Math. & Comput. Sci., 6(2) 2025, 77–91

Online: ISSN 2717-2708

Mathematics and Computational Sciences

Journal Homepage: http://mcs.qut.ac.ir/mcs

Computer networks anomaly detection by using PCA & pattern recognition

Elham Bideh[a], Javad Vahidi[b,*]

[a]Master Science of Computer Networks, Shomal University, Amol, Iran.

[b]Department of Computer Science, Iran University of Science and Technology,Tehran, Iran.

Abstract

The detection of anomalies in computer networks is one of the most considerable challenges that experts in this field are facing nowadays. Thus far, different artificial intelligence methods and algorithms have been proposed, tested, and utilized for detecting these anomalies. However, attempts made to enhance the speed and accuracy of these anomalies detection process are continuously ongoing. In this research, pattern recognition based on artificial neural networks is applied to automatically detect anomalies in computer networks. Also, to increase the speed of the pattern recognition based on the process of the neural network, the principal component analysis algorithm will be used as a method for dimension reduction of training samples. The results of the performed simulations based on the proposed methods in this research show that dimension reduction of training samples by principal component analysis algorithm and then applying the pattern recognition based on neural networks method leads to high-speed (less than 10 seconds) and high-accuracy (99-100%) detection of anomalies in computer networks.

Keywords: Computer Networks Anomaly Detection, Pattern Recognition, Artificial Neural Networks, Back Propagation Algorithm, Principal Component Analysis

2020 MSC: 68M15, 62H25

1. Introduction

The importance of detecting anomalies in the computer networks domain is that unprotected data may put significant and vital information at risk. Hence, anomaly detection has been considered for several decades in order to recognize and elicit abnormal components in computer networks [22]. For example, detecting an unusual computer network traffic pattern may indicate a hacking attack [15], and timely detection of this anomaly will lead to preventing the attack. Accordingly, an efficient algorithm is required for identifying anomalies that attempt to destroy a computer network's features.

The authors of a paper published in 2018 introduced a modern method for describing network abnormality. Adjusting the Simple Network Management Protocol (SNMP) Management Information Base (MIB) parameters with the Fuzzy Rule Interpolation (FRI) was the capability of the introduced method. The proposed method was examined and measured and acquired a 93% identification rate [5].

A research paper related to 2019, introduces an approach using Hidden Markov Model (HMM) based on

*Corresponding author

Email addresses: e.bideh@outlook.com (Elham Bideh), jvahidi@iust.ac.ir (Javad Vahidi)

doi: 10.30511/mcs.2025.2052129.1301

Received: 30 January 2025 Accepted: 24 April 2025

the Viterbi algorithm for anomaly detection on the SNMP-MIB dataset. The obtained results indicate the introduced method was found effective [3].
In a paper published in 2021, an unsupervised anomaly detection method was presented which incorporates Sub-Space Clustering (SSC) and One-Class Support Vector Machine (OCSVM) to detect anomalies without any previous information. The proposed method was measured using the NSL-KDD dataset, and the tentative outcomes illustrated that the proposed method works better than some other manners [23].
In the research performed in 2021, a deep learning method named Stacked Autoencoder (SA) is suggested for discovering known network anomalies using the SNMP-MIB data. This method transforms the input data into a different set of decreased outputs (encoding). Former outputs are decoded to obtain the intended output of N dimension equal to the original input. The proposed method achieved high accuracy in the experimental results [6].
In January 2022 a paper proposed an intricate method to recognize anomaly data. This method was based on signature and entropy analysis and machine learning anomaly detection using fractal and recurrence analysis. In this research referring to the taken outcomes, the random forest classifier and deep residual neural network as machine learning methods had good classification accuracy. The effects of signature and entropy analysis also was displaying a high degree of anomaly detection. So, the paper suggested a method of anomaly detection based on the geminated use of explained and examined methods. This method was introduced as a complex anomaly detection method [2].
Similar to the mentioned papers, especially in recent years, most of the research has used unsupervised algorithms, complex procedures, or deep learning methods. In unsupervised algorithms, because the input data is not labeled by people in advance, the machine requires doing this itself. So unsupervised algorithms have less accuracy in the results [17]. Furthermore, the methods based on deep learning are intrinsically extremely reliant on computing power [25]. Also, most complex combined methods have high time complexity. Accordingly proposing a simple, effective and efficient method at maximum speed and precision for the automatic detection of anomalies in computer networks with the help of the pattern recognition method based on Artificial Neural Networks (ANN), is the main focus of this research and in the meantime, Principal Component Analysis (PCA) algorithm will be applied in order to increase the speed of the proposed method.
In the continuation of this paper, the basic concepts used in this research will be introduced in the second section. The third section describes and implements the proposed method. The fourth section includes the comparisons and finally, the conclusion and suggestions will be discussed in the fifth section.

2. Basic Concepts

2.1. Pattern Recognition

The operation of recognition can be separated into two main groups: Identifying actual items and identifying abstract items. Identifying abstract items means identifying items that are not real and do not exist physically [16]. The problem of classifying anomalies on computer networks can also be included in the category of abstract cases which we need to recognize and we will be able to solve this problem if we could achieve a suitable model for this diagnosis.
Recognition of an item includes three-phase of processing [16]:

- Input filtering
- Feature extraction
- Classification

Input filtering:
Filtering is removing unwanted information or data from input, which in turn, helps with the process of

feature extraction. Depending on the type of problem, the filter algorithm or method will change [16].
Feature extraction:
Feature extraction is a process of studying and deriving useful information from the filtered input patterns. The derived information may be general features, which are evaluated to ease further processing. The methods of feature extraction and the extracted features depend on the nature of the problems [16].
In many usages, data, which is the issue of analysis and processing in data mining, is multidimensional and has a multitude of features. The so-called curse of dimensionality relevant to many learning algorithms, defines the severe increase in computational complexity and the classification error in numerous dimensions [1]. Hence, the dimensionality of the feature space is mostly decreased before classification is undertaken [26]. In other words, feature extraction is a dimensionality reduction technique [10] that involves the process of deriving new features from the original features in order to reduce the cost of feature measurement, increase classifier efficiency and allow higher classification accuracy [7].
There were many different methods being used to explore the space of features and decide which feature could maximize the recognition rate such as [7]:

- Principal Component Analysis (PCA)
- Genetic Algorithm (GA)
- Neural Networks (NN)
- Threshold method

In this paper, PCA for the implementation of the feature extraction process will be examined.
Classification:
The final phase of pattern recognition is Classification. In this phase, an automatic system promulgates that the input item belongs to a specific category. Many classifiers can be used in this stage but designing the classifier is according to the following concepts [16]:

- Clustering concept
- Common property concept
- Member-roster concept

The pattern recognition methods based on these concepts are done through direct calculation by machines. Direct calculation is according to math-related techniques but there is another method applying biological concepts to machines for recognizing patterns. The invention of artificial neural networks is the outcome of this effort [16]. In this paper, artificial neural networks will be used to implement the classification phase.

2.2. Principal Component Analysis (PCA)

One of the most prevalent used feature extraction methods is PCA which is based on extracting the axes on which the data shows the highest variation. PCA converts the original set of features into a more little subset of linear combinations that include most of the variance in the initial set [18]. The basic idea of PCA is to specify the features which could illustrate as much of the total variation in the data as possible with as few of these features as possible. In PCA we have a penchant for finding a projection W [26]:

$$Y = W^{T}X \tag{2.1}$$

Where Y is a $P * 1$ transformed data point, W is a $P * P$ transformation matrix, and X is a $P * 1$ original data point. PCA can be done through eigenvalue decomposition of the covariance matrix S of the original data:

$$S = \sum_{i=1}^{n} (x_i - m)(x_i - m)^T \tag{2.2}$$

Where n is the number of instances, x_i is the ith instance, and m is the mean vector of the input data. Computation of the principal components can be presented by a framework as shown in Fig.1.

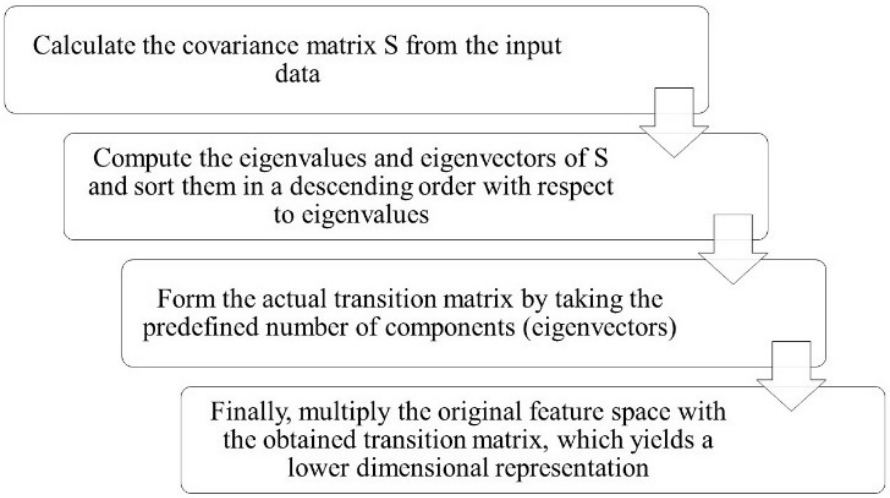

Figure 1: The principal component analysis framework

Some of the PCA properties are [26]:

- There is a maximum variance of the extracted features
- There is no correlation between the extracted features
- There is maximum information in the extracted features set
- PCA finds the best linear approximation in the mean-square sense

Therefore, if PCA is used as a feature extraction technique, a subset of features with a minimum number of features and a minimum correlation between the features is formed, which has the maximum possible information and variance. Using this subset as a neural network input will reduce the complexity of the neural network and increase the speed of its training and lead to an increase in the accuracy of classification as a result.

2.3. Artificial Neural Network (ANN)

One of the information processing systems is the neural network. A neural network includes enormous and simple processing modules with a high degree of interconnection between each module. The processing modules work cooperatively with each other and attain enormous parallel distributed processing. The neural networks imitate some biological actions of brains and neural systems in their design and function. Self-organization, adaptive learning and fault-tolerance capabilities are some of the advantages of neural networks. Because of their tolerance of noise, neural networks with proper training will respond correctly to unknown patterns. Due to these prominent abilities, neural networks can be used for the pattern recognition process. High-order nets, time-delay neural networks, recurrent nets and Back Propagation (BP), are some of the best neural models [16].

Neural networks can learn from their wrongs by presenting feedback to the input patterns. This type of feedback would be used to rebuild the input patterns and make them free from errors; so, it will enhance the efficiency of the neural networks. Such types of neural networks are very intricate to create. These types of networks are called auto-associative neural networks which use back propagation algorithms [16].

2.4. Neural Network with Back Propagation (BP) Algorithm

BP neural network learning is a type of supervised learning. The architecture of this type of neural network is shown in Fig.2. and the general steps for pattern recognition by using the BP algorithm are as follows [27]:

- Determine a sensible network structure. Particularly, considering the suitable number of neurons in the hidden layer of the network is an important factor in network structure and network performance.
- Determine the training set and test set. The training samples are used for training the network, while the test samples should monitor the efficacy of training and the rise of the network valence.
- Train the neural network for pattern recognition according to the training sample set and the test results.

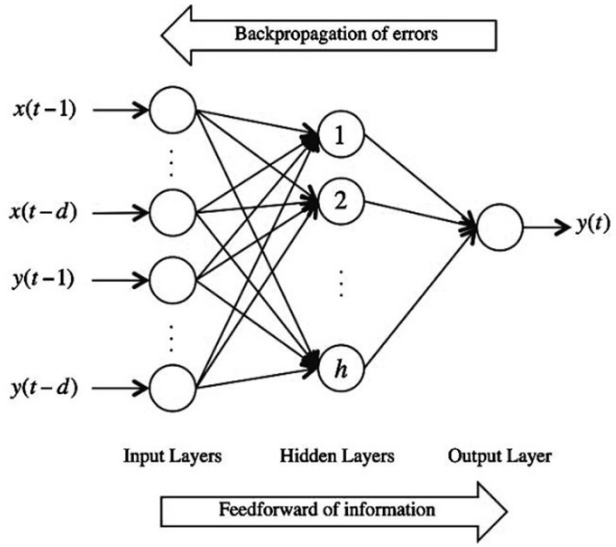

Figure 2: Feed forward backpropagation neural network architecture [8]

For the neural network training phase with the BP algorithm, there are various training methods or functions such as gradient descent (Traingd) [11], gradient descent with adaptive learning rate (Traingda) [12], gradient descent with momentum (Traingdm) [13], gradient descent with momentum and adaptive learning rate (Traingdx) [14], Levenberg-Marquardt (Trainlm) [20], etc.

2.5. Levenberg-Marquardt Method (lm)

Trainlm is a training function for the neural network with the BP algorithm that updates weight and bias amounts pursuant to Levenberg-Marquardt optimization. It does need more memory than other training functions, however, trainlm is often the swiftest BP training function and is extremely recommended as a best-choice supervised method. Trainlm uses the Jacobian for computations, which considers that performance is a mean or sum of squared errors. So, networks trained with the trainlm must benefit either the MSE or SSE performance function.
Trainlm is implemented through the following equations [27]:

$$F(x) = \sum_{i=1}^{N} V_i^2(x) = v^T(x)v(x) \tag{2.3}$$

In equation 2.3, F is the objective function. Also:

$$v(x) = (v_1(x), v_2(x), ..., v_n(x))^T \tag{2.4}$$

$$\Delta x_k = x_{k-1} - x_k = -[H(x_k) + \mu_k I]^{-1} J^T(x_k) v(x_k) \tag{2.5}$$

In equation 2.5, J is the Jacobian matrix of F and k is the iteration. Equation 2.6 represents the Jacobian matrix:

$$J(x) = \begin{bmatrix} \frac{\partial v_1(x)}{\partial x_1} & \frac{\partial v_1(x)}{\partial x_2} & \cdots & \frac{\partial v_1(x)}{\partial x_n} \\ \frac{\partial v_2(x)}{\partial x_1} & \frac{\partial v_2(x)}{\partial x_2} & \cdots & \frac{\partial v_2(x)}{\partial x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial v_N(x)}{\partial x_1} & \frac{\partial v_N(x)}{\partial x_2} & \cdots & \frac{\partial v_N(x)}{\partial x_n} \end{bmatrix} \tag{2.6}$$

Also, H is an approximation of the Hessian matrix of the matrix F, taken as:

$$H(x_k) = J^T(x_k) J(x_k) \tag{2.7}$$

μ_k is a parameter larger than zero that is used internally to control iterations in the lm algorithm. If μ_k is near zero, lm algorithm is near to the Gauss-Newton manner. If μ_k is large, the lm algorithm approximates the steepest descent method [27].
Newton's method is more rapid and more correct near a minimum of error. As a result, the purpose is to shift toward Newton's manner as swiftly as possible. μ_k is declined after any favored step (decrease in performance function error) and is grown only when an experimental step would append the performance function error. Therefore, the performance function error will evermore be decreased at each repetition of the algorithm. If μ_k becomes larger than μ_k maximum amount, the algorithm is stopped.

3. Proposed Method

The proposed method in this paper will solve the anomaly detection issue in computer networks by means of the pattern recognition process and to do so in the classification phase it gets help from the neural network which uses the BP algorithm with trainlm method. Also, in a separate approach, this method will use PCA in the feature extraction phase. Fig.3. shows a diagram related to the presented method and the algorithms of the proposed method are as follows:

Algorithm 1 Feature extraction phase by PCA:

1: Get $[x]_{n*m}$ (Data matrix with n sample and m features)
2: Normalize $[x]_{n*m}$
3: Calculate covariance matrix $S = \frac{1}{n-1} x^t x$
4: Calculate eigenvalues (D) and eigenvectors (V) from $S = VDV^{-1} = VDV^T$
5: Sort D descending
6: Reorder (V, order)

Based on previous comparisons and analyses [9, 21], most datasets related to computer network anomalies are relatively small and exhibit limited diversity in attack types and traffic patterns. Consequently, this study employs one of the most comprehensive, well-documented, and practical datasets available in this field. Researchers in similar studies have widely used this dataset to evaluate their proposed methods. As a result, its use in the present research not only enhances the reliability and credibility of the results but also enables a meaningful comparison with other proposed methods in this domain. The mentioned dataset meets key requirements for generating a series of effective data points. It includes important attack scenarios and injection methods. The dataset comprises 4,998 records, organized into 8 distinct clusters

Algorithm 2 Classification phase by neural network with BP algorithm & lm train function:

1: Initialize:
- Randomly initialize small weights W_1 & W_2
- Set initial damping factor $\mu(e.g., 0.001)$
- Define stopping threshold and maximum iterations

2: Repeat until stopping condition is met:

A. Forward pass:
- $H_{input} = X * W_1$
- $H_{output} = f(H_{input})$
- $O_{input} = H_{output} * W_2$
- $O_{output} = f(O_{input})$

B. Compute error:
- $E = (1/2) * \sum (T - O_{output})^2$

C.Compute Jacobian J:
- $d_O = (T - O_{output}) * f'(O_{output})$
- $d_H = (d_O * W_2^T) * f'(H_{output})$
- $J(W_2) = H_{output}^T * d_O$
- $J(W_1) = X^T * d_H$
- $J = [J(W_1), J(W_2)]$

D. Compute Hessian approximation & gradient:
- $H \simeq J^T * J$
- $g = J^T * e$

E. Update weights:
- $\Delta W = (H + \mu I)^{-1} * g$
- $W_1 = W_1 - \Delta W_1$
- $W_2 = W_2 - \Delta W_2$

F. Adjust damping factor μ :
- If error decreases, $\mu = \mu/10$
- If error increases, $\mu = \mu * 10$

3: Return final weights W_1&W_2

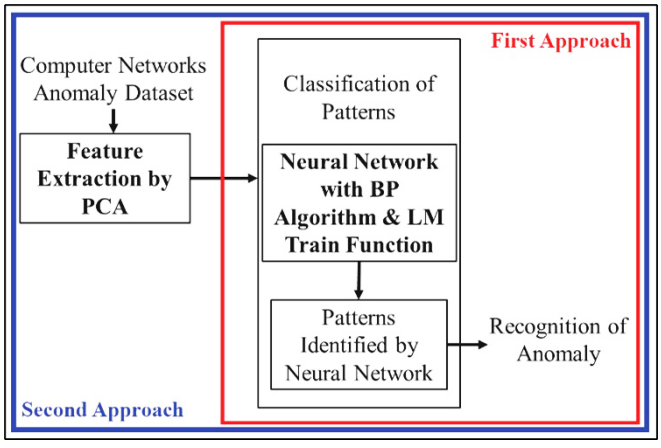

Figure 3: Diagram of the proposed method

[4]. Each sample in the dataset is described by 34 collected features. In this research, these features as two separate approaches, once participate completely in the classification phase of the pattern recognition process and another time after going through the feature extraction phase using PCA algorithm and turn into a smaller subset of features, enter the classification phase. Finally, in addition to evaluating the

recognition accuracy **(accuracy is the percentage of correctly identified normal and abnormal traffic)** of the proposed method, the results of simulating both approaches will be reviewed and compared in terms of neural network training time and the measure of Mean Squared Error (MSE). It is notable that due to the nature of the issue and the studied data set, the first phase, which is the input filtering, is not discussed in this research. The details of the step-by-step implementation and simulation of the proposed method by MATLAB software are described below and the obtained results are discussed.
It is also important to mention that this research has been implemented in an environment with the following hardware and software specifications:

- Processor: Intel Core i5-8265U, 1.60GHz 1.80 GHz
- RAM: 8 GB
- Operating system: Windows 10
- MATLAB: R2022a version

3.1. First Approach: No Implementation of Feature Extraction Phase

In this approach, in order to recognize computer network anomalies through the existing neural network tool in MATLAB software or command line functions, the neural network is trained for recognizing a suitable pattern according to all features available in the dataset.

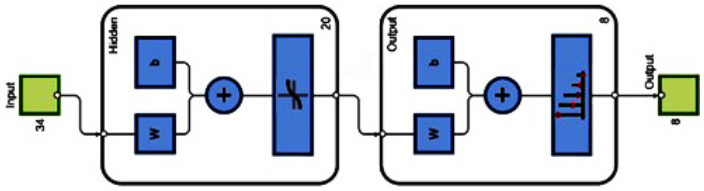

Figure 4: Neural network designed for classification with all features

The number of neurons of the input layer is equal to the number of features in each sample of the dataset, i.e., 34 neurons. The number of neurons in the output layer is equal to that of output labels, i.e., 8 neurons. Considering the fact that a feed-forward network that has one hidden layer and enough neurons in its hidden layer can matchable every limited input-output mapping subject, the number of hidden layers intended for implementation of this method is just one hidden layer with 20 neurons (close to the average number of neurons in input and output layers). Train and performance functions are lm and MSE, respectively. The initial mu parameter is equal to 0.001, mu decrease factor equal to 0.1, mu increase factor equals 10 and the maximum amount of mu is 1e10. Moreover, maximum iterations and maximum validation failures have been fixed at 50 and 6 respectively. Dataset has been divided into 3 sections: train, validation and test forming 70%, 15% and 15% of the total dataset, respectively. Fig.4. indicates a general scheme of the designed artificial neural network.
Any time a neural network is trained, it can sequel in a variant result due to diverse initial weight and bias amounts and disparate partitions of data into training, validation and test sets. So, several neural networks trained on the identical issue can give several outputs for the equal input. Therefore, it is necessary to repeat training several times to ensure that a neural network of proper accuracy has been found. In the following, artificial neural network training of the proposed approach is repeated 10 times, and the amount of time required for the training process, as well as the performance error (MSE) and recognition accuracy (ACC) on the test set for every 10 replications (and the average of ACC), are collected in table 1. Also, the diagram in Fig.5. includes the process of reducing performance error in the neural network training process for the eleventh time of training in this approach, and Fig.6. shows the recognition accuracy on the test set for the eleventh time, which is equal to 99.7%.

Table 1: Required time for the training process, performance error (MSE) and recognition accuracy on the test set for 10 replications in the first approach.

Training Replication Number	Required Time (Second)	MSE	ACC (Percent)
1	90	1.8988e-06	100
2	85	0.0012	99.9
3	83	0.0018	99.3
4	83	0.0015	99.7
5	84	4.5784e-04	100
6	84	7.6357e-04	100
7	83	9.8008e-04	99.3
8	84	0.0012	100
9	83	0.0011	99.5
10	84	0.0011	99.9
Average			**99.76**

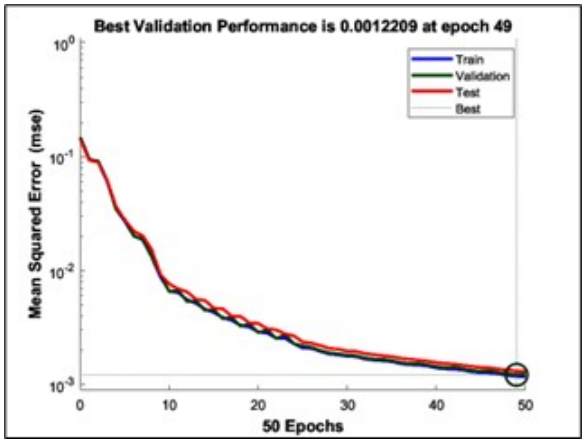

Figure 5: Process of reducing performance error in the first approach for the eleventh time of training

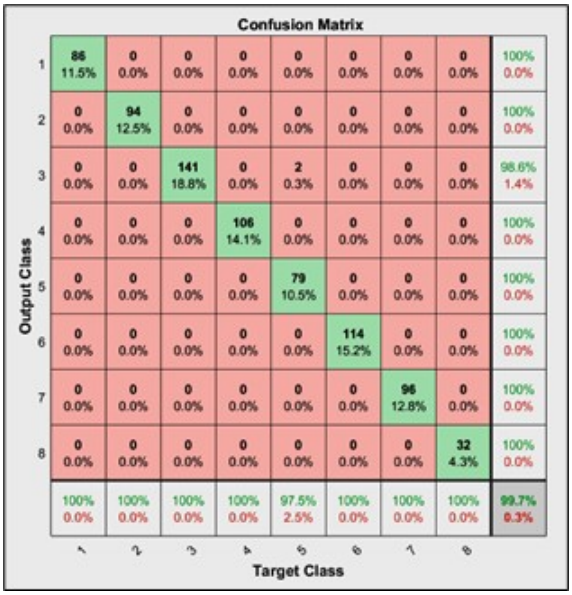

Figure 6: Recognition accuracy on the test set in the first approach for the eleventh time of training

3.2. Second Approach: Feature Extraction Through PCA

In this approach, in order to implement the feature extraction phase through the PCA method, the following stages are followed prior to the classification phase through the neural network:
$[W_p, T_p] = pca(Z)$;
At first, the above command line is implemented in which Z is the input data. Since conventional PCA is an unsupervised method [19], the only input dataset consisting of various combinations of features is made available to PCA and the labels of the samples obtained from such various combinations (targets) are not provided to PCA. In the input dataset, rows are samples and columns are dimensions (features). W_p puts eigenvectors in its columns in an order of priority and T_p is the corresponding values of input data on eigenvectors.
Then, the N number of desired dimensions (with higher priorities) are selected from eigenvectors through the following command line and are multiplied by the input data to create the new TN dataset with N features for each sample (N is smaller than and/or equal to the number of features of the input data, i.e., 34). Finally, TN will be provided to the neural network to implement the classification phase.
$TN = Z * W_p(:, 1 : N)$;
In the implementation carried out based on the conducted studies and evaluation of different numbers as N, finally, N was considered as 20.
The number of input layer neurons decreases to 20 neurons by reducing the features to 20 features. By considering 10 neurons as the neurons of the hidden layer and without making any change to the size of other parameters as compared to the first approach, Fig.7. indicates a general scheme of the designed neural network and table 2 includes the results obtained from the first to the tenth times of training. Also, the diagram in Fig.8. includes the process of reducing performance error in the neural network training process for the eleventh time of training in the second approach, and Fig.9. shows the recognition accuracy on the test set for the eleventh time, which is equal to 100%.

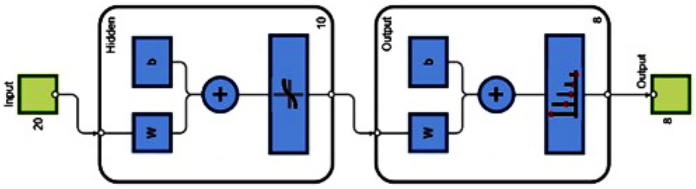

Figure 7: Neural network designed for classification with 20 features chosen by PCA

Table 2: Required time for the training process, performance error (MSE) and recognition accuracy on the test set for 10 replications in the second approach.

Training Replication Number	Required Time (Second)	MSE	ACC (Percent)
1	4	8.8837e-09	100
2	7	4.2231e-07	100
3	4	9.2533e-05	99.9
4	9	7.8617e-07	100
5	6	1.5172e-08	100
6	4	1.0808e-09	100
7	5	6.1925e-10	100
8	3	4.9531e-05	99.9
9	3	1.8658e-04	100
10	5	1.3899e-09	100
Average			**99.98**

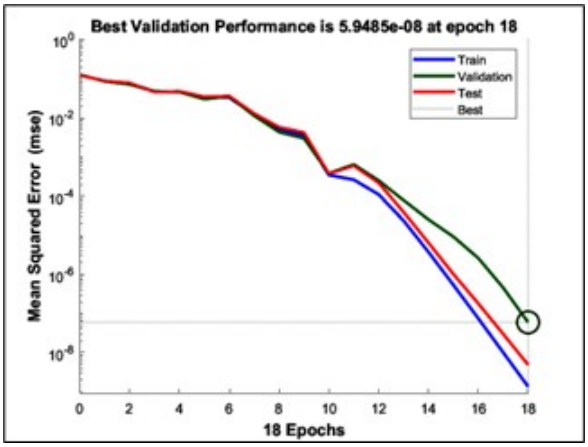

Figure 8: Process of reducing performance error in the second approach for the eleventh time of training

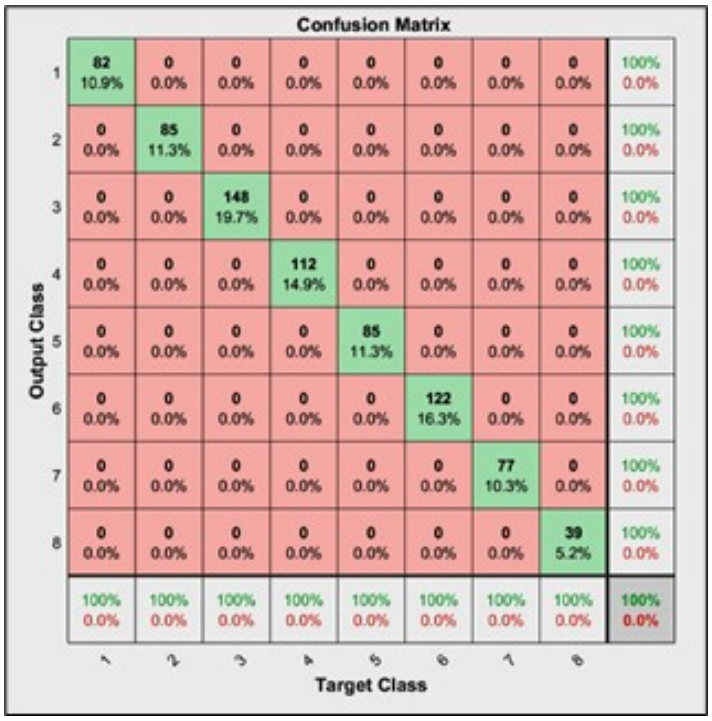

Figure 9: Recognition accuracy on the test set in the second approach for the eleventh time of training

4. Comparisons

4.1. Comparison of Proposed Approaches in Terms of Efficiency

After reviewing the presented tables and diagrams related to implementing the proposed method in this paper, two proposed approaches can be compared in Fig.10. to Fig.12. In these comparisons, it is observed that both presented approaches are able to detect and classify the anomalies in computer networks on the test set with 99-100% accuracy. But what more distinguishes these two approaches, is the difference in the neural network training time and the final amount of performance error on the entire data set. The maximum time required to train the neural network without performing the feature extraction phase prior to the classification phase is 90 seconds and the minimum time is 83 seconds. While if the feature extraction is performed through the PCA algorithm prior to the classification phase, the above times would have a range between 3 to 9 seconds.

Moreover, regarding the amount of performance error on the entire data set, the first approach has the error of 1.8988e-06 at best, and reaching this amount of error took 90 seconds. Whereas the second

approach, in its best replication, was able to reduce the amount of performance error to 6.1925e-10 in 5 seconds. With that considered, it can be concluded that using the PCA algorithm as a dimension reduction tool in the feature extraction phase of the pattern recognition process can improve the performance of this process and increase its speed and precision.

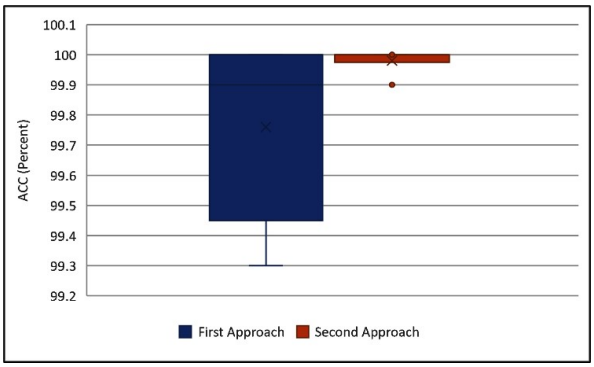

Figure 10: Diagram of comparing the accuracy of the proposed approaches

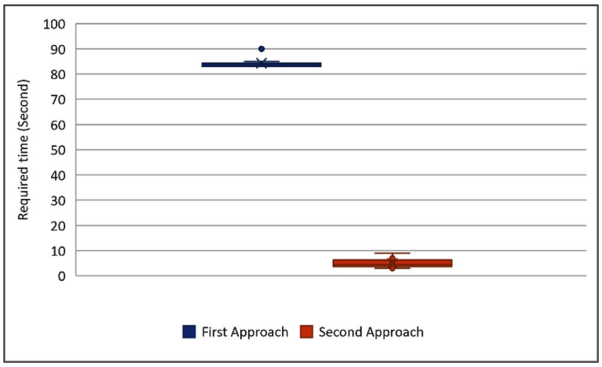

Figure 11: Diagram of comparing the required time for the proposed approaches

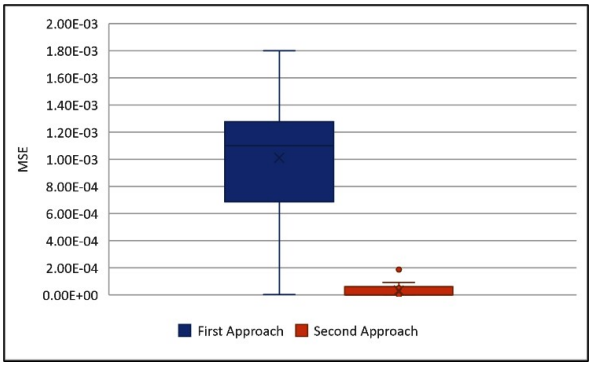

Figure 12: Diagram of comparing the MSE of the proposed approaches

4.2. Analysis and Comparison of Time Complexities

In the first approach, the time complexity is exclusively related to the execution of the neural network using the backpropagation algorithm with the lm training function, which is expressed as $O(nw^2 + w^3)$. Where n represents the number of samples, and w denotes the total number of weights, which in the first approach is calculated as $34 * 20 * 8 = 544034$.
In the second approach, the time complexity of the first phase, ie, feature extraction using the PCA method is $O(nd^2)$, considering that the number of features is less than the number of samples. Where

n is the number of samples and d is the number of features. The time complexity of the second phase is identical to that of the first approach, given by $O(nw^2 + w^3)$. Since the second phase has a higher complexity than the first phase, the overall time complexity of the second approach ultimately remains $O(nw^2 + w^3)$, similar to the first approach. With the difference that in the second approach, w equals $20 * 10 * 8 = 1600$. This means that, overall, the second approach, by reducing the number of features, also appears significantly more favorable than the first approach concerning time complexity.

4.3. Comparison with Some Other Proposed Methods

Regarding the comparison of the proposed methods of this paper with some other previous methods used in the field in question, it is possible to present Fig.13. based on the study conducted with a joint data set in 2019 [3].
The data in the diagram of Fig.13. are also summarized in Table 3 and show the proposed methods in the present study have higher performance with 99.76% and 99.98% accuracy (average) than the compared methods, which shows the supereminence of the methods presented in this paper.

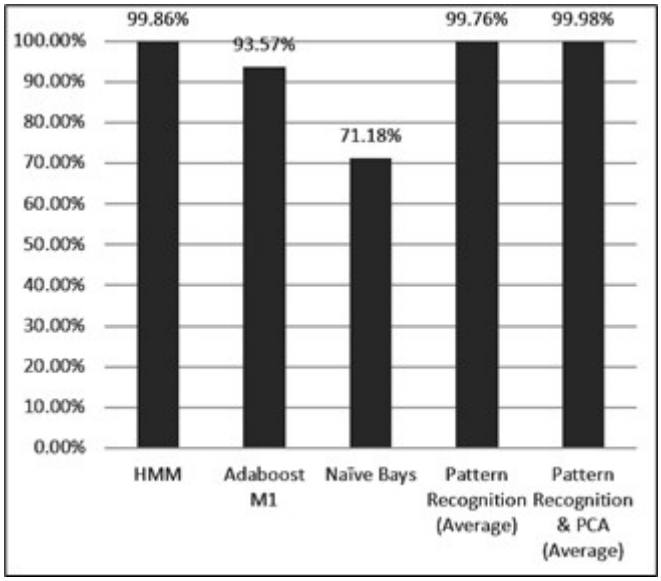

Figure 13: Diagram of comparing the accuracy of the proposed methods with some other methods

Table 3: Comparison of the accuracy of the proposed methods with some other methods.

Method	ACC (percent)
HMM	99.86
Adaboost M1	93.57
Naïve Bays	71.18
Pattern Recognition (Average)	99.76
Pattern Recognition & PCA (Average)	99.98

Also, in comparison with the research that was conducted in January 2022 [2], and part of its data set is the same as the data set of this research, we can present Fig.14. The chart of Fig.14. shows the comparison between the percentage of correctly identified anomalies by the proposed methods of the present paper with the "Machine Learning Method", "Entropy Analysis Method", "Signature Analysis Method" and "Complex Method". Based on the results presented in Fig.14. which are also summarized in Table 4, it can be concluded that the method of pattern recognition based on the neural network, both in the

approach of implementing the feature extraction phase and in the approach without implementing the feature extraction phase, has higher accuracy in the field of computer networks anomaly detection.

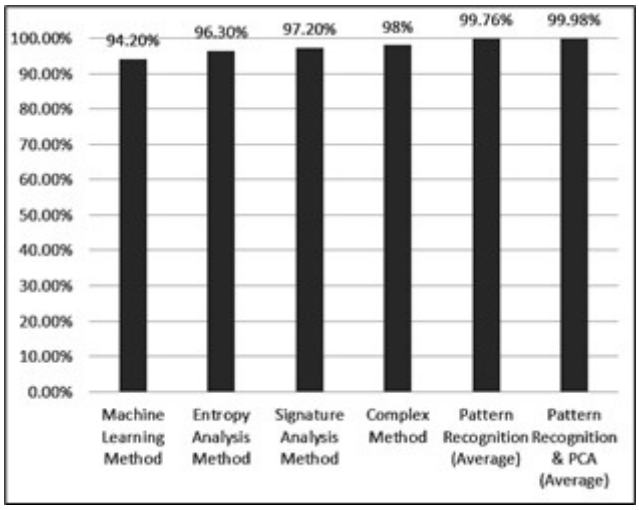

Figure 14: Diagram of comparing the correctly identified anomalies by the proposed methods with some other methods

Table 4: Comparison of the correctly identified anomalies by the proposed methods with some other methods.

Method	ACC (percent)
Machine Learning Method	94.20
Entropy Analysis Method	96.30
Signature Analysis Method	97.20
Complex Method	98
Pattern Recognition (Average)	99.76
Pattern Recognition & PCA (Average)	99.98

5. Conclusion and Suggestions

Computer networks anomaly detection is one of the most significant challenges that specialists in this field face to maintain the normal characteristics of the network and protect the information. In this regard, the proposed approach in this paper that uses pattern recognition based on neural networks succeeded in detecting anomalies in computer networks with 99-100% accuracy. Also, due to the very high speed after feature reduction using PCA algorithm, the total time required to perform the proposed method is less than 10 seconds, which makes it possible to use this method in online systems.
In order to enhance the proposed method of this research, considering some weaknesses of the algorithms used in this research, it can be combined with other algorithms. For example, the genetic algorithm or some other metaheuristic methods can be replaced with the PCA algorithm in the feature extraction phase. Moreover, since the detection of anomalies in computer networks belongs to the diagnosis problems, these anomalies can be detected through a fuzzy inference system [24] that can be the subject of future researches.

References

[1] Aha, D. W., Kibler, D., Albert, M. k. (1991). Instance-based learning algorithms. Machine Learning, 6, 37-66. 2.1
[2] Alghawli, A. S. (2022). Complex methods detect anomalies in real time based on time series analysis. Alexandria Engineering Journal, 61(1), 549-561. 1, 4.3

[3] Alhaidari, S., Alharbi, A., Alshaikhsaleh, M., Zohdy, M., Debnath, D. (2019). Network traffic anomaly detection based on viterbi algorithm using SNMP MIB data. In Proceedings of the 2019 3rd International Conference on Information System and Data Mining (ICISDM 2019), 92-97. 1, 4.3
[4] Al-Kasassbeh, M., Al-Naymat, G., Al-Hawari,E. (2016). Towards generating realistic SNMP-MIB dataset for network anomaly detection. International Journal of Computer Science and Information Security (IJCSIS), 14(9). 3
[5] Almseidin, M., Alkasassbeh, M., Kovacs, S. (2018). Fuzzy rule interpolation and SNMP-MIB for emerging network abnormality. ArXiv. 1
[6] Al-Naymat, G., Hussain, H., Al-Kasassbeh, M., Al-Dmour, N. (2021). Accurate detection of network anomalies within SNMP-MIB data set using deep learning. International Journal of Computer Applications in Technology, 66(1), 74-85. 1
[7] Analoui, M., Fadavi Amiri, M. (2008). Feature reduction of nearest neighbor classifiers using genetic algorithm. International Journal of Computer and Information Engineering, 2(5). 2.1
[8] Chan Phooi Mng, J., Mehralizadeh, M. (2016). Forecasting east asian indices futures via a novel hybrid of wavelet-PCA denoising and artificial neural network models. PLOS ONE. 2
[9] Fernat'ndez, M., Camacho, J., Magat'n-Carriot'n, R., Garct'a-Teodoro, P., Theron, R. (2018). Ugr16: a new dataset for the evaluation of cyclostationarity-based network IDSs. Computers & security, 73, 411424. 3
[10] Fukunaga, K. (1990). Introduction to Statistical Pattern Recognition. Second Edition. Academic Press. 2.1
[11] Gradient descent backpropagation: http://matlab.izmiran.ru/help/toolbox/nnet/traingd.html 2.4
[12] Gradient descent with adaptive learning rate: http://matlab.izmiran.ru/help/toolbox/nnet/traingda.html 2.4
[13] Gradient descent with momentum: http://matlab.izmiran.ru/help/toolbox/nnet/traingdm.html 2.4
[14] Gradient descent with momentum and adaptive learning rate: http://matlab.izmiran.ru/help/toolbox/nnet/traingdx.html 2.4
[15] Gogoi, P., Bhattacharyya, D. K., Borah, B., Kalita, J. K. (2011). A survey of outlier detection methods in network anomaly identification. The Computer Journal, 54(4), 570-588. 1
[16] Jesan, J. (2004). The neural approach to pattern recognition. ACM Digital Library, 2004. 2.1, 2.3
[17] Johnson, D. (2023). Unsupervised machine learning: Algorithms, types with example. Guru99. 1
[18] Jolliffe, I. T. (1986). Principal component analysis. Springer Series in Statistics. 2.2
[19] Karamizadeh,S., Abdullah, S.M., Manaf, A.A., Zamani, M., Hooman, A. (2013). An overview of principal component analysis. Journal of Signal and Information Processing, 4, 173-175. 3.2
[20] Levenberg-Marquardt backpropagation: http://matlab.izmiran.ru/help/toolbox/nnet/trainlm.html 2.4
[21] Monshizadeh, M., Khatri, V., Atli, B. G., Kantola, R., Yan, Z. (2019). Performance evaluation of a combined anomaly detection platform. IEEE Access 7, 100964100978. 3
[22] Nassif, A. B., Talib, M. A., Nasir, Q., Dakalbab, F. M. (2021). Machine learning for anomaly detection: A Systematic review. IEEE Access, 9, 78658-78700. 1
[23] Pu, G., Wang, L., Shen, J., Dong, F. (2021). A hybrid unsupervised clustering-based anomaly detection method. In Tsinghua Science and Technology, 26(2), 146-153. 1
[24] Robles, E.O., Melin, P. (2019). A hybrid design of shadowed type-2 fuzzy inference systems applied in diagnosis problems. Engineering Applications of Artificial Intelligence, 86, 43-55. 5
[25] Thompson, N. C., Greenewald, K., Lee, K., Manso, G. F. (2022). The computational limits of deep learning. ArXiv. 1
[26] Tsymbal, A., Puuronen, S., Pechenizkiy, M., Baumgarten, M., Patterson, D. (2002). Eigenvector-based feature extraction for classification. FLAIRS-02 Conference Proceedings. 2.1, 2.2, 2.2
[27] Wang, Q. (2015). Computer network fault diagnosis based on neural network. International Journal of Future Generation Communication and Networking, 8(5), 39-50. 2.4, 2.5, 2.5