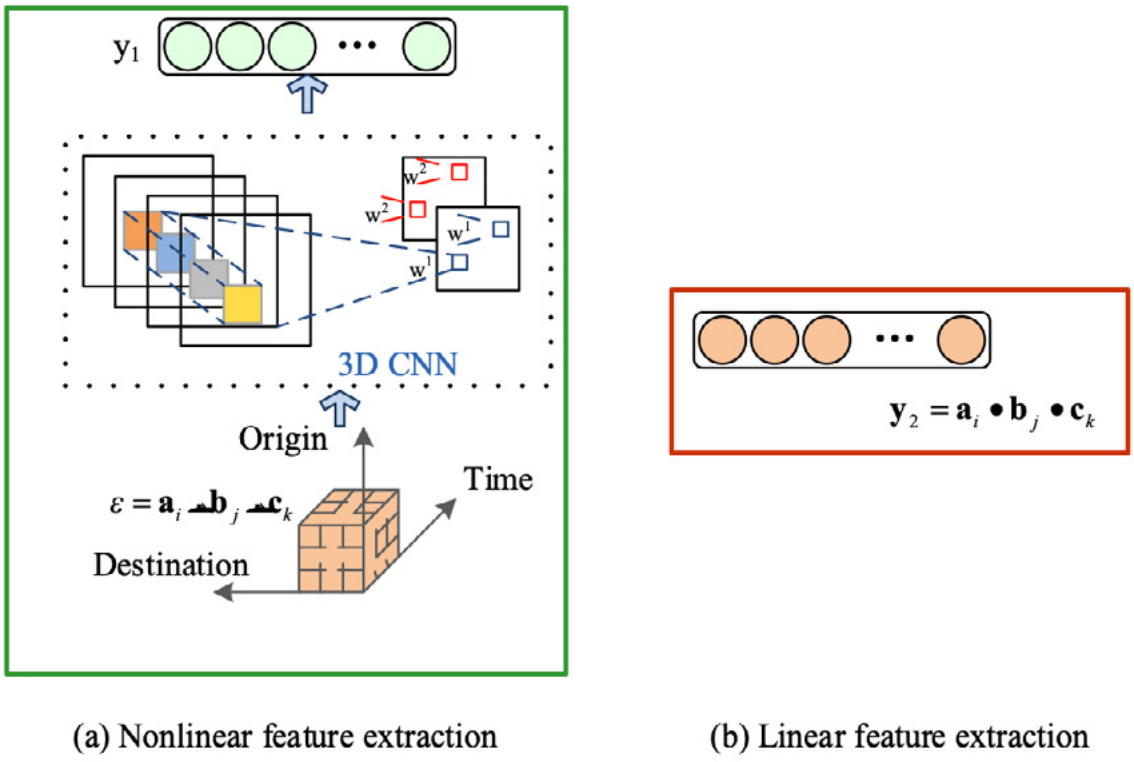
$$\mathbf{a}_i = \mathbf{e}_i\mathbf{A}, \mathbf{b}_j = \mathbf{e}_j\mathbf{B}, \mathbf{c}_k = \mathbf{e}_k\mathbf{C} \tag{14}$$

Among them, the potential eigenvectors of the source node i, target node j, and time k are a_i, b_j and c_k. $A \in R^{I\times R}$, $B \in R^{J\times R}$ and $C \in R^{K\times R}$. The feature nodes are embedded in the feature matrix of the source node. And the target node respectively. The latent characteristic matrices A, B, and C contain the features of the whole network monitoring data. It is the key point of the whole model training.

In theory, the performance of the fusion model may be limited if the linear model. And the nonlinear model shares the same embedding layer. Because this means that the linear model. Because the nonlinear model must use the same size as the embedding vector. For the two models with large changes in the optimal embedding dimension, the optimal feature extraction may not be obtained. Moreover, using the same embedding layer, the fusion model is not flexible enough.

The feature extraction layer of the Fu NTC model is divided into two parts: linear feature extraction and nonlinear feature extraction. The nonlinear feature extraction follows the NTC model. And it uses the outer product of the potential feature vector as the interaction function to generate a 3D interactive mapping tensor. And then it uses 3D CNN to extract high-order features. The nonlinear model can extract effective information from network measurement data. And the recovery performance of unmeasured data is also verified in the simulation experiment in the previous chapter. However, linear feature extraction is the representation of CP decomposition in the neural network. Different from the traditional CP decomposition, the result of the inner product is not added at this time. Instead, it is extended into a tensor to facilitate the fusion operation of the feature fusion layer.

https://doi.org/10.17993/3ctic.2023.121.330-350



(a) Nonlinear feature extraction (b) Linear feature extraction

Figure 4. Linear feature extraction and nonlinear feature extraction

In the feature fusion layer, the feature vector y_1 is obtained from the nonlinear model. It is spliced with the feature vector y_2 obtained from the linear model to form a fusion vector

$$\mathbf{z} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix} \tag{15}$$

Where [] represents the vector splicing operation. The feature vector z contains both nonlinear features and linear features. During the network training, the reverse update operation will adjust the nonlinear model and the linear model synchronously. To realize the synchronous extraction of nonlinear features and linear features for the whole model.

It is only different from the NTC model which only uses single-layer MLP. The number of MLP layers in the Fu NTC model needs to be adjusted dynamically.

3.3. REAL DATA SET EXPERIMENTAL SIMULATION

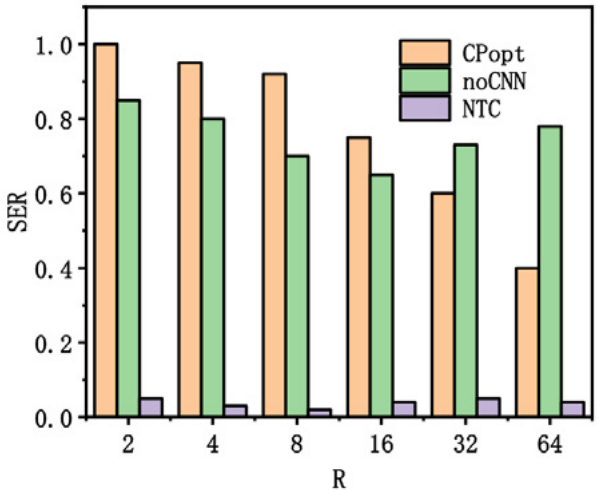
An equally large number of experiments were performed in this study to answer the following questions:

1. Effect of key hyperparameters (i.e., potential feature space dimension (R), number of convolution kernels (T), and number of convolution layers (L)) on the performance of the Fu NTC model.
2. Whether the CNN-based mode has an impact on the recovery performance of Fu NTC.

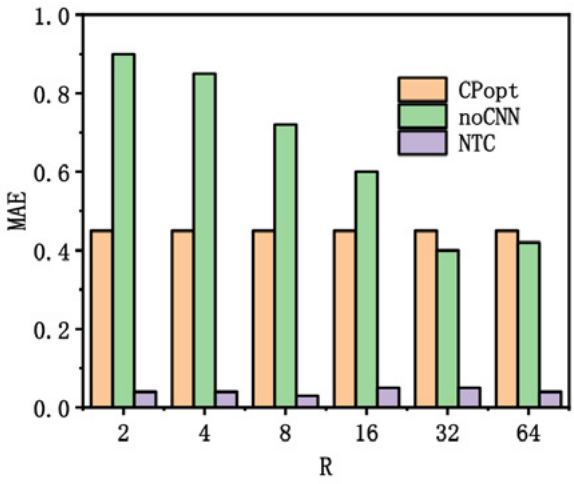
https://doi.org/10.17993/3ctic.2023.121.330-350

3. The performance of the Fu NTC model can surpass the current optimal NTC model.

The influence of latent feature space dimension (R). R is the number of hidden features captured. In the NTC model. R directly affects the size of the 3D interactive mapping tensor, while in the Fu NTC model, r not only affects the extraction of nonlinear features. That is the size of the 3D interactive mapping tensor of the outer product. But also, affects the extraction of linear features. That is the size of the vector formed by the inner product. The performance of the Fu NTC model is better than other contrast algorithms in all kinds of R. It shows that no matter how much R. It is, the fused neural tensor filling model is better than the single (linear or nonlinear) feature extraction model in mining potential features in the data. This shows that. The optimal number of latent features for nonlinear feature extraction is also the best latent feature data for linear feature extraction. It also effectively proves the rationality of sharing embedded layers between nonlinear models. And the linear model in the previous model design.



(a)



(b)

Figure 5. The influence of spatial dimension (R) on potential features

https://doi.org/10.17993/3ctic.2023.121.330-350

The influence of the number of convolution kernels (T). The number of convolution kernels is an important factor affecting the feature extraction performance of the 3D CNN model. NTC model will be affected by hyperparameter t. And Fu NTC model will also be affected by T. Because the nonlinear feature extraction part of the Fu NTC model still adopts the frame structure of 3D CNN. In the experimental algorithm, only Fu NTC, NTC, and not out (special cases of NTC) use 3D CNN to extract information. The recovery performance of the Fu NTC model is better than that of the NTC and no-out algorithm, especially in the comparison between the NTC model and the NTC model. It shows that the NTC model only extracts nonlinear information is not enough, even using the 3D CNN framework. It is very effective for feature extraction. The effectiveness of the tensor filling model is also demonstrated. Similarly, the performance trend of the Fu NTC model increases first and then decreases with the increase of the T value. It indicates that the number of convolution kernels is not better. According to the experimental results, the number of convolution kernels of the Fu NTC model in Abilene is set to t = 32, and that of WS-dream is set to t = 64.

The influence of CNN mode. Different CNN patterns affect the effectiveness of feature extraction. As we have known before. 3D CNN has one more convolution dimension than 2D CNN. It means that 3D CNN can simultaneously extract features from three dimensions of data: source node, target node, and time, while 2D CNN cannot. Experiments on the NTC model and Fu NTC model verify the effect of 3D CNN. In the two models, the recovery performance of 3dcnn is always better than that of 2D CNN. And Fu NTC model has always been better than the NTC model. Whether in 3D CNN mode or 2D CNN mode. It also shows the effectiveness of the Fu NTC model.

In the experimental simulation, the performance of the seven missing data recovery algorithms increases with the increase in sampling rate. In particular, the performance of the Fu NTC model is slightly improved compared with the NTC model. This means that the fusion neural tensor-filling model can extract more features than the neural tensor-filling model. It only extracts nonlinear features. When the sampling rate is 1%. The term of the Fu NTC model is about 0.04 and 0.11. What are the 24 and 8 areas of the best CP. It is based tensor filling algorithm except for the NTC model.

3.4. REAL NETWORK PLATFORM EXPERIMENT

The real network platform and NTC model adopt the same platform. The Fu NTC model is integrated into the network monitoring platform. And the performance of the proposed Fu NTC model is evaluated. It is verified in the real network environment.

The performance of the small network was monitored by NTC for 10 minutes every day. Then all the comparison algorithms and the Fu NTC model are tested. And the recovery performance of the Fu NTC model for missing data is verified through the analysis of the experimental results.

https://doi.org/10.17993/3ctic.2023.121.330-350

4. SUMMARY

Through the new neural network framework, the NTC model and CP inner product model share the embedded layer. And then splice the output of their respective interaction functions to speculate the missing data. The experimental results show that. The fusion neural tensor filling model has better data recovery performance than the traditional tensor filling algorithm and NTC model. In this paper, for the large-scale network monitoring system based on the new sparse network monitoring technology. We make the following contributions to improving the prediction accuracy of unmeasured data

1. In this paper, a 3D interactive mapping tensor is proposed to explicitly show the feature interaction between the source node, target node, and time. The interactive mapping tensor uses the outer product operation to model tensor filling to capture the complex correlation between feature dimensions.
2. To extract hidden features and recover lost data more accurately, this paper proposes a 3D CNN framework based on a 3D interactive mapping tensor. It proves theoretically that. CNN can be used to study the great-order correlation between feature scopes from local scope to global scope.
3. To excerpt the potential features unseen in the network performance statistics more comprehensively. This paper proposes a new fusion neural tensor filling model (funtc). It can extract both nonlinear and linear features.
4. In this paper, comprehensive experiments are carried out on two open actual network checking statistics sets to calculate and verify the efficacy of NTC and Fu NTC. The experimental results show that NTC and Fu NTC can realize better retrieval precision even at a very low selection rate.

5. DATA AVAILABILITY

The data used to support the findings of this study are available from the corresponding author upon request.

6. CONFLICT OF INTEREST

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

REFERENCES

(1) Chen, J., Gao, W., & Wei, K. (2021). **Exact matrix completion based on low rank Hankel structure in the Fourier domain**. *Applied and Computational Harmonic Analysis*, 55(2-3), 301-320.

https://doi.org/10.17993/3ctic.2023.121.330-350

(2) Yadav, S. K., & George, N. V. (2020). **Fast Direction-of-Arrival Estimation via Coarray Interpolation Based on Truncated Nuclear Norm Regularization**. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 67(6), 1206-1210.

(3) Tl, A., Av, B., & Rg, A. (2021). **Ranking recovery from limited pairwise comparisons using low-rank matrix completion**. *Applied and Computational Harmonic Analysis*, 54, 227-249.

(4) Douik, A., & Hassibi, B. (2021). **Low-Rank Riemannian Optimization for Graph-Based Clustering Applications**. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(7), 2558-2572.

(5) Wang, P., & Zhang, J. (2021). **A Singular Value Thresholding Based Matrix Completion Method for DOA Estimation in Nonuniform Noise**. *Journal of Beijing Institute of Technology*, 30(4), 368-376.

(6) Dragomir, R. A., Giannessi, F., & Hull, D. G. (2021). **Quartic First-Order Methods for Low-Rank Minimization**. *Journal of Optimization Theory and Applications*, 189(4), 1216-1248.

(7) Kao, S. Y., Katsumi, S., Han, D., et al. (2020). **Postnatal Expression and Possible Function of RANK and RANKL in the Murine Inner Ear**. *Bone*, 136, 115837.

(8) Zhang, F., Hou, J., Wang, J., et al. (2020). **Uniqueness Guarantee of Solutions of Tensor Tubal-Rank Minimization Problem**. *IEEE Signal Processing Letters*, 27, 540-544.

(9) Wei, H., Qu, Z., Wu, X., et al. (2021). **Decentralized Approximate Newton Methods for Convex Optimization on Networked Systems**. *IEEE Transactions on Control of Network Systems*, 8, 2792-2802.

(10) Dai, S., Jin, M., & Zhang, X. (2020). **Kernel affine projection p-norm (KAPP) filtering under alpha-stable distribution noise environment**. *International Journal of Pattern Recognition and Artificial Intelligence*, 34(2), 2059006.

(11) Yang, Y., Yang, Z., Li, J., et al. (2020). **Foreground-Background Separation via Generalized Nuclear Norm and Structured Sparse Norm Based Low-Rank and Sparse Decomposition**. *IEEE Access*, 8, 222957-222970.

(12) Qiao, W., & Yang, Z. (2020). **An Improved Dolphin Swarm Algorithm Based on Kernel Fuzzy C-Means in the Application of Solving the Optimal Problems of Large-Scale Function**. *IEEE Access*, 8, 2073-2089.

(13) Ahn, H. S., Park, S. H., & Ye, J. C. (2020). **Quantitative susceptibility map reconstruction using annihilating filter-based low-rank Hankel matrix approach**. *Magnetic Resonance in Medicine*, 83(3).

(14) Huang, Y., Zhang, X., Guo, H., et al. (2020). **Phase-constrained reconstruction of high-resolution multi-shot diffusion weighted image**. *Journal of Magnetic Resonance*, 312, 106690-.

(15) Arridge, S., Fernsel, P., & Hauptmann, A. (2020). **Joint Reconstruction and Low-Rank Decomposition for Dynamic Inverse Problems.**

(16) Huang, C., & Yang, C. F. (2020). **An Empirical Approach to Minimize Latency of Real-Time Multiprocessor Linux Kernel**. *In 2020 International Computer Symposium (ICS).*

https://doi.org/10.17993/3ctic.2023.121.330-350

(17) Yang, H. J., Lei, Y. X., Wang, J., et al. (2022). **Tensor decomposition based on the potential low-rank and p-shrinkage generalized threshold algorithm for analyzing cancer multiomics data**. *Journal of Bioinformatics and Computational Biology*, 20(02).

(18) Saeedi, T., & Rezghi, M. (2020). **A Novel Enriched Version of Truncated Nuclear Norm Regularization for Matrix Completion of Inexact Observed Data**. *IEEE Transactions on Knowledge and Data Engineering*, PP (99), 1-1.

(19) Zhang, C. J., & Qin, Y. U. (2020). **Spatiotemporal RPCA algorithm for moving target detection in complex scene**. *Computer Engineering and Design.*

(20) Zhao, H., & Zheng, S. (2020). **Joint Extreme Channels-Inspired Structure Extraction and Enhanced Heavy-Tailed Priors Heuristic Kernel Estimation for Motion Deblurring of Noisy and Blurry Images**. *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences*, E103.A(12), 1520-1528.

(21) Zhang, X., Zheng, J., Yan, Y., et al. (2019). **Joint Weighted Tensor Schatten p-Norm and Tensor lp-Norm Minimization for Image Denoising**. *IEEE Access*, 1-1.

(22) Chen, W. G., & Peng, L. I. (2019). **Truncated sparse approximation property and truncated q-norm minimization**. *Applied Mathematics: A Journal of Chinese Universities.*

(23) Chen, J., Lin, Z., Ren, J., et al. (2020). **Distributed multi-scenario optimal sizing of integrated electricity and gas system based on ADMM**. *International Journal of Electrical Power & Energy Systems*, 117, 105675-.

(24) Rava, R. A., Mokin, M., Snyder, K. V., et al. (2020). **Performance of angiographic parametric imaging in locating infarct core in large vessel occlusion acute ischemic stroke patients**. *Journal of Medical Imaging*, 7(1), 1.

(25) Liu, J., He, D., Zeng, X., et al. (2019). **ManiDec: Manifold Constrained Low-Rank and Sparse Decomposition**. *IEEE Access*, PP(99), 1-1.

(26) Vitali, V., Chevallier, F., Jinaphanh, A., et al. (2021). **Eigenvalue separation and eigenmode analysis by matrix-filling Monte Carlo methods**. *Annals of Nuclear Energy*, 164, 108563-.

(27) Liu, Y., Zhou, D., Nie, R., et al. (2020). **Construction of High Dynamic Range Image Based on Gradient Information Transformation**. I*ET Image Processing*, 14(6).

(28) Xue, Z., Dong, J., Zhao, Y., et al. (2019). **Low-rank and sparse matrix decomposition via the truncated nuclear norm and a sparse regularizer**. *The Visual Computer*, 35(11), 1549-1566.

(29) Guo, M.-H., Xu, T.-X., Liu, J.-J., Liu, Z.-N., Jiang, P.-T., Mu, T.-J., Zhang, S.-H., Martin, R. R., Cheng, M.-M., & Hu, S.-M. (2022). **Attention mechanisms in computer vision: A survey**. *Computational Visual Media*, 8(03), 331-368.

(30) Battisti, R., Claumann, C. A., Manenti, F., Machado, R. A. F., & Marangoni, C. (2021). **Machine learning modeling and genetic algorithm-based optimization of a novel pilot-scale thermosyphon-assisted falling film distillation unit**. *Separation and Purification Technology*, 259.

https://doi.org/10.17993/3ctic.2023.121.330-350

(31) Chung, Y., Lee, S., & Kim, W. (2021). **Latest Advances in Common Signal Processing of Pulsed Thermography for Enhanced Detectability: A Review**. *Applied Sciences*, 11(24).

(32) Pekedis, M., Ozan, F., Koyuncu, S., & Yildiz, H. (2022). **The finite element method-based pattern recognition approach for the classification of patient-specific gunshot injury**. *Proceedings of the Institution of Mechanical Engineers. Part H, Journal of Engineering in Medicine*, 236(5).

(33) Dewani, A., Memon, M. A., & Bhatti, S. (2021). **Development of computational linguistic resources for automated detection of textual cyberbullying threats in Roman Urdu language**. *3C TIC. Cuadernos de Desarrollo Aplicados a las TIC*, 10(2), 101-121. https://doi.org/10.17993/3ctic.2021.102.101-121

(34) Mei, D. (2022). **What does students' experience of e-portfolios suggest**. *Applied Mathematics and Nonlinear Sciences*, 7(2), 15-20. https://doi.org/10.2478/AMNS.2021.2.00166/10.2478/AMNS.2021.2.00166

https://doi.org/10.17993/3ctic.2023.121.330-350