Comparison of Data Dimension Reduction Methods in The Problem of Detecting Attacks

Trang - Linh Le Thi *, Van-Truong Nguyen **, Trong - Minh Hoang ***, Quang - Huy Dinh***
* Information technology faculty, Electric Power University, Hanoi, Viet Nam
** VNPT Thua Thien Hue, Hue, Vietnam
***Telecommunication Faculty, Posts and Telecommunications Institute of Technology, Hanoi, Viet Nam
linhltt@epu.edu.vn, huydq.B17VT167@stu.ptit.edu.vn, hoangtrongminh@ptit.edu.vn

***Abstract*—Data dimension reduction issue is an important problem in the data pre-processing stage of data intelligent computing systems. The performance of data dimension reduction methods not only ensure compatibility with machine learning techniques, but also improve data processing efficiency. However, the performance of a dimensional reduction processing method in a data set is always an open challenging issue since it is closely tied to the data features. This paper presents the results of comparing the performance of several approaches in two common approaches on the UNSW-NB 15 data set for attack detection. Our experimental results show that RF-MLP method is very effective for deploying IDSs against DOS attacks.**

***Index Terms*—Data reduction, PCA, MLP, IDS, UNSW-NB15 data set**

I. Introduction

Attack detection in computer networks has always been a challenge faced by security administrators. Intrusion Detection Systems (IDS) [1] have become a primary choice and a popular tool for identifying anomalous and malicious activities in computer systems and networks. Currently, intrusion detection systems often apply machine learning techniques such as neural networks, CMAC, SVM, Fuzzy logic and many others. Among them, some neural networks such as CMAC restrict the number of input dimensions or machine learning methods do not work effectively when the number of input vectors is too large [1], [2], so before applying these networks, it is necessary to solve the problem of data dimensionality reduction.

Currently, there are two main approaches to reduce the data dimension: The first is to find the combination of features to give new features, this direction has some following methods[3], [4]: Principal Component Analysis (PCA), Factor Analysis (FA), Linear Discriminant Analysis (LDA), Truncated Singular Value Decomposition (SVD), t-distributed Stochastic Neighbor Embedding (t-SNE), Multidimensional Scaling (MDS), Isometric mapping (Isomap), etc. The second direction is to find the most important features and remove unnecessary feature of the original data set, this include Backward elimination [5], Forward selection [6].

In this paper, we apply both approaches to reduce data dimensional. For the first direction, we use PCA method, and in the second direction, we propose a new feature extraction method RF-MLP based on the combination of two methods Random Forest and MLP. The comparison results of the two methods were tested on the UNSW NB 15 attack detection data set [7]. The selected feature are tested to build an attack detection system based on the CMAC neural network.

II. Data Dimension Reduction Methods

A. Principal Component Analysis (PCA)

Principal Component Analysis (PCA) uses a linear transform method to map data from a high-dimensional space to a new space with fewer dimensions. The goal of the PCA method is to reduce the dimensionality of a set of vectors so that the most important information can be preserved. PCA can be thought of as a method of finding an orthonormal basis system that acts as a rotation, such that in this new basis the variance in some dimensions is very small, and we can ignore it. . The steps of the PCA algorithm are as follows[8]:

Step 1: We have x_D where D = 1, 2 ..., N are random n-dimensional input data records with mean $\bar{x}$, the mean value is defined by the following formula :

$$\bar{x} = \frac{1}{N}\sum_{n=1}^{N} x_n \tag{1}$$

Step 2: Calculate the covariance matrix of x_D:

$$S = \frac{1}{N}\sum_{D=1}^{N} (x_D - \bar{x}).(x_D - barx)^T \tag{2}$$

Step 3: Calculate the eigenvalues of covariance matrix:

$$v_i = \lambda_i v_i \tag{3}$$

Where $\lambda_i (i = 1, 2, ..., D)$ are the eigenvalues and $v_i (i = 1, 2, ..., D)$ are the eigenvalues, respectively.

Step 4: To represent the data records using low-dimensional vectors, simply compute the K eigenvectors (called the largest directions) corresponding to those K largest eigenvectors $(K < D)$. The variance of the projections of the input data onto the principal direction is larger than the variance of any other direction.

We have:

$$\Lambda = diag[\lambda_1, \lambda_2, \ldots, \lambda_K],\ \phi = [v_1,\ v_2, \ldots, v_K] \tag{4}$$

Where Λ is a square diagonal matrix with eigenvalues $\lambda_i (i = 1, 2, ..., K)$ lying on the main diagonal, ϕ is a matrix of

corresponding eigenvalues.
We have:

$$S\phi = \phi\Lambda \tag{5}$$

The parameter v represents the approximate accuracy of K largest eigenvectors so that the following relation remains the same.

$$\frac{\sum_{i=1}^{K}\lambda_i}{\sum_{i=1}^{D}\lambda_i} \geq \ v \tag{6}$$

Step 5: Based on (5) and (6) the number of symbols that can be chosen, and for an exact parameter v, the new input data low-dimensional feature vector x is determined by:

$$x_f = \ \phi^T x \tag{7}$$

B. Random forest - Multilayer Perceptron method

The RF-MLP feature feature selection method is built by combining the Random forest (RF) method and the Multilayer Perceptron (MLP) neural network [9]. Start selecting features by using RF algorithm to determine Gini impurity, the most important feature will correspond to the highest Gini impurity. For the data set X containing n classes, the Gini impurity index is determined by the following formula [10]:

$$i(X) = 1 - \sum_{i=1}^{n} p_i^2 \tag{8}$$

Where p_i is the percentage of each class. After dividing the set X with A features into 2 smaller data sets and with the corresponding quantity and . The Gini index is determined by the following formula:

$$i_A(X) = \frac{N_1}{N} i(X_1) + \frac{N_2}{N} i(X_2) \tag{9}$$

According to formula (8): $i(X_1) = 1 - \sum\limits_{i=1}^{n} p_i^2(X_1)$, and $i(X_2) = 1 - \sum\limits_{i=1}^{n} p_i^2(X_2)$.
The best feature is the feature with the value $\Delta i(A) = i(X) - i_A(X)$ reach highest. After identifying important features, the order of features will be rearranged in descending order of Gini index. feature sets with a descending number of 41 or less will be progressively reduced one by one and fed into the MLP network to compare the results using 42 features.

If the result when using the reduced set of features is higher or equal to the result when using 42 features, it will proceed to reduce 1 feature. The feature reduction process will end when the result of applying the number of reduced features is lower than the result when using 42 features. The process of selecting the features of the RF-MLP method is described as follows:

Algorithm 1 Features selecting

procedure FIND FEATURES SET
  $i := [1, 2..., 9]$, (i: the number of features in the feature set F)
  $j := \left[1, ...c_9^i\right]$, ($j$: numerical order in i)
  for $i = 9 \rightarrow 1$ **do**
    Set threshold $\vartheta := [0.1, ..., 0.9]$ and step parameter = 0.01
    Set B_{ij}: experimental value, A_{42}: original MLP value
    if $B_{\text{ij}} < A_{42}$ **then return End**
    else Select a feature set with the highest accuracy
    $i := i - 1$.
  End

III. DATASET UNSW-NB 15

Currently, there are various attack detection datasets available: DARPA 98 [11], KDD Cup 99 [12], NSL KDD [13], PESIM 2005 [14], ADFA Intrusion Detection Data Set [15], UNB ISCX Intrusion Detection Evaluation Data Set [16], University of New Mexico (UNM) Data Set [17], HTTP DATASET CSIC 2010 [18], UNSW-NB 15. Of the datasets listed, most have been exported has been around for a long time, does not contain these new attack data types. In this experiment, we choose UNSW-NB15, one of the newer attack datasets.
This dataset was developed in 2015 using IXIA PerfectStorm to create a combination of modern standard attacks on network traffic. The tcpdump tool was used to collect 100 GB of raw network traffic. Each pcap file contains 1000 MB for easier packet analysis. Argus, Bro-IDS and 12 procedures are executed in parallel to generate 44 features for each attack type. This dataset contains 2,540,044 records stored in four CSV files. After removing duplicate records, the number of remaining records is 2059419, all records are split into 4 files containing only data about common information and corresponding attack types below:

- Normal: normal transaction processes.
- Fuzzers: an attack that injects large amounts of randomly generated data into a program or network.
- Analysis: includes different types of attacks such as port scanning, spamming and text scripts (HTML files).
- Backdoor: is a form of attack in which the system's authentication mechanism is invisibly bypassed so that an attacker can gain unauthorized access to the system.
- DoS: Denial of Service attack.
- Exploits: attacks exploit software program errors, system vulnerabilities leading to unexpected server or network problems.
- Generic: This attack is against all block ciphers and uses hash functions to cause collisions without knowing the block cipher structure.
- Reconnaissance: this type of attack will collect all information about the computer network to bypass the security systems of that computer network.
- Shellcode: An attacker uses a piece of malicious software containing code to control the compromised computer.
- Worms: depending on the security errors of the computer that it penetrates to access the network, the worm will copy itself and use the computer network to spread to other machines.

The number of above attack types in the UNSW-NB 15 dataset

TABLE I
TYPES OF ATTACKS OF UNSW-NB15 DATASET

Types of attacks	Number of record
Normal	1959775
Reconnaissance	13357
Backdoor	1983
DoS	5665
Exploits	2799
Analysis	2184
Fuzzers	21795
Worms	171
Shellcode	1511
Generic	25378

is listed in Table 1. Each record includes 44 features of network traffic belonging to five value categories: identifier, integer, real number, time time, binary, where the last two features contain information about the attack type of each record.

IV. EXPERIMENTAL RESULTS

A. Results using PCA method

The training of the MLP network was performed on 4412 DoS attack records and 126485 non-DoS attack records (80% of the UNSW-NB15 dataset). The number of layers of the MLP network is as follows: 15-10-1, 30-20-1, 50-30-1, 100-50-1, 30-20-10-1, during training using the activation function sigmoid. The test was performed on 1103 DoS attack records and 31616 non-DoS attack records (20% of the records from the UNSW-NB15 dataset).

TABLE II
TEST RESULTS USING PCA METHOD

No	Number of features used	MLP network test results	
		DoS (%)	Not Dos (%)
2	40	0.842	0.829
3	20	0.691	0.601
4	6	0.540	0.490

B. RF-MLP method results

After applying RF-MLP method, 9 features with the highest Gini impurity were obtained: Proto, Service, Sttl, Dttl, Synack, Smeansz, Ct_srv_src, Ct_state_ttl, Ct_srv_dst. Records from the UNSW-NB 15 dataset with the 9 features selected above were fed into the MLP network to compare the results with the MLP network using 42 features. If the result when using 9 features is higher or equal to the result when using 42 features, it will proceed to reduce 1 feature. feature reduction will end when the result of applying the number of reduced features is lower than the result of using 42 features.

The training of the MLP network was performed on 4412 DoS attack records and 126485 non-DoS attack records (80% of the UNSW-NB15 dataset). The input of MLP will use 42 features of all attack types, the algorithms used are: trainlm, traindx, trainscg, trainbfg. During learning and testing use 3 classes (15-10-1, 30-20-1, 50-30-1, 100-50-1, 100-100-1, 150-100-1, 200- 100-1, 200-150-1) and 4 layers (30-20-10-1). The threshold for classification will run from 0.1 to 0.9 in increments of 0.01. The test was performed on 1103 DoS attack records and 31616 non-DoS attack records (20% of the records from UNSW-NB 15 dataset).

The selection of the number of characteristic features stops at 6 features: Service, Sttl, Dttl, Smeansz, Ct_state_ttl, Ct_srv_dst. The results of classification of DoS attack and not DoS attack when using these 6 characteristic features are as follows: 85.31% - 84.71%, respectively: From 6 features specific to DoS attack, we continue to test for CMAC network [18,19]. To enter the CMAC network, it is necessary to quantize the input vector based on the maximum and minimum values of each feature. The maximum value for quantization corresponding to each feature applied to the CMAC neural network is: 17, 257, 257, 1025, 9, 65. The learning process of the CMAC neural network depends on the value of generic parameter ρ, when $\rho = 2,\ 4,\ 6,\ 8,\ 16,\ 32$. In addition, the accuracy depends on the threshold. The number of training steps is 10 000 000. The CMAC neural network test is performed with different threshold values ϑ from 0.1 to 0.9 with a step of 0.01, comparing the obtained results, the recognition results. The highest DoS attack and non-DoS attack were obtained when the threshold value was 0.57, the recognition rate of DoS attack and not DoS attack was 86.49% and 85.13%.

V. CONCLUSION

From the above results, it is shown that the RF-MLP method gives higher results than the PCA method on all feature sets. The PCA method only gives the number of features after transforming in the new space, but does not specify the names of the selected features from the original data set like the RF-MLP method. The results found that 6 features specific to DoS attack of RF-MLP method as input of CMAC neural network also give better results than all other networks. In future studies, we will apply this method to other types of attacks and test it on different types of networks.

REFERENCES

[1] S. E. Smaha *et al.*, "Haystack: An intrusion detection system," in *Fourth Aerospace Computer Security Applications Conference*, vol. 44. Orlando, FL, USA, 1988.

[2] 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 Applications*, vol. 36, no. 1, pp. 16–24, 2013.

[3] H. Zhao, Z. Wang, and F. Nie, "A new formulation of linear discriminant analysis for robust dimensionality reduction," *IEEE Transactions on Knowledge and data engineering*, vol. 31, no. 4, pp. 629–640, 2018.

[4] S. L. France and J. D. Carroll, "Two-way multidimensional scaling: A review," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 41, no. 5, pp. 644–661, 2010.

[5] N. Pilnenskiy and I. Smetannikov, "Feature selection algorithms as one of the python data analytical tools," *Future Internet*, vol. 12, no. 3, p. 54, 2020.

[6] G. Borboudakis and I. Tsamardinos, "Forward-backward selection with early dropping," *The Journal of Machine Learning Research*, vol. 20, no. 1, pp. 276–314, 2019.

[7] "Dataset adfa-nb15," https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/, accessed: 2020-10-18.

[8] S. Ma and Y. Dai, "Principal component analysis based methods in bioinformatics studies," *Briefings in bioinformatics*, vol. 12, no. 6, pp. 714–722, 2011.

[9] T.-M. Hoang and T.-L. Le Thi, "A study on ids based cmac neuron network to improve the attack detection rate," in *International Conference on Industrial Networks and Intelligent Systems*. Springer, 2021, pp. 504–511.

[10] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.
[11] "1998 darpa intrusion detection evaluation dataset," https://www.ll.mit.edu/ideval/data/1998data.html, accessed: 2020-10-18.
[12] "Kdd cup 1999 data," http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, accessed: 2020-10-18.
[13] "defcom17nsl_kdd," https://github.com/defcom17/NSL_KDD, accessed: 2020-10-18.
[14] K. Rieck and P. Laskov, "Detecting unknown network attacks using language models," in *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*. Springer, 2006, pp. 74–90.
[15] "Adfa-ids-datasets," https://www.unsw.adfa.edu.au/australian-centre-for-cyber-security/cybersecurity/ADFA-IDS-Datasets/, accessed: 2021-6-18.
[16] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *computers & security*, vol. 31, no. 3, pp. 357–374, 2012.
[17] "Forrest, s. university of new mexico (unm) intrusion detection dataset," http://www.cs.unm.edu/ immsec/systemcalls.htm, accessed: 2020-6-18.
[18] "Http dataset csic 2010," https://www.tic.itefi.csic.es/dataset/, accessed: 2021-7-10.