Journal of Engineering and Applied Sciences 13 (14): 5890-5896, 2018
ISSN: 1816-949X

Toward Mining Salient Attributes Based on Developing Principle Component Analysis Algorithm

Karim Hashim Al-Saedi
Mustansiriyah University, Baghdad, Iraq

Abstract: An intrusion detection system gathers and tests information from different parts within a computer network to characterize possible security threats that include threats from both outside and inside of the organization. This system generates a large volume of alerts by detecting these threats which contains irrelevant and redundant attributes. Attribute selection, therefore is an important step in data mining. Most researches use all attributes in their databases while some of these features may be irrelevant or redundant and they do not participate to the process of intrusion detection. Therefore, different attribute ranking and attribute selection techniques are proposed. In this study, presented salient attributes mining technique according on improved principle component analysis algorithm which are utilized to select, rank reliable attributes and remove inefficient attributes to have a more precise and reliable intrusion detection process standard database.

Key words: Attributes extraction, network security, data mining, reliable, detection, database

INTRODUCTION

During the last years, the number of unauthorized activities, intrusions and attacks in computer networks has grown extensively. With the explosive increase in number of services accessible through the internet, information security of using internet as the media needs to be carefully concerned and a sufficient protection is needed against cyber-attacks (Mokarian *et al*., 2013). An Intrusion Detection System (IDS) is a hardware device or software program that analyzes computer system activities and/or network traffics to detect malicious activities and produces alerts to security experts (Masdari and Bakhtiari, 2014). IDS can be classified according to IDS's environment as: a Network-based IDS (NIDS) that is a dedicated computer or special hardware platform with detection software installed that captures packets in a promiscuous mode or as a Host-based IDS (HIDS) that monitors the resource usage of the Operating System (OS) and the network. HIDS can only monitor the resource usage of the applications and not the applications themselves (Hashim and Abdulmunem, 2013).

However, NIDSs usually generated thousands of alerts even for a day. Worse, those alerts are in low quality because they mixed with false positives and repeated warnings for the same attack or alert notifications from erroneous activity. Therefore, manually analyze those alerts are tedious, time-consuming and error-prone (Siraj *et al*., 2009). These alerts is consists of set of attributes, sensor, alert type, classification, priority, date, time (hours, minutes, seconds and milliseconds), source IP address, destination IP address, source port number, destination port number, protocol, TTL, TOS, ID, Iplen, Dgmlen, type, code and packet type (El-Taj and Abouabdalla, 2010). Feature selection is a pre-processing data mining technique that finds a minimum subset of features that captures the relevant properties of a dataset. Given that no loss of relevant information is incurred with a reduction in the original feature space, feature selection has been widely used (Karimi and Harounabadi, 2013). The remaining part of this study organized as follows: next study presents problem statement and then study presents methodology of feature extraction. Finally, a discussion about the resulted feature extraction method and the obtained results presented followed by conclusions.

Problem statement: IDS deals with large amount of alerts which contains various irrelevant and redundant features. These alert attributes causing slow training and testing process, higher resource consumption and decreasing computational time.

MATERIALS AND METHODS

The goal of this research is to find the best of PCA method for reducing intrusion alerts attributes. Our system architecture composed of three main components: alert normalization, alert preprocessing and dimension reduction as illustrated in Fig. 1.

In the first component, alerts that were generated by IDS snort was collected and stored in database before they were modeled and converted into a numeric format. The formatted alerts were represented in numerical format and scaled to produce a balanced dataset. Since, the number of alerts was huge and the alerts information was massive, we reduced the dimensionality of data using PCA.

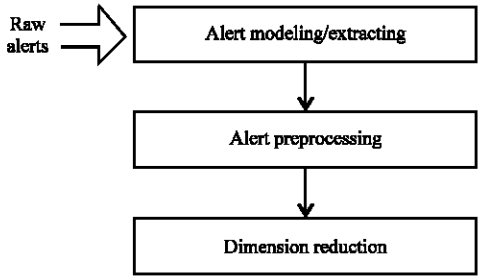

Fig. 1: Proposed system architecture

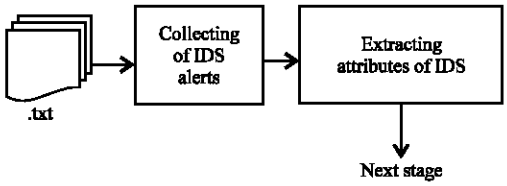

Fig. 2: The first stage of system components

Alert modeling/ extracting: It determines the different data formats and helps exchange and share information of interest as well as to the management systems that might be included.

Alert modeling/extracting has two main components: collecting of IDS Alerts and extracting attributes of IDS alerts as shown in Fig. 2.

Collecting of IDS alerts: It is the first component of alert modeling/normalization that receives IDS alerts from IDS and saves the said alerts into one text file. Figure 3 clearly shows parts of the IDS Snort alert text file with its attributes.

Extracting attributes of IDS alerts: Data extraction of IDS alerts is the second component of alert modeling/ normalization which is responsible for extracting the standard features from the IDS alert file after the first

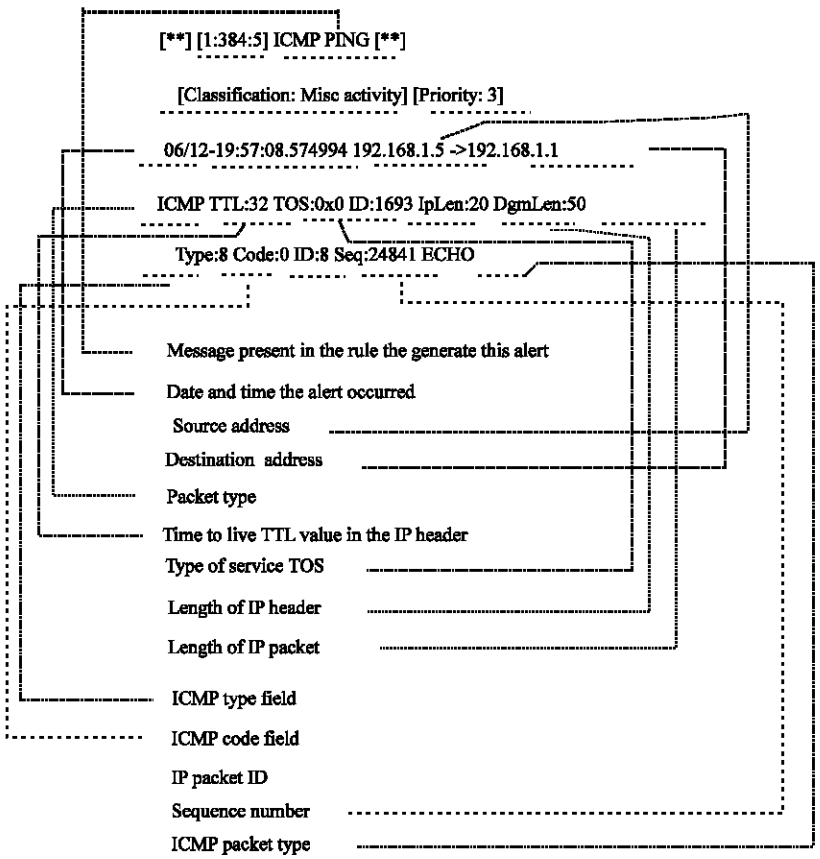

Fig. 3 : The full mode components (El-Taj and Abouabdalla, 2010)

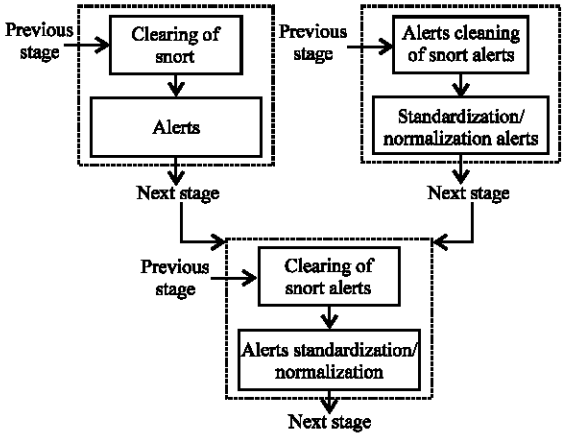

Fig. 4: The second stage of system components

component has performed its function. We extracted twenty one attributes for each alert. Thus, a vector for an alert A = {AlertID, Sensor ID, Alert Type, classification, priority, date, hour and minute, millisecond, Source IP Address, Destination IP Address, Source Port, Destination Port, Service Protocol, TTL, TOS, id, Iplen, dgmlen, type1, endtype }. To manage all attributes in more manageable way, each attribute is stored in a field as shown in the following algorithm.

Algorithm 1; Converting algorithm:

Input : Alerts Log File
Output: Log Table (LT)
Begin
1) Open a DB connection
2) Create a table to store log file
3) Open log file
4) While not end of log file
   a. Read an entry of log file
   b. Tokenize the fields depending on delimiter character
   c. Insert all fields into the Log Table (LT)
5) End while
6) Close a DB connection and Log File
End

Alerts preprocessing: The purpose of it is to produces result that can be used to improve and optimize the content of data. In this phase, the starting and critical point for successful log mining is alerts preprocessing. The input of this module is a log table and the output is a processed log table (contains fills missing values, interested fields, uninterested fields). It has two main components: cleaning of IDS alerts and alerts standardization/normalization as shown in Fig. 4.

Clearing: It is the first stage from data preprocessing. Because ICMP is used, port attributes for each source and destination in alerts is not included, so, they replaced by -1 value because it does not exist between port ranges (0-65536). When snort generated alerts for TCP or UDP service protocols, these attributes in these alerts did not contain any values, e.g., type1, code and endtype attributes. This problem will be solved by adding values that do not conflict with values in ICMP service protocol. In the type1 attribute, we will insert the 256 value because it does not exist between (0-255) ICMP types. The code attribute will insert the 16 value because it does not exist between (0-15) ICMP subtypes while the endtype attribute will insert the "none" value because it text values and later replace by zero value as shown in the Algorithm 2.

Algorithm 2; Cleaning algorithm:

Input : Alerts Log File
Output: Log Table (LT)
Begin
1. Open log file
2. While not end of log file
3. Read an entry of log file
4. While line != " "
5. Read lines until line3
6. If number x before (->)symbol even (:)symbol then Source Port number = x
7. Else Source Port number = -1
8. If number x between last (:)symbol and end line3 then destination Port number = x
9. Else destination Port number = -1
10. Read lines until line5
11. If serviceprotocl == "TCP" or "UDP" then
12. Type1 attribute =256 and code attribute = 16 and endtype attribute = "none"
13. End while
14. End while
End

Alerts standardization: Alert attributes are in the form of numerical and non-numerical values. Attributes that contain numerical values are Alert ID, Sensor ID, priority, source port, destination port, miliseconds, TTL, TOS, iplen, dgmlen, type1 and code. The rest are non-numerical values (i.e., AlertType, classification, date, time, SourceIPaddress, DestinationIPaddress, ServiceProtocol, Tos and endtype) and have to be mapped into numerical values. For instance to convert a 32 bit IP address (IPaddr) which in X1.X2.X3.X4 format, mapping as Eq.1 was used:

$$IP_{addr} = ((X1*256+X2)*256+X3)*256+X4 \quad (1)$$

Preprocessing unit converts string values of attributes of alert to numerical data as shown in Eq. 2:

$$\text{Service protocol} = \begin{cases} 1, \text{protocol} = \text{ICMP} \\ 2, \text{protocol} = \text{ICMP} \\ 3, \text{protocol} = \text{ICMP} \end{cases} \quad (2)$$

Alerts normalization is used for normalizing the data attempts to give all attributes an equal weight. In z-score normalization (or zero-mean normalization) as shown in Eq. 3:

$$V_{new} = (V_{old}\text{-mean})/\text{standard deviation} \quad (3)$$

$$\text{Mean} = \sum_{i=1}^{n} x_i / n \quad (4)$$

$$\text{Standard deviation} = \sqrt{\sigma^2} = \sqrt{\sum_{i=1}^{N} \frac{(X_i-\mu)}{N-1}} \quad (5)$$

Dimension reduction: In order to make IDS more efficient, reducing the data dimensions and complexity have been used as simplifying features. Dimensionality reduction is the process of reducing the number of random variables or attributes under consideration. It can reduce both the data and the computational complexity. It can also get more efficient and find out the useful feature subsets (Han and Kamber, 2012).

Principle component analysis: Main thought of principal component analysis is variable's dimension reduction. It is a statistical analysis method that transforms multiple variables into fewer main variables (Xie, 2014). It involves a mathematical procedure that transforms a number of correlated variables into a (smaller) number of uncorrelated variables called principal components (Paul *et al*., 2013). Here are five steps to get the PCA for a given data (Abbas, 2015).

Step 1: Get some data and Organize the data set. The data is an table which is represented by p*p matrix.

Step 2: Calculate the mean and Subtract the mean. Find (Centered Data Matrix) by subtracting the mean from all the entities. The mean along each column in data set and subtract the mean from each of the data dimensions. To find the mean, it is defined to be:

$$\mu = \sum_{i=1}^{n} X_i \quad (6)$$

$$\text{Centered matrix} = \frac{x_i\text{-mean}}{\text{Standard deviation}} \quad (7)$$

Step 3: Calculate the covariance matrix. Covariance is a measure of how much each of the dimensions varies from the mean with respect to each other. The covariance between one dimension and itself is the variance (Anderson, 2013):

$$\begin{bmatrix} \sum \frac{(X_i-\mu)(X_i-\mu)}{N} & \sum \frac{(X_i-\mu)(Y_i-\mu)}{N} \\ \sum \frac{(X_i-\mu)(Y_i-\mu)}{N} & \sum \frac{(Y_i-\mu)(Y_i-\mu)}{N} \end{bmatrix} \quad (8)$$

Since, the covariance matrix is square (P*P), we can calculate the Eigenvectors and Eigenvalues for this matrix, using the following condition:

$$(A. = x.\lambda) \quad (9)$$

Where:

A = The matrix

μ = The eigenvalue and xthe eigenvector

Now by solving this equation will produce the Eigenvalues and Eigenvectors (Anton, 2010; Brown *et al*., 2011).

Step 4: Calculate the Eigenvectors and Eigenvalues of the covariance matrix.

Step 5: Choosing components (Rearrange the Eigenvectors and Eigenvalues) and forming a feature vector.

In fact, it turns out that the eigenvector with the highest Eigenvalue (variances of the principal components) is the principle component of the data set. The eigenvector with the largest eigenvalue was the one that pointed down the last of the data. It is the most significant relationship between the data dimensions. In general, once Eigenvectors are found from the covariance matrix, the next step is to order them by eigenvalue, highest to lowest. This gives the components in order of significance (Paul *et al*., 2013; Ilin and Raiko, 2010):

$$\text{Feature vector} = (\text{Eigenvector 1, eigenvector 2 and ..., eigenvector p}) \quad (10)$$

RESULTS AND DISCUSSION

The proposal has been implemented on the following platform: Windows 7 Ultimate Service Pack1 and 64-bit OS, 4 GB RAM and Intel® Core (TM) i5 CPU with 2.5 GHz and by using MATLAB R2013a programming language and Microsoft Excel 2010. The results are tested by using many log files (written in tcpdum format) in website https://www.ll.mit.edu/ideval/data/1999data .html. The Darpa 1999 is used in the research for reducing

1	-0.73	0.51	-0.81	-0.36	0.15	0.33	-0.30	0.64	-0.04	0.64	0.75	-0.17	0	0.60	0	0.72	0.02	0.01	-0.03
-0.7	1	-0.55	0.90	0.39	-0.13	-0.36	0.40	-0.74	0.02	-0.71	-0.83	-0.03	0	-0.72	0	-0.46	0.17	-0.18	-0.14
0.51	-0.55	1	-0.54	-0.17	0.04	0.40	0.19	0.49	0.40	0.46	0.35	-0.08	0	0.35	0	0.34	0.00	0.13	-0.02
-0.81	0.90	-0.54	1	0.44	-0.14	-0.39	0.46	-0.87	0.08	-0.84	-0.96	0.20	0	-0.79	0	-0.63	-0.03	-0.19	0.07
-0.36	0.39	-0.17	0.44	1	-0.40	0.07	0.31	-0.37	0.18	-0.37	-0.44	0.08	0	-0.37	0	-0.29	-0.16	-0.08	0.18
0.15	-0.13	0.04	-0.14	-0.40	1	0.02	-0.11	0.10	-0.08	0.13	0.15	-0.03	0	0.13	0	0.13	0.09	0.03	-0.10
0.33	-0.36	0.40	-0.39	0.07	0.02	1	0.00	0.33	0.18	0.29	0.34	-0.09	0	0.41	0	0.14	-0.06	0.19	0.03
-0.30	0.40	0.19	0.46	0.31	-0.11	0.00	1	-0.29	0.56	-0.41	-0.57	0.10	0	-0.62	0	-0.28	-0.05	-0.08	0.06
0.64	-0.74	0.49	-0.87	-0.37	0.10	0.33	-0.29	1	-0.10	0.67	0.82	-0.18	0	0.58	0	0.42	0.01	0.13	-0.04
-0.04	0.02	0.40	0.08	0.18	-0.08	0.18	0.56	-0.10	1	0.01	-0.19	0.02	0	-0.12	0	0.05	0	0.01	0.00
0.64	-0.71	0.46	-0.84	-0.37	0.13	0.29	-0.41	0.67	0.01	1	0.85	-0.15	0	0.68	0	0.68	0.01	0.15	-0.04
0.75	-0.83	0.35	-0.96	-0.44	0.15	0.34	-0.57	0.82	-0.19	0.85	1	-0.20	0	0.78	0	0.59	0.03	0.17	-0.07
-0.17	-0.03	-0.08	0.20	0.08	-0.03	-0.09	0.10	-0.18	0.02	-0.15	-0.20	1	0	-0.16	0	-0.09	-0.83	-0.04	0.82
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0	0
0.60	-0.72	0.35	-0.79	-0.37	0.13	0.41	-0.62	0.58	-0.12	0.68	0.78	-0.16	0	1	0	0.49	0.05	0.18	-0.08
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0	0
0.72	-0.46	0.34	-0.63	-0.29	0.13	0.14	-0.28	0.42	0.05	0.68	0.59	-0.09	0	0.49	0	1	0.04	0	-0.04
0.02	0.17	0.00	-0.03	-0.16	0.09	-0.06	-0.05	0.01	0.00	0.01	0.03	-0.83	0	0.05	0	0.04	1	-0.01	-0.98
0.01	-0.18	0.13	-0.19	-0.08	0.03	0.19	-0.08	0.13	0.01	0.15	0.17	-0.04	0	0.18	0	0.00	-0.01	1	-0.17
-0.03	-0.14	-0.02	0.07	0.18	-0.10	0.03	0.06	-0.04	0.00	-0.04	-0.07	0.82	0	-0.08	0	-0.04	-0.98	-0.17	1

Fig. 5: Covariance matrix

-1.06E-15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p	-1.27E-16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	4.63E-17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0.006	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0.047	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0.09	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0.15	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0.17	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0.19	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0.29	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0.37	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0.44	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0.53	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0.71	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.794	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1.11	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1.22	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.016	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.81	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7.04

Fig. 6: Eigenvalues matrix

alerts features. After installed and ran snort 2.9.9.0 on personal computer and downloaded log file in dcpdump format.

In the PCA, the input alerts consists of 21attributes and the important step in this method is covariance matrix by 20*20 size with excepted alert-id attributes as shown in Fig. 5.

By using Fig. 5 covariance matrix S, the next step is find Eigenvalues and Eigenvectors. The Eigenvalues is 20*20 matrix that all its elements equal zero except the main diagonal elements (Eigenvalues) as shown in Fig. 6.

The main diagonal elements is concluded then rearrange order in descending order as shown in Table 1. In addition to the Eigenvalues in Table 1 there are two columns. First because the Eigenvalues are variances of the principal components, the proportion of variance explained by the first k components

While the other column (second) is cumulative proportion that calculate the cumulative sum of Eigenvalues. In the Eigenvectors, it extracted from covariance matrix and rearrange reversely (the final column is become the first columns vice versa and so on). It displays Eigenvectors that are 20*20 matrix. Rows are

Table 1: Eigenvalues

Eigenvalues	Proportion of variance	Cumulative proportion
7.035500	39.08580	39.0858
2.813400	15.63010	54.7160
2.016100	11.20080	65.9168
1.223100	6.79480	72.7115
1.109900	6.16620	78.8778
0.793600	4.40910	83.2869
0.708000	3.93320	87.2200
0.530700	2.94840	90.1684
0.440500	2.44700	92.6155
0.373300	2.07370	94.6892
0.292100	1.62300	96.3122
0.192000	1.06680	97.3791
0.174400	0.96870	98.3478
0.151800	0.84330	99.1911
0.093000	0.51690	99.7080
0.046700	0.25930	99.9673
0.005900	0.03270	100
4.633e-17	2.56e-16	100
-1.266e-16	-7.03e-16	100
-1.055e-15	-5.81e-15	100

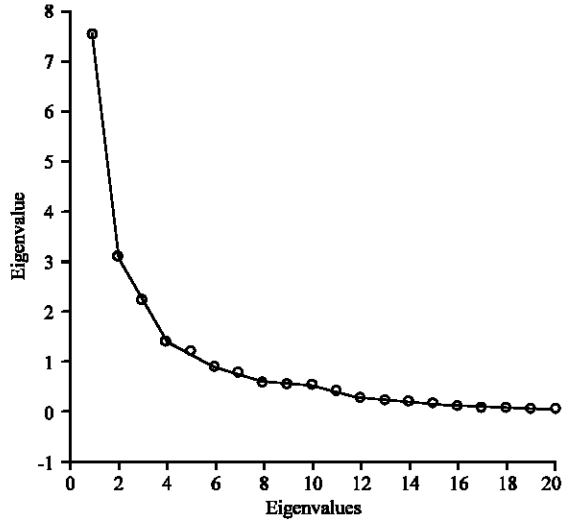

Fig. 7: Scree graph for Eigenvalues

represents the attributes while the columns are principle components (comp 1, comp 2, …, comp 20). As shown in Fig. 7, The first three Eigenvalues form a steep curve followed by a bend and then a straight-line trend with shallow slope. The recommendation is to retain those Eigenvalues in the steep curve before the first one on the straight line. Thus in Fig. 7, three components would be retained.

In Table 2, the first three principal components accounted by Scree graph as shown in Fig. 7. The corresponding Eigenvectors.

We will take the absolute values in Table 2 because the negative sign meaning the direction of the movement. Because that values in Table 2 is a little, we will multiply weights for each component in Table 2. The weighted vector is [1.9; 1; 1.3] and Table 2 will be as follow:

Table 2: The corresponding Eigenvectors for scree graph

Attributes	Component 1	Component 2	Component 3
Sensor	-0.3149	0.0320	-0.0671
Type	0.3304	-0.1523	0.0252
Classf	-0.2047	0.0716	-0.4563
Prio	0.3707	-0.0218	0.0180
Date	0.1867	0.1021	-0.2079
Hm	-0.0764	-0.0728	0.1316
Spart	-0.1546	0.0751	-0.3091
Ip_s	0.1910	0.0527	-0.4964
Port_s	-0.3113	0.0258	-0.0437
Ip_d	0.0329	0.0519	-0.5771
Port_d	-0.3264	0.0285	-0.0255
Protocol	-0.3583	0.0060	0.1016
Ttl	0.0820	0.3198	0.1109
Tos	-1.3553e-20	8.6736e-19	0
Id	-0.2139	5.8569e-04	0.0890
Iplen	0	-2.7756e-17	0
Dgmlen	-0.2563	0.0146	-0.0350
Typel	-0.0251	-0.2800	-0.0732
Code	-0.0731	-0.0362	-0.0637
endtype	0.0381	0.0118	0.0838

Table 3: The Eigenvectors after multiplying them by weighted vector

Attributes	Component 1	Component 2	Component 3
Sensor	0.5983	0.0320	0.0872
Type	0.6278	0.1523	0.0327
Classf	0.3890	0.0716	0.5932
Prio	0.7043	0.0218	0.0235
Date	0.3547	0.1021	0.2703
Hm	0.1451	0.0728	0.1711
Spart	0.2938	0.0751	0.4019
Ip_s	0.3629	0.0527	0.6453
Port_s	0.5916	0.0258	0.0568
Ip_d	0.0625	0.0519	0.7503
Port_d	0.6202	0.0285	0.0331
Protocol	0.6807	0.0060	0.1321
Ttl	0.1558	0.3198	0.1442
Tos	2.570e-20	8.676e-19	0
Id	0.2964	5.859e-04	0.1157
Iplen	0	2.776e-17	0
Dgmlen	0.4869	0.0146	0.0455
Typel	0.0476	0.2800	0.0952
Code	0.1389	0.0362	0.0828
endtype	0.0724	0.0118	0.1089

Table 4: The final results for improved PCA

Attributes	Component 1	Component 2	Component 3
Type	0.5983	0.0320	0.0872
Classf	0.6278	0.1523	0.0327
Prio	0.3890	0.0716	0.5932
Spart	0.7043	0.0218	0.0235
Ip_s	0.3629	0.0527	0.6453
Port_s	0.5916	0.0258	0.0568
Ip_d	0.0625	0.0519	0.7503
Port_d	0.6202	0.0285	0.0331
Protocol	0.6807	0.0060	0.1321

From the results that displayed in Table 3. The variables that have up of 0.5 value will use in the next stage as shown in Table 4. This value selected because mean in Gaussian by comparing the results shown in Table 4 which are based on the improved method, we observe that when performing the PCA, the results in Table 5 are shows a large number of attributes compared to the number of attributes shown in Table 6 with keeping

Table 5: The final results for PCA

Attributes	Component 1	Component 2	Component 3
Type	0.5689	0.5678	0.1240
Classf	0.0245	0.6423	0.0245
Prio	0.5648	0.2154	0.4578
Spart	0.6789	0.1145	0.2451
Ip_s	0.7845	0.2451	0.4578
Port_s	0.6584	0.0124	0.5564
Ip_d	0.0024	0.1428	0.6421
Port_d	0.7598	0.3265	0.0124
Protocol	0.3254	0.5648	0.2451
Ttl	0.5697	0.2546	0.2698
Id	0.6698	0.0012	0.2679
Typel	0.5487	0.6698	0.0167
Code	0.0245	0.2514	0.5014
endtype	0.2140	0.2457	0.6548

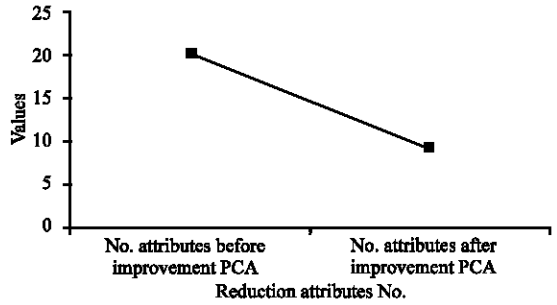

Fig. 8: Reduction rate

main attributes (Ports and IP addresses). The improved algorithm greatly reduced the number of attributes which effectively contributed to the speed of performance and reduced the size of memory used as well as accuracy in the results obtained (Fig. 8).

CONCLUSION

During the outcomes that shown in this study. We saw that improved PCA can decrease the number of attributes in dataset Darpa 1999 (20 attributes). After using improved PCA technique, the number of attributes that showing are 9 attributes. This meaning that the improved PCA can remove irrelevant attributes by elimination rate almost 55% with storing the main and important attributes.

REFERENCES

Abbas, E.I., 2015. Effect of eigenfaces level on the face recognition rate using principal component analysis. Eng. Technol. J., 33: 729-737.

Anderson, A., 2013. Business Statistics for Dummies. John Wiley & Sons, Hoboken, New Jersey, USA.,.

Anton, H., 2010. Elementary Linear Algebra. 10th Edn., John Wiley & Sons, Hoboken, New Jersey, USA., ISBN:978-0-470-45821-1, Pages: 567.

Brown, B.L., S.B. Hendrix, D.W. Hedges and T.B. Smith, 2011. Multivariate Analysis for the Biobehavioral and Social Sciences. John Wiley & Sons, Hoboken, New Jersey, USA.,.

El-Taj, H.R. and O.A. Abouabdalla, 2010. False positive reduction by correlating the intrusion detection system alerts: Investigation study. J. Commun. Comput., 7: 25-31.

Han, J. and M. Kamber, 2012. Data Mining Concepts and Techniques. 3rd Edn., Elsevier, New York, USA., ISBN:9789380931913, Pages: 703.

Hashim, S.H. and I.A. Abdulmunem, 2013. A proposal to detect computer worms (Malicious Codes) using data mining classification algorithms. Eng. Technol. J., 31: 142-155.

Ilin, A. and T. Raiko, 2010. Practical approaches to principal component analysis in the presence of missing values. J. Mach. Learn. Res., 11: 1957-2000.

Karimi, Z. and A. Harounabadi, 2013. Feature ranking in intrusion detection dataset using combination of filtering methods. Intl. J. Comput. Appl., 78: 21-27.

Masdari, M. and F.C. Bakhtiari, 2014. Alert management system using K-means based genetic for IDS. Intl. J. Secur. Appl., 8: 109-118.

Mokarian, A., A. Faraahi and A.G. Delavar, 2013. False positives reduction techniques in intrusion detection systems-a review. Intl. J. Comput. Sci. Netw. Secur., 13: 128-134.

Paul, L.C., A.A. Suman and N. Sultan, 2013. Methodological analysis of Principal Component Analysis (PCA) method. Intl. J. Comput. Eng. Manage., 16: 32-38.

Siraj, M.M., M.A. Maarof and S.Z.M. Hashim, 2009. Intelligent alert clustering model for network intrusion analysis. Int. J. Adv. Soft Computing Appl., 1: 1-16.

Xie, X., 2014. Principal component analysis-based sports dance development influence factors research. J. Chem. Pharm. Res., 6: 970-976.