International Journal of Security and Its Applications
Vol. 10, No. 7 (2016) pp.249-258
http://dx.doi.org/10.14257/ijsia.2016.10.7.22

Novel Intrusion Detection Method based on Triangular Matrix Factorization

QI Yingchun[1] and *NIU Ling*[2]

[1]*Zhoukou Normal University, Zhoukou Henan Province, 466001, China*
E-mail: Qiyingchun@163.com
[2]*Zhou Kou Normal University, Zhoukou 466001, China;*
Niuling@zknu.edu.cn

Abstract

In order to deal with the issue of network attacks and enhance the security of the network environment, intrusion detection is gaining more and more attention all over the world. In this paper, a novel intrusion detection method based on improved triangular matrix factorization is presented. As a type of famous mathematical tool, triangular matrix factorization has a good ability to reduce the large amount of high dimensional data. However, the traditional triangular matrix factorization has its inherent drawbacks such as the difficulty of setting the parameter adaptively, so the model of an improved version of triangular matrix factorization together with its concrete algorithm is proposed in this paper firstly. Then, improved triangular matrix factorization is employed to convert the high dimensional data of the network into low dimensional vectors of several matrices, with which the anomaly detection can be realized. Experimental results indicate that the proposed method is promising, and it does significantly enhance the detection accuracy and computational efficiency compared with other current popular ones.

Keywords*: intrusion detection, triangular matrix factorization, network environment, dimensionality*

1. Introduction

The rapid development of the network has greatly enhanced and changed the human life styles, and more and more people can freely share the information via the network all over the world. However, the unrestrained and explosive use also leads to the emergence of the side effect, and the attack on the network has already become an urgent security issue [1]. Although there already exist many kinds of protection mechanisms in our computers such as anti-virus software and firewalls existing in the operation system (OS) or requiring installation by ourselves, the update rate of the feature database in each mechanism almost always lags behind that of the new intrusion ways, so that they cannot identify all of the anomaly intrusion in time. What is worse, many advanced hackers can still successfully breach the existing protections to steal our personal privacy or monitor the actions of users. A corresponding research report demonstrated that the number of illegal code signature increased by more than 256% over the previous year [2].

Based on the above current situation, how to design a more effective and intelligent intrusion detection system (IDS) is increasingly becoming an urgent task for us. Generally speaking, the basic idea of intrusion detection can be mainly classified into two categories including signature-based detection [3] and anomaly detection [4]. The former is also called pre-knowledge or active detection method. Firstly, the feature database is pre-defined which covers the main features of the current typical anomaly attacks. Then, the intrusion detection mechanism extracts the feature of each suspicious hostile intrusion and compares it with the above feature database available to find whether there exist similarities between them. Finally, if the above answer is 'Yes', the suspicious access is

ISSN: 1738-9976 IJSIA

considered to be anomaly attacks, else it will be regarded as a normal access. Obviously, signature-based detection is applicable for detecting known attacks, and its detection accuracy is dependent of the completeness of the pre-defined feature database. However, it is incapable of identifying other unknown or recently emerging attacks, so that it may leads to a large number of missing alarms. Unlike the signature-based detection, anomaly detection is based on the adverse concept and is often called passive detection method. The pre-defined feature database stores the fundamental feature of the normal behaviors, so if the feature of the suspicious access is not consistent with the one in the pre-defined database, the access is considered to be hostile. In spite of the high sensitivity to unknown types of network attacks, we have to admit the truth that the anomaly detection is prone to resulting in so many false alarms.

To date, a variety of algorithms which combines several theories in other research domains for intrusion detection have been devised and proposed. For example, support vector machine (SVM) [5-7] is utilized to conduct the intrusion detection. Wang *et. al.*, attempted to deal with the issue of intrusion detection based on artificial neural network (ANN) [8]. With the development of the uncertainty theory, fuzzy sets [9] and rough sets [10] are investigated to conduct intrusion detection. Besides, self-organizing maps (SOM) [11, 12] and principal component analysis (PCA) [13, 14] also belong to the current typical methods. In spite of a lot of relevant intrusion detection models, how to select the features is always an essential issue we have to face. As known to us, the purpose on feature selecting focuses on two points. One is to cut the training and predicting time as much as possible, and the other is to eliminate the data redundancy and irrelevancy. According to [15], the mainstreamed typical feature selecting methods include filter method and wrapper method [3, 16] which are based on retaining features and removing features respectively.

As to anomaly detection, how to deal with the high dimensionality data from an actual computer system is an onerous task for the design course of the intrusion detection. Take the conventional ftp calls for example; more than one million system calls may emerge in a short time. Thus, fast processing of a large amount of high-dimensional data is crucial to build a real-time intrusion detection model so that an intrusion can be detected before substantial damage to the computer system is done. An efficient method able to process a massive amount of data is required for building an intrusion detection model [17].

Triangular matrix factorization [18, 19] is a recently developed matrix analysis algorithm, which can not only describe the low-dimensional intrinsic structures in the high-dimensional space, but achieve the linear representation of the original sample data by imposing the non-negativity constraints on its bases and coefficients. In recent years, more and more improved matrix factorization models have been proposed such as the local matrix factorization [20], the sparse matrix factorization [21], and the weighted matrix factorization [22]. However, the values of W and H are commonly initialized at random in the basic matrix factorization model which leads to the result and computational cost often greatly vary from one to one.

The purpose of this paper is to develop a novel intrusion detection based on an improved matrix factorization model. Improved matrix factorization which is an extensive version of the traditional matrix factorization is designed and presented firstly to overcome the drawback of random initialization of its two low-dimensional matrices. Then, improved matrix factorization is employed to reduce the high-dimensional data vectors and achieve anomaly detection in low dimensions. Several typical testing data sets are utilized in this paper, and the experiments results indicate that the proposed method is superior to other current typical ones in terms of detection accuracy and computational costs.

The remainder of this paper is organized as follows. In Section 2, the proposed improved triangular matrix factorization model is presented. Some experiments are conducted to evaluate and test the superiority of the proposed method, and their results,

together with relevant discussions, are reported in Section 3. Conclusions and future work are summarized in the end.

2. Proposed Improved Triangular Matrix Factorization

In order to describe the augmented matrix factorization model conveniently in the context, it is necessary to review the basic content of triangular matrix and several relevant conclusions of Singular Value Decomposition (SVD) [23, 24].

In the mathematical discipline of linear algebra, a triangular matrix is a special kind of square matrix. A square matrix is called lower triangular if all the entries above the main diagonal are zero. Similarly, a square matrix is called upper triangular if all the entries below the main diagonal are zero. A triangular matrix is one that is either lower triangular or upper triangular. A matrix that is both upper and lower triangular is called a diagonal matrix.

Because matrix equations with triangular matrices are easier to solve, they are very important in numerical analysis. By the LU decomposition algorithm, an invertible matrix may be written as the product of a lower triangular matrix L and an upper triangular matrix U if and only if all its leading principal minors are non-zero.

A matrix of the form

$$L = \begin{bmatrix} l_{1,1} & & & 0 \\ l_{2,1} & l_{2,2} & & \\ l_{3,1} & l_{3,2} & \ddots & \\ \vdots & \vdots & \ddots & \ddots \\ l_{n,1} & l_{n,2} & \cdots & l_{n,n-1} \quad l_{n,n} \end{bmatrix}$$

is called a lower triangular matrix or left triangular matrix, and analogously a matrix of the form

$$U = \begin{bmatrix} u_{1,1} & u_{1,2} & u_{1,3} & \cdots & u_{1,n} \\ & u_{2,2} & u_{2,3} & \cdots & u_{2,n} \\ & & \ddots & \ddots & \\ & & & \ddots & u_{n-1,n} \\ 0 & & & & u_{n,n} \end{bmatrix}$$

is called an upper triangular matrix or right triangular matrix. The variable L (standing for lower or left) is commonly used to represent a lower triangular matrix, while the variable U (standing for upper) or R (standing for right) is commonly used for upper triangular matrix. A matrix that is both upper and lower triangular is diagonal.

Matrices that are similar to triangular matrices are called triangularisable.

Many operations on upper triangular matrices preserve the shape:

The sum of two upper triangular matrices is upper triangular.

The product of two upper triangular matrices is upper triangular.

The inverse of an invertible upper triangular matrix is upper triangular.

The product of an upper triangular matrix by a constant is an upper triangular matrix.

Together these facts mean that the upper triangular matrices form a sub algebra of the associative algebra of square matrices for a given size. Additionally, this also shows that the upper triangular matrices can be viewed as a Lie sub algebra of the Lie algebra of square matrices of a fixed size, where the Lie bracket [a, b] given by the commutator ab-ba. The Lie algebra of all upper triangular matrices is often referred to as a Borel sub algebra of the Lie algebra of all square matrices.

All these results hold if "upper triangular" is replaced by "lower triangular" throughout; in particular the lower triangular matrices also form a Lie algebra. However, operations mixing upper and lower triangular matrices do not in general produce triangular matrices. For instance, the sum of an upper and a lower triangular matrix can be any matrix; the product of a lower triangular with an upper triangular matrix is not necessarily triangular either.

Theorem 1 Let $A \in C_r^{m\times n}(r>0)$, there exists a rank-*m* unitary matrix *U* and a rank-*n* unitary matrix *V* satisfy the following equation:

$$A = U\begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} V^T \tag{1}$$

Where $\Sigma=\mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r)$.

Lemma 1.1 σ_i, which meets the requirement that $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r$, is referred to as the positive singular value of matrix *A*, where $i=1, 2, \ldots, r$.

Lemma 1.2 The singular value σ_i is uniquely decided by matrix *A*, whereas the corresponding unitary matrices *U* and *V* are not commonly single, accordingly, the factorization equation as Eq. (7) is usually not single either.

Lemma 1.3 The number of the positive singular values of matrix *A* is equal to rank(*A*), which is the rank of matrix *A*, moreover, there exists $\mathrm{rank}(A)\leq\min(m, n)$.
According to Lemma 1.1, the original matrix *A* can be rewritten as:

$$\begin{aligned} A &= \sum_{i=1}^{r}\sigma_i u_i v_i^T = \sum_{i=1}^{r} u_i \sigma_i v_i^T = \sum_{i=1}^{r}\sqrt{\sigma_i}u_i(\sqrt{\sigma_i}v_i)^T \\ &= (\sqrt{\sigma_1}u_1)(\sqrt{\sigma_1}v_1)^T + (\sqrt{\sigma_2}u_2)(\sqrt{\sigma_2}v_2)^T + \ldots + (\sqrt{\sigma_r}u_r)(\sqrt{\sigma_r}v_r)^T \\ &= (\sqrt{\sigma_1}u_1, \sqrt{\sigma_2}u_2, \ldots, \sqrt{\sigma_r}u_r)\cdot((\sqrt{\sigma_1}v_1)^T, (\sqrt{\sigma_2}v_2)^T, \ldots, (\sqrt{\sigma_r}v_r)^T)^T \end{aligned} \tag{2}$$

Where the sizes of each column vector $\sigma_i^{1/2}u_i$ of matrix *U* are all $m\times 1$, furthermore, the row vector $(\sigma_i^{1/2}v_i)^T$, whose sizes are all $1\times n$, can be obtained via the transposition of the product between the row vector v_i of matrix *V* and its corresponding square root of the singular value σ_i. Consequently, if we substitute matrices *B* and *C* for the above column vectors and row vectors respectively, Eq. (8) can be further modified as:

$$A_{m\times n} = B_{m\times r}C_{r\times n} \tag{3}$$

Clearly, compared with Eq. (1), Eq. (3) strongly resembles with it in two aspects of the form.

(a) Any matrix *A* can be represented by the product of two relatively low-dimensional matrices *B* and *C* by using SVD or matrix factorization;

(b) Matrices *B* and *C* are not commonly single.

Nevertheless, there is still an essential difference between SVD and matrix factorization, which lies in that matrix factorization, imposes non-negativity constraints on the two low-dimensional matrices *B* and *C*, but SVD doesn't advance any requirements to them at all. Therefore, efforts can be exerted to explore the intrinsic relation of the algorithms SVD and matrix factorization from the non-negativity point of view. Eq. (2) can be rewritten as:

$$\begin{aligned} A &= \sqrt{\sigma_1}u_1(\sqrt{\sigma_1}v_1)^T + \ldots + \sqrt{\sigma_k}u_k(\sqrt{\sigma_k}v_k)^T + \ldots + \sqrt{\sigma_r}u_r(\sqrt{\sigma_r}v_r)^T \\ &= A_1 + \ldots + A_k + \ldots + A_r \end{aligned} \tag{4}$$

Where $1 \le k \le r$. Eq. (4) indicates that any matrix A can always be represented as the sum of several nonzero sub-matrices of the same size, each of which corresponds to a nonzero singular value of A, furthermore, the number of sub-matrices is the same as that of nonzero singular values. Inspired by reference [24], we can deal with each sub-matrix like this:

$$\begin{aligned} &A_k = A_k^+ - A_k^- \\ &A_k^+(i,j) = \begin{cases} A_k(i,j), & if \quad A_k(i,j) \ge 0; \\ 0, & else \end{cases} \\ &A_k^-(i,j) = \begin{cases} 0, & if \quad A_k(i,j) \ge 0; \\ -A_k(i,j), & else \end{cases} \end{aligned} \tag{5}$$

Where $1 \le i \le m$, $1 \le j \le n$.

In Eq. (5), each nonzero sub-matrix A_k is transformed into the difference between two constructive nonnegative matrices A_k^+ and A_k^-. Based on the above, the modification of Eq. (11) from the column vector point of view is given as follows:

$$\begin{aligned} A_k &= (\sqrt{\sigma_k}u_k)(\sqrt{\sigma_k}v_k)^T \\ &= [(\sqrt{\sigma_k}u_k)^+ - (\sqrt{\sigma_k}u_k)^-][(\sqrt{\sigma_k}v_k)^+ - (\sqrt{\sigma_k}v_k)^-]^T \\ &= [(\sqrt{\sigma_k}u_k)^+((\sqrt{\sigma_k}v_k)^+)^T + (\sqrt{\sigma_k}u_k)^-((\sqrt{\sigma_k}v_k)^-)^T] - \\ &\quad [(\sqrt{\sigma_k}u_k)^+((\sqrt{\sigma_k}v_k)^-)^T + (\sqrt{\sigma_k}u_k)^-((\sqrt{\sigma_k}v_k)^+)^T] \\ &= [(\sqrt{\sigma_k}u_k^+)(\sqrt{\sigma_k}v_k^+)^T + (\sqrt{\sigma_k}u_k^-)(\sqrt{\sigma_k}v_k^-)^T] - [(\sqrt{\sigma_k}u_k^+)(\sqrt{\sigma_k}v_k^-)^T + (\sqrt{\sigma_k}u_k^-)(\sqrt{\sigma_k}v_k^+)^T] \end{aligned} \tag{6}$$

Eq. (6) converts sub-matrix A_k into the subtraction of two groups of nonnegative matrices. In order to facilitate the representation in the context, we rewrite Eq. (6) as:

$$A_k^+ = [(\sqrt{\sigma_k}u_k)^+((\sqrt{\sigma_k}v_k)^+)^T + (\sqrt{\sigma_k}u_k)^-((\sqrt{\sigma_k}v_k)^-)^T] \tag{7}$$

$$A_k^- = [(\sqrt{\sigma_k}u_k)^+((\sqrt{\sigma_k}v_k)^-)^T + (\sqrt{\sigma_k}u_k)^-((\sqrt{\sigma_k}v_k)^+)^T] \tag{8}$$

Note that by using matrix factorization for intrusion detection, we commonly take the condition r=1 into account, since a single eigen basis can be obtained here, which covers the whole properties of the original data matrix. The above proposition has been proved in reference [24]. Moreover, analyzed from the view of SVD, the largest singular value σ_1 and its relevant eigenvectors u_1 and v_1 contain the prime character of the original matrix A, thereby we are allowed to only consider the nonzero sub-matrix A_1 approximately in the course of image fusion, that is:

$$\begin{aligned} A_1 &= [(\sqrt{\sigma_1}u_1)^+((\sqrt{\sigma_1}v_1)^+)^T + (\sqrt{\sigma_1}u_1)^-((\sqrt{\sigma_1}v_1)^-)^T] - \\ &\quad [(\sqrt{\sigma_1}u_1)^+((\sqrt{\sigma_1}v_1)^-)^T + (\sqrt{\sigma_1}u_1)^-((\sqrt{\sigma_1}v_1)^+)^T] \\ &= [(\sqrt{\sigma_1}u_1^+)(\sqrt{\sigma_1}v_1^+)^T + (\sqrt{\sigma_1}u_1^-)(\sqrt{\sigma_1}v_1^-)^T] - [(\sqrt{\sigma_1}u_1^+)(\sqrt{\sigma_1}v_1^-)^T + (\sqrt{\sigma_1}u_1^-)(\sqrt{\sigma_1}v_1^+)^T] \\ &= A_1^+ - A_1^- \end{aligned} \tag{9}$$

$$A_1^+ = [(\sqrt{\sigma_1}u_1^+)(\sqrt{\sigma_1}v_1^+)^T + (\sqrt{\sigma_1}u_1^-)(\sqrt{\sigma_1}v_1^-)^T] \tag{10}$$

Furthermore, due to the non-negativity of the intrusion number in real situations, the constructive nonnegative matrix A_r^+ embodies the majority of energy and information of the original nonzero sub-matrix A_r, which enlightens us to approximately replace A_r with A_r^+. Based on the above discussions, during the course of intrusion detection, we can

improve the basic matrix factorization model by simplifying the matrix A to be A_1^+ and adaptively setting the values of initial vectors W and H with the column ones u_1 and v_1.

By deeply researching on SVD and matrix factorization, we notice that the two approaches can be organically integrated together, which is manifested as that the traditional problem of initialization of W and H can be changed into the analysis of the corresponding column vectors u_1 and v_1. The concrete steps are listed as follows:

Step 1: Compute the L_1-norm of the column vectors u_1^+, u_1^-, v_1^+ and v_1^- as $\|u_1^+\|$, $\|u_1^-\|$, $\|v_1^+\|$ and $\|v_1^-\|$. If any one of them equals zero, add an infinitesimal positive *eps* to it;

Step 2: Implement column normalization by dividing $\|u_1^+\|$, $\|u_1^-\|$, $\|v_1^+\|$, $\|v_1^-\|$ into u_1^+, u_1^-, v_1^+, v_1^- respectively, the results of which are recorded as $(u_1^+)_{\text{norm}}$, $(u_1^-)_{\text{norm}}$, $(v_1^+)_{\text{norm}}$, $(v_1^-)_{\text{norm}}$. This step can effectively avoid the problem of "scaling" during the matrix decomposition;

Step 3: In order to satisfy Eq. (10), compute the norm evaluated coefficients var^+, var^- of the column vectors u_1^+, v_1^+ and u_1^-, v_1^- respectively as the following formulas:

$$\text{var}^+ = sqrt(\| u_1^+ \| \cdot \| v_1^+ \|) \tag{11}$$

$$\text{var}^- = sqrt(\| u_1^- \| \cdot \| v_1^- \|) \tag{12}$$

Step 4: Compare var^+ with var^-, if $\text{var}^+ \geq \text{var}^-$, then var^+, u_1^+, v_1^+ will be chosen as the ultimate vector factors to determine W and H; contrarily, var^-, u_1^-, v_1^- will be treated as the corresponding factors.

The reason for step 4 is that, the larger var^* is, the more energy relevant column vectors u_1^*, v_1^* possess; accordingly, the overall feature of A_1^+ can be illuminated to a more extent. We can modify Eq. (10) like this:

$$A_1^+ = (\sqrt{\sigma_1}\,\text{var}^* u_1^*)(\sqrt{\sigma_1}\,\text{var}^* v_1^*)^T \tag{13}$$

Where the asterisk '*' is used to mark the sign '+' or '-'.

Step 5: Complete the initialization of vectors W_1 and H_1 as follows:

$$\begin{cases} W_1 = (\sqrt{\sigma_1}\,\text{var}^* u_1^*) \\ H_1 = (\sqrt{\sigma_1}\,\text{var}^* v_1^*)^T \end{cases} \tag{14}$$

Step 6: Obtain the L_1-norm of the row vector H_1 as $\|H_1\|$;

Step 7: Achieve the initialization of W and H:

$$\begin{cases} W = W_1 \cdot \| H_1 \| \\ H = H_1 / \| H_1 \| \end{cases} \tag{15}$$

In reference [24], the issue was deeply investigated and a corresponding algorithm was proposed in detail as well. However, the algorithm winded up at the step 5 in this paper, namely regarding the values of W_1 and H_1 as the initial values of W and H. As we all know, if the initial values of W and H are rationally evaluated, it usually results in much less iterations to reach an optimal state; contrarily, it may lead to many more iterations and often a poor convergence state.

Based on the proposed improved matrix factorization model mentioned above, the frequency properties of system calls and commands can be regarded as the program and user behaviors, respectively. In each given data set, the frequencies of individual system calls are calculated and then recorded as an element in the matrix H. Obviously, the sum

of the elements in the matrix H should satisfy the following requirement under ideal conditions.

$$\sum_{i=1}^{r} H_{ij} = 1, \quad j = 1,2,...,m \tag{16}$$

In other words, any block of normal training data is characterized by a single value 1 by adding all the elements in each column vector h of H. This is the normal behavior profiled from the training data set [17].

According to Eq. (16), we can conclude a reasonable line of thinking that if the real system calls correspond to normal behaviors, the sum of the elements in the matrix H should be close to 1. As a result, we can estimate the anomalous extent by analyzing the difference between the sum in the matrix H and 1. If the disparity between them is wide and large, the corresponding data can be considered as anomalous. In this paper, the disparity extent we can tolerate is not more than 0.3.

3. Experimental Results and Related Analysis

In order to verify the effectiveness of the proposed method, a series of simulation experiments are conducted in this section. Two data sets, live lpr system call data from UNM and command data from AT&T lab, are used to test the intrusion detection model developed in this paper. The section mainly consists of two parts as follows. (a) Experimental introduction: this part includes PC hardware conditions, the data sets used, and a brief list of the current popular methods to compare with in the following experiments. (b) Performance evaluation and related analysis: this part is mainly composed of the objective quantitative evaluation, and further analysis and discussions on the simulation results are carried out in this part.

The simulation experiments are conducted on a PC with Intel Core i5/2.3GHz/2G. Two data sets are used to validate the effectiveness of the proposed method. Firstly, the live lpr system call data from UNM including 2703 traces of normal data and 1001 traces of intrusion data is used. In order to facilitate the experiment course, we only choose the first 500 traces of the normal data and the first 500 traces of the intrusion data as the sample. The whole data sets can be downloaded at the relevant website. Secondly [17], another category of data set namely the command data sets collected in AT&T's Shannon Research Laboratory is also used in our experiments. Referenced in [17], the command data consist of user names and the associated command sequences. Fifty users are included with 15000 consecutive commands for each user, divided into 150 blocks of 100 commands. The first 50 blocks are uncontaminated and used as training data. Starting at block 51 and onward, some masquerading command blocks, randomly drawn from outside of the 50 users, are inserted into the command sequences of the 50 users. The goal is to correctly detect the masquerading blocks in the user community. The data used in the experiments are available for downloading at http://www.schonlau.net/intrusion.html for more details of the contamination procedure. The description of the lpr simulation experiment is given in Table. 1.

Table 1. The Variants of the lpr Simulation Experiment

Data set	System calls	Distinct system calls	Normal traces	Abnormal traces
lpr	828937	39	500	500

By conducting the simulation experiments, the results of the proposed method are very satisfactory. Unlike the traditional matrix factorization model, the parameters of the improved matrix factorization are not setting randomly any more. Besides, the dimension

of the improved matrix factorization model declines a lot also. As a result, we don't have to face the problem of setting the parameter r, which mainly relied on a number of experiments or personal experience before. In order to verify the effectiveness and the efficiency of the proposed method, we bring 400 abnormal traces and 400 normal traces into the lpr simulation experiment. The result is given in Table. 2. It can be obviously noticed that the abnormal traces detection ability of the proposed matrix factorization method is very strong, which means that almost each abnormal trace can be found by the proposed matrix factorization method. However, its false alarm rate namely some normal traces may also be mistaken for abnormal ones, and the probability is close to 3.0% according to the experimental statistics.

Table 2. Missing Alarm Rates and False Alarm Rates

Abnormal traces	Normal traces	Missing alarm	False alarm	Missing alarm rate	False alarm rate
400	400	0	12	0.0%	3.0%

Unlike the lpr system call data, the difference between blocks of the command data is relatively large. Several classic methods are used to compare with the proposed matrix factorization model, and the results of the false alarm rates and missing alarm rates are reported in Table. 3. Note that all of the relevant methods are based on the same intrusion samples. Obviously, compared with the former two methods, the proposed matrix factorization model has remarked superiorities in both terms of the missing alarm rate and the false alarm rate. In comparison to PCA, the missing alarm rate of the proposed matrix factorization model is a bit higher, but the performance of its false alarm rate is much better.

Table 3. The Comparison of the Several Intrusion Detection Methods

Methods	Missing alarm rate	False alarm rate
Compression	4.9%	65.2%
Sequence-match	3.8%	62.9%
PCA	1.8%	71.2%
The proposed method	2.9%	49.6%

4. Conclusion

In this paper, a novel intrusion detection method based on improved matrix factorization model is proposed. Compared with the traditional matrix factorization model, the improved matrix factorization model modifies the random parameter setting mode existing in the former, so that the variables W and H can reach the global optimum points with much fewer iteration times to a large extent. Besides, the proposed matrix factorization model has a good ability to reduce the large amount of high dimensional data, and extract the fundamental and important information into the final low dimensional vectors of several matrices, with which the anomaly detection can be realized. Two data sets, live lpr system call data from UNM and command data from AT&T lab, are used to test the effectiveness and efficiency of the proposed matrix factorization model. However, of course, we have to face the true that the proposed matrix factorization model does not have a very satisfactory performance in terms of the missing alarm rate compared with the PCA method for example. Consequently, investigations of the inherent

reasons for the phenomena as mentioned and how to further optimize the performance of the new method will attract our attention and be the emphasis of our future work.

Acknowledgements

The authors thank the anonymous reviewers and editors for their invaluable suggestions. The work was supported in part by the soft science research project of Henan province of China under Grant 142400411133, in part by the Science and technology project of Henan Province of China under Grant 152102210367.

References

[1] H. J. Liao, C. H. R. Lin, Y. C. Lin and K. Y. Tung, "Intrusion detection system: A comprehensive review", Journal of Network and Computer Application, vol. 36, no. 1, **(2013)**, pp. 16-24.
[2] G. C. Tjhai, S. M. Furnell, M. Papadaki and N. L. Clarke, "A preliminary twostage alarm correlation and filtering system using SOM neutral network and Kmeans algorithm",. Computers & Security, vol. 29, no. 4, **(2010)**, pp. 712-723.
[3] S. Mukherjee and N. Sharma, "Intrusion detection using naive Bayes classifier with feature reduction", Procedia Technology, vol. 4, no. 1, **(2012)**, pp. 119-128.
[4] D. E. Denning, "An intrusion-detection model", IEEE Transactions on Software Engineering, vol. 13, no. 2, **(1987)**, pp. 222-232.
[5] J. F. C. Joseph, A. Das, B. S. Lee and B. C. Seet, "CARRADS: Cross layer based adaptive real-time routing attack detection system for MANETS", Computer Networks, vol. 54, no. 7, **(2010)**, pp. 1126-1141.
[6] M. N. Mohammed and N. Sulaiman, "Intrusion Detection System Based on SVM for WLAN", Procedia Technology, vol. 1, no. 1, **(2012)**, pp. 313-317.
[7] S. Seongjun, L. Seungmin, K. Hyunwoo and K. Sehun, "Advance probabilistic approach for network intrusion forecasting and detection", Expert Systems with Applications, vol. 40, no. 1, **(2013)**, pp. 315-322.
[8] G. Wang, J. Hao, J. Ma and L. Huang, "A new approach to intrusion detection using artificial neutral networks and fuzzy clustering", Expert Systems with Applications, vol. 37, no. XX, **(2010)**, pp. 6225-6232.
[9] P. A. R. Kumar and S. Selvakumar, "Detection of distributed denial of service attacks using an ensemble of adaptive and hybrid neuro-fuzzy systems", Computer Communications, vol. 36, no. 3, **(2013)**, pp. 303-319.
[10] S. Nandita, S. Jaydeep, S. Jaya and S. Moumita, "Designing of online intrusion detection system using rough set theory and Q-learning algorithm", Neurocomputing, vol. 111, no. 2, **(2013)**, pp. 161-168.
[11] W. Wang, X. Guan, X. Zhang and L. Yang, "Profiling program behavior for anomaly intrusion detection based on the transition and frequency property of computer audit data", Computer Security, vol. 25, no. 7, **(2006)**, pp. 539-550.
[12] G. C. Tjhai, S. M. Furnell, M. Papadaki and N. L. Clarke, N. L, "A preliminary twostage alarm correlation and filtering system using SOM neutral network and Kmeans algorithm", Computers & Security, vol. 29, no. 4, **(2010)**, pp. 712-723.
[13] W. Wang, X. Guan and X. Zhang, "Processing of massive audit data streams for real-time anomaly intrusion detection", Computer Communication, vol. 31, no. 1, **(2008)**, pp. 58-72.
[14] J. Arunna, Z. Tan, X. He, N. Priyadarsi and R. P. Liu, "RePIDS: A multi tier realtime payload-based intrusion detection system", Computer Networks, vol. 57, no. 3, **(2013)**, pp. 811-824.
[15] B. Luo and J. B. Xia, "A novel intrusion detection system based on feature generation with visualization strategy", Expert Systems with Applications, vol. 41, no. XX, **(2014)**, pp. 4139-4147.
[16] Y. Li, J. Xia, S. Zhang, J. Yan, X. Ai and K. Dai, "An efficient intrusion detection system based on support vector machines and gradually feature removal method", Expert Systems with Applications, vol. 39, no. XX, **(2012)**, pp. 424-430.
[17] X. H. Guan, W. Wang and X. L. Zhang, "Fast intrusion detection based on a matrix factorization model", Journal of Network and Computer Applications, vol. 32, no. XX, **(2009)**, pp. 31-44.
[18] D. D. Lee and H. S. Seung, "Learning the parts of objects with nonnegative matrix factorization", Nature, vol. 401, no. 3, **(1999)**, pp. 788-791.
[19] D. D. Lee and H. S. Seung, "Algorithms for nonnegative matrix factorization", Advances in Neural Information Processing Systems, vol. 13, no. 2, **(2001)**, pp. 556-562.
[20] I. Buciu and I. Pitas, "NMF, LNMF, and DNMF modeling of neural receptive fields involved in human facial expression perception", Journal of Visual Communication and Image Representation, vol. 17, no. 5, **(2006)**, pp. 958-969.
[21] O. Samko, P. L. Rosin and A. D. Marshall, "Robust Automatic Data Decomposition Using a Modified Sparse NMF", Lecture Notes in Computer Science, vol. 4418, no. 1, **(2007)**, pp. 225-234.

[22] D. Guillamet, J. Vitria and B. Scheile, "Introducing a weighted matrix factorization for image classification", Pattern Recognition Letters, vol. 24, no. 14, **(2003)**, pp. 2447-2454.
[23] K. Konsstantinides and K. Yao, "Statistical analysis of effective singular values in matrix rank determination", IEEE Transactions on Acoustics, Speech, and Signal Procesing, vol. 36, no. 5, **(1988)**, pp. 757-763.
[24] C. Boutsidis and E. Gallopoulos, "SVD based initialization: A head start for nonnegative matrix factorization", Pattern Recognition, vol. 41, no. 4, **(2008)**, pp. 1350-1362.
[25] S. Forrest, S. A. Hofmeyr, A. Somayaji and T. A. Longstaff, "A sense of self for Unix processes", In: Proceedings of the 1996 IEEE symposium on research in security and privacy, Los Alamos, CA, **(1996)**, pp. 120-128.
[26] C. Warrender, S. Forrest and B. Pearlmutter, "Detecting intrusions using system calls: alternative data models", In: Proceedings of the 1999 IEEE symposium on security and privacy, **(1999)**, pp. 133-145.

Authors

Qi Yingchun, born on Jan. 10, 1963, Henan Province, China. Current position, grades: Associate Professor of Zhoukou Normal University.University studies: computer application technology. Scientific interest: network of computer

Niu Ling, received the B.Eng degree in Computer science from Henan normal university and M.Eng degree in Computer science from Chengdu University of Technology. She is currently researching on computer application technology.

