

Evaluating Security Anomalies by Classifying Traffic Using a Multi-Layered Model

Mohammadreza Samadzadeh^{*a}, Najmeh Farajipour Ghohroud^b

Department of Computer Iranians University an E-Institute of Higher Education, Tehran, Iran; samadzadeh@ut.ac.ir^a, n.farajipour@iranian.ac.ir^b

ABSTRACT

Accurate traffic classification is important for various network activities such as accurate network management and proper resource utilization. Port-based approaches, deep packet inspection, and machine learning are widely used techniques for classifying and analyzing network traffic flows. Most classification methods are suitable for small-scale datasets and cannot achieve a high classification accuracy owing to their shallow learning structure and limited learning ability. The emergence of deep learning technology and software-driven networks has enabled the application of classification methods for processing large-scale data.

In this study, a two-step classification method based on deep learning algorithms is presented, which can achieve high classification accuracy without manually selecting and extracting features. In the proposed method, an Autoencoder was used to extract features and remove unnecessary and redundant features. In the second step, the proposed method uses the features extracted by the autoencoder from a hybrid deep-learning model based on the CNN and LSTM algorithms to classify network traffic.

To evaluate the proposed method, the results of the proposed two-stage hybrid method is compared with comparative algorithms including decision tree, Naïve Bayes, random forest. The proposed combined CNN+LSTM method obtains the best results by obtaining values of 0.997, 0.972, 0.959, and 0.964, respectively, for the evaluation criteria of, accuracy, precision, recall, and F1 score.

The proposed method is a practical and operational method with high accuracy, which can be applied in the real world and used in the detection of security anomalies in networks using traffic classification and network data.

Keywords— Network Traffic Classification, Deep Learning, Software-oriented Network, Autoencoder.

1. Introduction


Network traffic identification is an essential function for network systems, which facilitates accurate management through the classification of network traffic flows [1]. Currently, there are many approaches for identifying and predicting network traffic, including port-based approaches, deep packet inspection, and machine learning [1,2,3]. Port-based approaches are also known as payload-based approaches. Today, due to the prevalence of dynamic port numbers in applications, these approaches are no longer effective [4]. The deep packet inspection approach is an expensive approach due to the high computational cost [5] and also the inability to inspect encrypted traffic. Therefore, to face these challenges, more intelligence should be used in network tools. Traditional computer networks consist of a large number of forwarding equipment, i.e., routers and/or switches, which are operated by many protocols and run a wide range of applications. The existence of this heterogeneity in infrastructure complicates network management and performance optimization. Traditional networks are distributed systems where each forwarding device maintains a local view of the entire network. Therefore, using machine learning techniques in a system whose elements have limited visibility is another big challenge [6].

Software-defined networking is an architecture that separates control and data traffic. This architecture consists of three layers including the data, control, and application layer. SDN has three APIs namely north, south, and east-to-west. The Northbound API is an interface for connecting network and control layer applications. Southbound API connects data and control layer in SDN [7]. SDN architecture has certain challenges and limitations in the field of security, scalability, and supportability. The feature of separating the data level from the control level in these networks has brought new advantages and challenges to researchers.

Various methods have been developed and presented to deal with attacks and intruders on computer systems and networks, which are known as intrusion detection methods. Detecting and identifying misuse and unauthorized use of computer network systems and resources by users is one of the main goals of intrusion detection methods. Intrusion detection systems form effective and efficient classification patterns and models for detecting and identifying normal behaviors from suspicious, and abnormal behaviors.

2. Related works

Kalkan et al. [8] proposed a joint entropy-based scoring system (JESS) to detect and mitigate DDoS attacks. They use

 <http://dx.doi.org/10.22133/ijwr.2023.396115.1151>

Citation M. Samadzadeh, N. Farajipour Ghohroud, "Evaluating Security Anomalies by Classifying Traffic Using a Multi-Layered Model," *International Journal of Web Research*, vol.6, no.1 ,pp.17 -28, 2023, doi: <http://dx.doi.org/10.22133/ijwr.2023.396115.1151>.

**Corresponding Author*

Article History: Received:6 March 2023; Revised:18 May 2023; Accepted: 6 June 2023

Copyright © 2022 University of Science and Culture. Published by University of Science and Culture. This work is licensed under a Creative Commons Attribution-Noncommercial 4.0 International license(<https://creativecommons.org/licenses/by-nc/4.0/>). Noncommercial uses of the work are permitted, provided the original work is properly cited.

shared entropy as a Criterion to detect DDoS attacks without significantly increasing the workload on switches.

Lima et al [9] introduced a method to more effectively protect the network against DDoS attacks through statistical analysis of traffic entropy. They presented a model in Mininet for verification.

Kumar et al. [10] presented a solution, which can effectively detect and mitigate SYN flood attacks in SDN, which starts by calculating the entropy of the destination IP addresses, then uses a set of TCP flags chosen as random variables, and finally identifies the attacker through adaptive thresholding.

Peng et al. [11] proposed an abnormal traffic detection algorithm, DPTCM-KNN. This algorithm can effectively improve the abnormal flow detection accuracy while reducing the false alarm rate in the DDoS detection process. Although many researchers have proposed various solutions based on machine learning algorithms for DDoS detection in SDN, these methods still have problems in accuracy and efficiency.

X.-D. Zang et al. [12] started with flow characteristics and proposed a set of more accurate and comprehensive flow indicators. The authors extracted 9 single features and 39 dual features from different dimensions such as time, location, category, and intensity to form the spectrum of IP address traffic behavior features. Fine-grained traffic features greatly improve detection accuracy.

Kho et al. [13] improved the DDoS detection algorithm. They proposed a DDoS detection method based on K-FKNN and a detection system to improve detection efficiency and accuracy. In addition, some researchers have reduced the channel overhead between the data plane and the control plane by improving the flow data collection method and improving detection efficiency.

Kokila et al. [14] used a support vector machine (SVM) to classify DDoS attacks with high accuracy and low false positive rate. The SVM classifier was compared with other classifiers used for DDoS attack detection and SVM provided more accurate classification than other techniques. Real-time detection of DDoS and integration of traffic patterns built in SVM with SDN controller was their future work.

In the paper presented by Van et al. [15], fuzzy logic is used to detect real DDoS attack traffic in SDN. The authors have solved the existing problems of the OpenFlow protocol. They proposed a fuzzy logic-based DDoS mitigation algorithm that uses multiple criteria for DDoS detection. Their system demonstrated the ability to detect and filter 97% of attack streams with a false positive rate of 5%. They would like to extend the OpenFlow protocol to achieve robust and faster performance.

In this research, some important methods in the field of anomaly detection systems in software-based networks are studied and investigated. All methods try to discover unusual traffic in a short time and with the highest detection accuracy. The possibility of using the anomaly detection system in large and high-speed networks is one of the most important challenges in the field of security in these networks. But the discussed methods are not very accurate in large networks and high volume of traffic. Most classification methods are suitable for small-scale datasets and cannot achieve high classification

accuracy due to their shallow learning structure and limited learning ability. The emergence of deep learning technology and software-based networks enable applied classification methods to process large-scale data. Deep learning is a promising approach for classifying and predicting big data to extract hidden patterns such as traffic characteristics. Therefore, in this research, the classification of network traffic using deep learning algorithms is used to detect and reduce security anomalies in software-based networks using network traffic classification. In this regard, feature extraction from network traffic data is first done using the self-encrypting deep learning algorithm, to reduce the dimensions of the data and remove redundant features. In the second step, the features extracted by the autoencoder will be used as input for classification based on deep learning algorithms.

3. Method

Machine learning is a data analysis method that identifies internal patterns and makes decisions based on the collected information. Machine learning algorithms can be studied in two ways, supervised learning and unsupervised learning. Supervised learning uses labeled training data. But in unsupervised learning, unlabeled training data is used, this method tries to extract information through classification based on the degree of similarity in observation points [16].

In this section, the proposed method is a two-step method based on feature extraction with a deep Autoencoder, and classification with a proposed hybrid method based on CNN and LSTM deep learning algorithms. This method is proposed to detect and reduce security anomalies in software-based networks using network traffic classification and the different parts and steps of the proposed two-step method are described. The proposed two-step method consists of different sections and steps, for better understanding, the flowchart of the proposed two-step method is shown in Figure 1.

According to the flowchart of the proposed method shown in Figure 1, the purpose of this research is to detect and reduce security anomalies in software-based networks. This method is based on network traffic classification using a two-stage system based on feature extraction with deep self-encryption and hybrid deep learning algorithms based on CNN and LSTM. In this regard, the Moore traffic data set is first read and preprocessed. This preprocessing includes three steps: normalization, removing missing values, and converting string values to numbers if any. The target data set has 248 features, which in the first stage uses a deep Autoencoder to reduce the dimension and remove redundant features. New features are extracted so that the dimension of 248 reduces to 100 new features after dimension reduction.

This action removes redundant features, which lowers the classification accuracy. In the second step, the new extracted and reduced dimensionality features are divided into two parts, training and testing, and are used as input to the deep learning model based on CNN+LSTM algorithms. The training dataset is used to train and learn the model, and the test dataset is used to evaluate the performance of the model on the test data. In the current research, 20% of the data is considered as test data and 80% of the data is used to train the model. The results obtained by the proposed hybrid method based on CNN+LSTM deep learning algorithms are compared with CNN and LSTM deep learning algorithms and decision tree, Naïve Bayes, random forest, and AdaBoost machine learning

algorithms. To evaluate the comparative criteria of accuracy, correctness, recall, and f1 score, the obtained results are reported in the form of a graph.

3.1. Introduction of the dataset used

The proposed method in this research has been evaluated on the real traffic data set. In this regard, the Moore data set will be used, which was proposed by the computer laboratory at the University of Cambridge and it has been used in many research works related to traffic classification. This dataset consists of several separate datasets collected during different intervals of the day, and each dataset contains only TCP traffic flows. In each data set, for each TCP traffic flow, 248 flow attributes such as flow size, flow duration and the corresponding class label of that request are recorded. All network flows are classified into 10 different classes as shown in Table 1.

3.2. Preprocessing

Before processing the data, it is necessary to perform pre-processing on it, so that it can be used in other steps. Because in many cases, due to the existence of non-numeric values, the existence of missing values, and non-normality of the data, the data cannot be used directly. In this section, the pre-processing performed on the data is described.

In many cases, the data set may have missing values. In data analysis, sometimes some observations are considered missing for various reasons, that is, there is no valid value for one or more characteristics of that data. How to deal with these observations in data analysis is very important, because of the importance of their results, especially in sensitive decisions. To overcome the problem of missing data, the most common method is to delete the missing data. Therefore, in the proposed two-step method, the data set has been examined and if there are missing values, these values are removed.

Dataset standardization is a common requirement for many machine learning algorithms. A machine learning algorithm may behave badly if a single feature is not more or less similar to the standard normally distributed data. For example, many elements used in the objective function of learning algorithms assume that all features are centered at zero and have the same variance. If one feature has a larger variance than the others, it may dominate the objective function and the algorithm cannot learn from the other features as expected. In the proposed two-step method for normalization, Min-Max scaling is used to change the data to the interval between zero and one so that all columns become in the same interval, and the algorithm should not be biased toward larger values. For this purpose, the MinMaxScaler library from the Sklearn software package was used.

3.3. Deep learning models used in the proposed method

As mentioned earlier, a two-step deep learning method was used in this research to classify network traffic, in which the Autoencoder was first used to extract features. And then in the second step, the output of the Autoencoder, i.e., the extracted features, is fed as input to the proposed combined deep learning model based on CNN and LSTM. In the following, the description and explanation of the details of each of the models are given and the architecture of the implemented models is described.

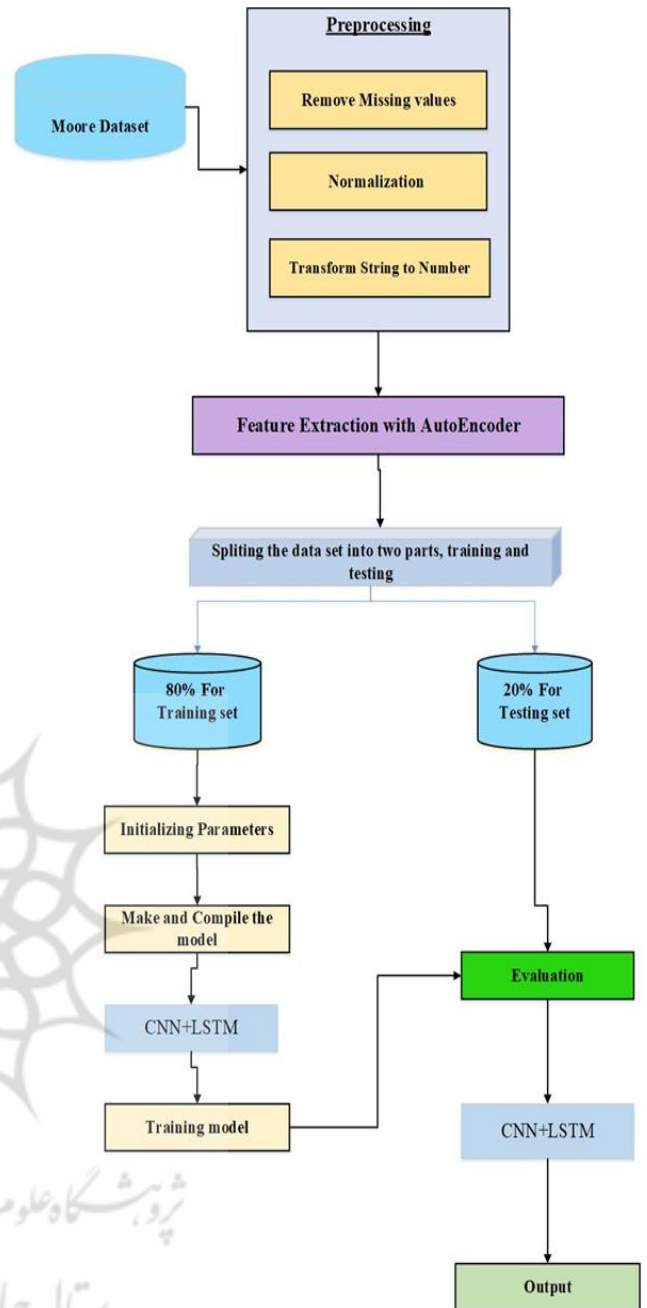


Figure 1. Flowchart of the proposed two-step method

Table 1. Classification of network flows

Traffic class	Applications and related programs
WWW	HTTP, HTTPS
MAIL	POP, SMTP, IMAP
FTP-CONTROL	FTP
FTP-PASV	FTP
FTP-DATA	FTP
ATTACK	Worms, Viruses, Port Scans
P2P	Kazaa, BitTorrent, Gnutella
DATABASE	postgres, mysql, oracle, ingres
MULTIMEDIA	Audio and video playback
SERVICES	DNS, IDENT, NTP

3.4. Autoencoder model used for feature extraction

In the proposed two-stage method, in the first stage, the features of the data related to the network flow are extracted by an Autoencoder. The extracted features are fed to the next stage as the input of the selected classification, which is a combination of CNN and LSTM algorithms in the proposed method. In the following, the details of the Autoencoder and the implemented Autoencoder architecture are described. Autoencoders [17] are unsupervised artificial neural networks that are used to generate new data through a sequence of encoding and decoding processes. The first is implemented by an encoder dedicated to compressing the input into a space of hidden variables. While the latter is implemented by a decoder that is involved in reconstructing the input based on the information contained in the hidden variables. For this purpose, the encoder-decoder pair is implemented by two symmetric neural networks. An autoencoder network has the same input and output dimensions. This network transforms the input into a hidden profile and then reconstructs the input from this hidden profile. Recently, autoencoders are used as production data models, where the input data is converted into an abstract form and then decoded to the original data by estimating the same function. One of the advantages of autoencoders is extracting useful features and discarding unnecessary features [18]. The autoencoder architecture consists of neural networks with one or two hidden layers such as multilayer perceptron. The main purpose of Autoencoder networks is input reconstruction. While in multilayer perceptron, the network tries to predict the desired output based on specified inputs. In autoencoders, the number of input and output nodes must be the same. In the encoding process, the input vector x is mapped to h with the transformation matrix W in the hidden layers. Then, in the decoding process, the Autoencoder reconstructs the vector x^{\wedge} with the new weight matrix W^{\wedge} . Mathematically, if $x=x^{\wedge}$, then: $W^{\wedge}=WT$. Figure 2 shows the general architecture of Autoencoder.

According to the above descriptions, encryption can be defined as Equ(1):

$$h = W^{(1)}x + b^{(1)} \quad (1)$$

Then, in the hidden layer h , the input is reconstructed as Equ (2):

$$y = g(W^{(2)}h + b^{(2)}) \quad (2)$$

Usually, the encoding and decoding layers are non-linear.

The main parameters of Autoencoders are $\theta = \{W^{(1)}, b^{(1)}; W^{(2)}, b^{(2)}\}$. If the encoding and decoding were appropriate, the cost function J_{θ} should have a minimum value.(Equ(3))

$$J_{\theta} = \min \frac{1}{m} \sum_{i=1}^m (y^{(i)} - x^{(i)})^2 \quad (3)$$

That $x^{(i)}$ is the i -th training sample.

Autoencoder training is done in two stages [19]. Unsupervised learning and fine-tuning of network weights. One of the most important factors in the first stage is the appropriate selection of the activation function. In this

research, the relu activation function is used. In the training phase, forward propagation is applied to each input to calculate the output value, and then the derivative x' of x is calculated. In the final step, the error is back-propagated through the network to update the weights. In the network optimization stage, standard learning methods and gradient descent algorithms are usually used to change the parameters of each layer. But this algorithm is one of the slowest optimizers. In this research, AdamOptimizer is used, which is mostly used in deep learning. AdamOptimizer uses Adam algorithm to control the learning rate. The advantage of this algorithm compared to the gradient descent algorithm is the use of the moving average of the parameters (momentum), which helps the Adam algorithm to use larger and more effective steps without the need for precise adjustment [20,21]. The implemented Autoencoder architecture is shown in Figure 3.

As shown in Figure 3. In the proposed method, an automatic encoder with a 2-layer encoding network and a 2-layer decoding network is used. The dimension of the hidden layer is set equal to 100. This value was set by testing different values and selecting the optimal value. They have an encoding dimension of 100, the input with dimensions of 248 is compressed by passing through the autoencoder layers, and finally, it is coded into 100 new features in the middle layer. That is, in the designed self-encoder, the encoded profiles will have dimensions equal to 100. In other words, redundant features are removed by the autoencoder, and 248 network traffic data features are coded into 100 new features. In the following, the classifier used to classify network traffic is described.

3.5. Combined CNN+LSTM model used for network traffic classification

In order to take advantage of both CNN and LSTM deep learning algorithms to extract spatiotemporal features. We extract location-dependent features by CNN and extract time-dependent features by LSTM. The architecture of the proposed CNN+LSTM hybrid model is shown in Figure 4.

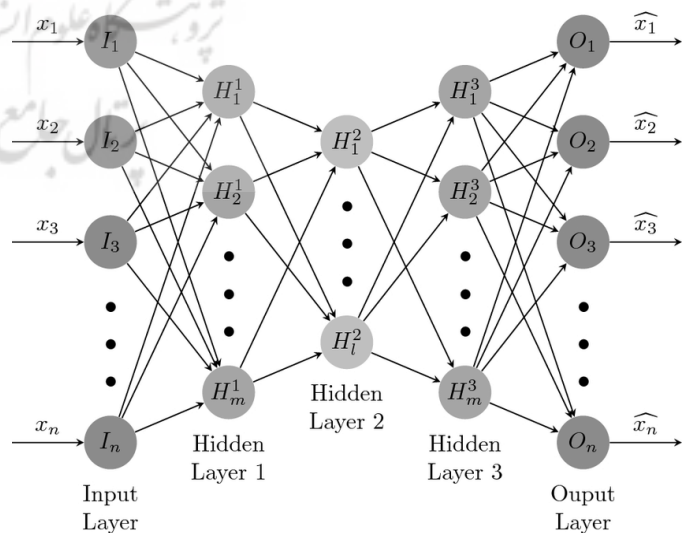


Figure. 2. The general structure of an Autoencoder

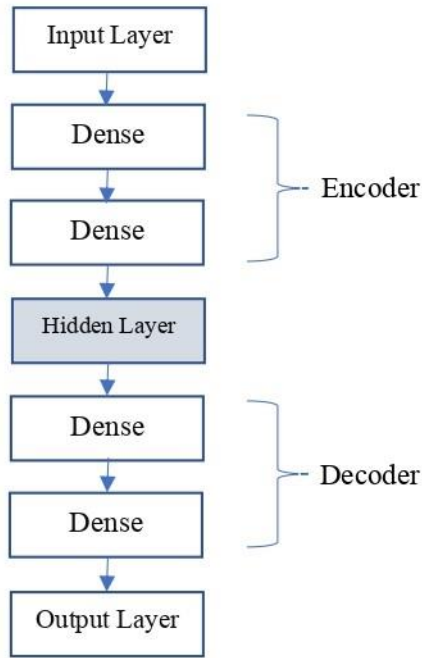


Figure 3. Figure 3: Autoencoder architecture implemented

3.6. Evaluation criteria

Three criteria of accuracy, precision, recall, and F1 score will be used to evaluate the proposed two-step method. To define these criteria, it is necessary to first define the Confusion Matrix. Because these criteria are obtained based on the Confusion Matrix Which is shown in Table 2. A confusion Matrix is a table that is often used to describe the performance of a classification model on a set of test data whose true values are known. The Confusion Matrix is expressed based on the following concepts:

True positive (TP): Data samples from the network flow that the algorithm predicted as attacks, and in reality, these samples are attacks.

True negative (TN): Data samples that the algorithm has predicted as normal data flow of the network, and in reality, this flow and data samples are normal.

False Positive (FP): Samples that the algorithm has mistakenly predicted as attacks, but in reality, these samples are normal network flow samples.

False negative (FN): samples that the algorithm has mistakenly predicted as normal samples, but in reality, these samples are attack samples, in other words, it is equal to an attack on the network that is not detected.

Accuracy: The ratio of the number of correctly classified requests to the total number of requests. This measure is used to evaluate the accuracy of the applied classifier in the entire data set. The accuracy of the classification focuses more on correctly distinguishing the positive class from the negative one, determines the overall performance of the classifier, and shows how many of the total test cases are correctly classified by the classifier. Accuracy is expressed as Equ(4).

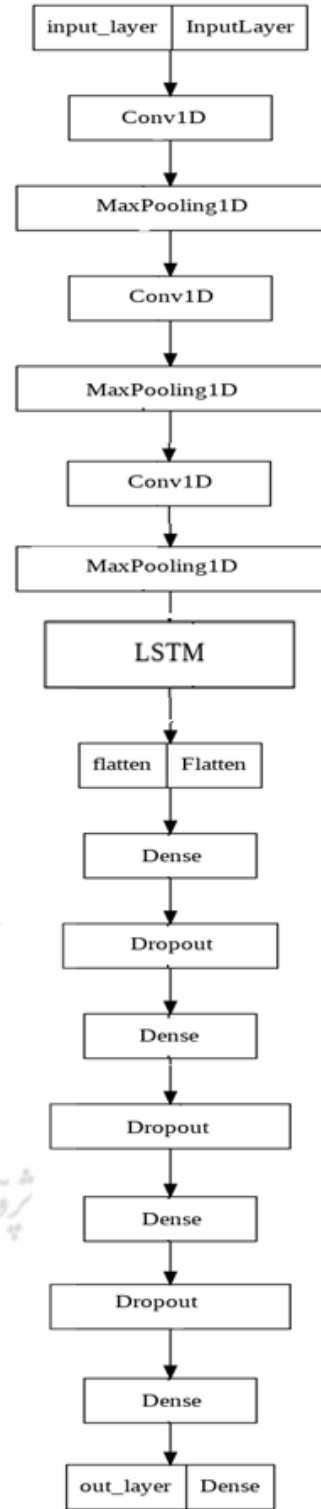


Figure 4. A hybrid CNN+LSTM model architecture is implemented

Table 2. Confusion Matrix

		Proposed class	
		Positive	False
Real class	Positive	True positive (TP)	False negative (FN)
	False	False Positive (FP)	True negative (TN)

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

Precision: Precision is the percentage of requests that are correctly classified in a given class. This criterion measures the reliability of the classification and expresses the ratio of real attacks among the warned attacks and is expressed as Equ(5).

$$\text{Precision} = \frac{TP}{TP + FP} \quad (5)$$

Recall: Recall is the percentage of requests belongs to the class i which are correctly classified into the specified class i . Recall, also known as sensitivity and true positive rate, represents the proportion of true attacks detected by the model. the recall is expressed in the form of Equ(6).

$$\text{Recall} = \frac{TP}{TP + FN} \quad (6)$$

F1 score: This is the weighted average of the true positive rate (recall) and precision. the recall is expressed in the form of Equ(7).

$$F1 \text{ score} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (7)$$

4. Results

In this section, the results of the implementation of the proposed two-step method and the comparative algorithms implemented on the Moore network traffic data set are discussed. For this purpose, using Autoencoder, features have been extracted from the data. In the next step, the classification of network data is done using the combined method based on CNN+LSTM deep learning algorithms. The results of the implementation of the proposed hybrid method have been compared with the deep learning algorithms of CNN and LSTM and the machine learning algorithms such as the decision tree, Naïve Bayes, random forest, and AdaBoost, and the results of the execution of each of the algorithms are reported and compared in terms of precision, recall, and f1 score.

4.1. Evaluation of the proposed two-step method

In this section, the proposed two-step method is implemented on the Moore data set, and the results obtained by the proposed hybrid model based on CNN+LSTM and comparative algorithms are reported based on the criteria of accuracy, precision, recall, and F1 score. In the following, the accuracy and loss diagram of the proposed combined method during training and validation phase is shown in Figure 5.

As it can be seen from the diagram in Figure 5, the Loss diagram of the model decreases during training with a uniform and continuous process without much fluctuation, and after 150 repetitions, it remains almost constant and finally reaches a value of 0.0027, which is close to zero. Also, according to the obtained results, which can be seen in the figure, during the training of the proposed hybrid model, the accuracy value increases and finally, the accuracy value of 0.9991 is obtained

from the training data. In the following, the results of the proposed hybrid model based on CNN+LSTM and comparative algorithms have been evaluated on the test data using the criteria of precision, recall, and F1 score.

4.2. Evaluation and comparison based on precision criteria

Figure 6 shows the evaluation of the proposed two-step method and comparative algorithms on the Moore data set based on the precision criterion.

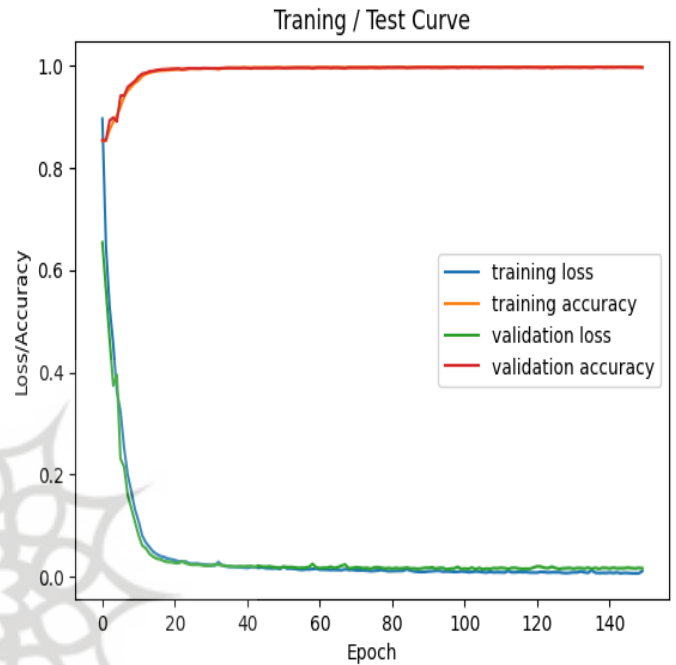


Figure 5. Accuracy and Loss diagram of the proposed model

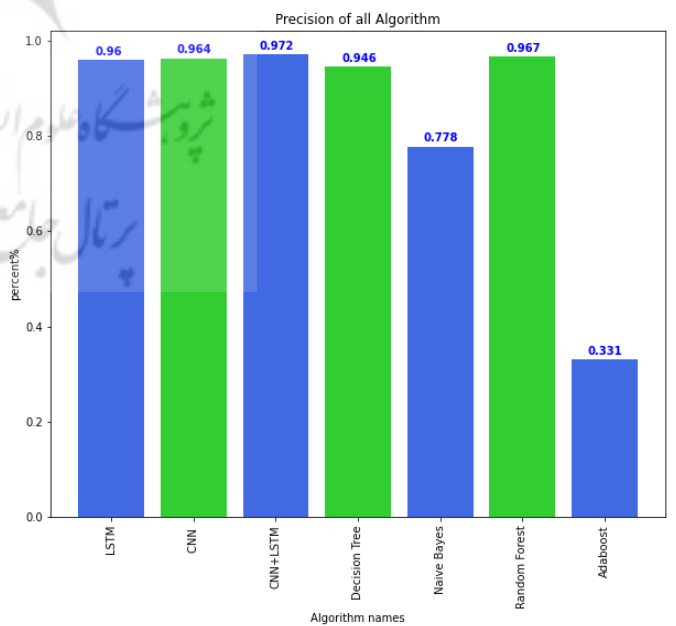


Figure 6. Evaluation of the proposed method and comparative algorithms based on precision criteria

As shown in Figure 6, the combined model based on CNN+LSTM has achieved the best performance with a precision value of 0.972. The random forest algorithm is in the next place with a precision value of 0.967. Adaboost algorithm also obtained the weakest result among the comparative algorithms by obtaining a precision value equal to 0.331. In the following, the evaluation of the proposed two-step method and comparative algorithms based on the recall criteria has been discussed.

4.3. Evaluation and comparison based on the recall criterion

The evaluation results of the proposed method and comparative algorithms based on the recall criteria are shown in Figure 7.

Based on the evaluation and the obtained results, which can be seen in Figure 7, the combined CNN+LSTM method has obtained the highest recall value of 0.959 compared to the comparative algorithms. Also, the decision tree algorithm has achieved a better performance than other algorithms by obtaining a value of 0.95 for the recall criterion and is placed in the next position. The weakest result is related to the Adaboost algorithm with a recall value of 0.258. In the following, the proposed method and comparative algorithms are evaluated based on the F1 score criterion.

4.4. Evaluation and comparison based on F1 score criteria

The evaluation results of the proposed combined CNN+LSTM model and comparative algorithms based on the F1 score criterion are shown in Figure 8. According to the evaluation results shown in Figure 8, the proposed CNN+LSTM model has the best performance and has performed better than other algorithms by obtaining F1 score equal to 0.964. The CNN algorithm is also placed in the next position with a F1 score value equal to 0.957. Adaboost algorithm also obtained the weakest result among the comparative algorithms by obtaining the value of F1 score equal to 0.198.

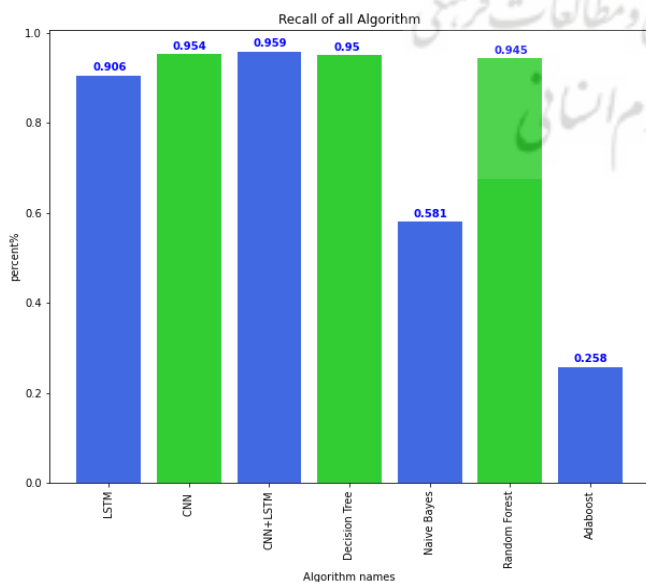


Figure 7. Evaluation of the proposed method and comparative algorithms based on the recall criteria

In the following, the evaluation of the algorithms and the obtained results are summarized.

4.5. Evaluation and comparison based on the Confusion Matrix

The evaluation results of the proposed combined CNN+LSTM model and comparative algorithms based on the Confusion Matrix are given in this section. In the Confusion Matrix, the vertical column shows the real class of each data and the horizontal column shows the class predicted by the model. The diameter of the Confusion Matrix also shows the correctly predicted values, that is, it shows examples of each class that the algorithm correctly recognized and classified as part of the same class. In Figure 9, the Confusion Matrix of the LSTM model is shown.

As shown in Figure 9, the algorithm had good accuracy in data classification and was able to distinguish the samples of each class with good accuracy. The most error of the model in recognizing the samples, belongs to the data of ATTACK and P2P class with 62 data, which are wrongly classified. Next, the Confusion Matrix of the CNN algorithm is shown in Figure 10.

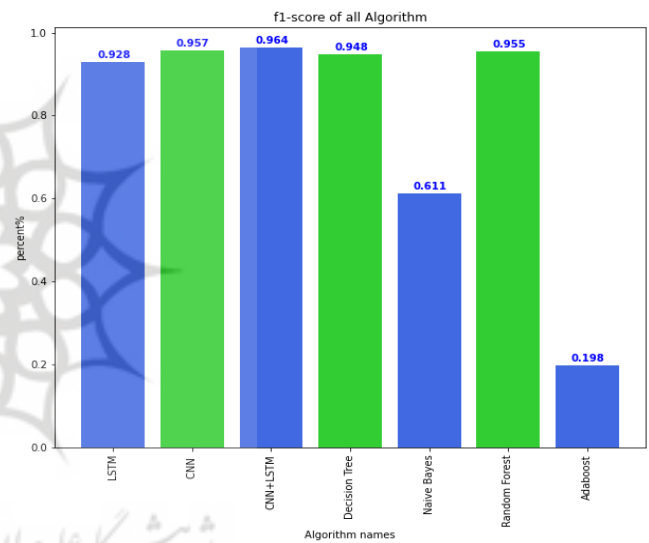


Figure 8. Evaluation of the proposed method and comparative algorithms based on F1 score criteria

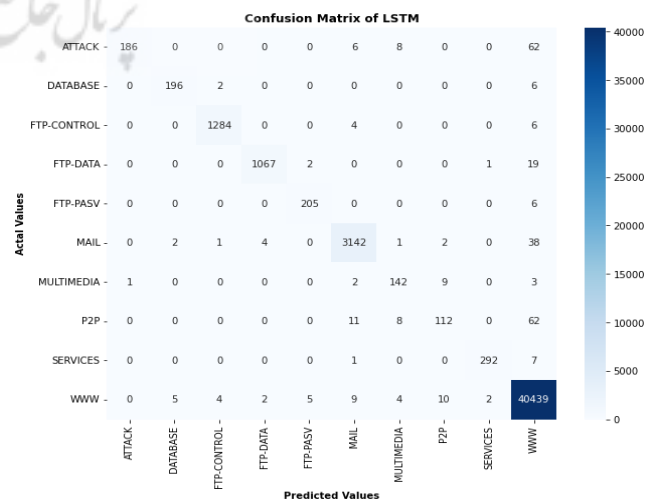


Figure 9. The confusion matrix related to the LSTM model

According to the obtained results shown in Figure 10, the CNN model has obtained better results than the LSTM algorithm and has been able to distinguish the samples of each class with good accuracy. The most error of the model in recognizing the samples belongs to the data from the ATTACK class with 44 samples, which is classified as WWW class. Next, the Confusion Matrix of the combined CNN+LSTM model is shown in Figure 11.

According to the Confusion Matrix diagram shown in Figure 11, The proposed combined CNN+LSTM method has better results than the two deep models of CNN and LSTM and other algorithms and has been able to distinguish the samples of each class with good accuracy. The most error of the model in recognizing the samples belongs to the data from the ATTACK class with the number of 38 data, which are wrongly classified as WWW class. In the following, the Confusion Matrix of the decision tree algorithm is given in Figure 12.

The confusion matrix of the decision tree algorithm is shown in Figure 12. According to the obtained results, the decision tree algorithm performed poorly compared to deep learning models and has a few wrong classifications in all classes. The most error of the decision tree algorithm in recognizing samples belongs to data from the mail class with 44 data, which are wrongly classified. Next, the Confusion Matrix of the Naïve Bayes algorithm is shown in Figure 13.

As shown in Figure 13, the Naïve Bayes algorithm performed very poorly. And data classification has many errors in such a way that it has predicted a large part of the data in the wrong class. In some classes such as DATABASE, there was not even one correct prediction. Next, the Confusion Matrix of the random forest algorithm is shown in Figure 14.

According to the obtained results, which are shown in Figure 14, the random forest algorithm had good accuracy in

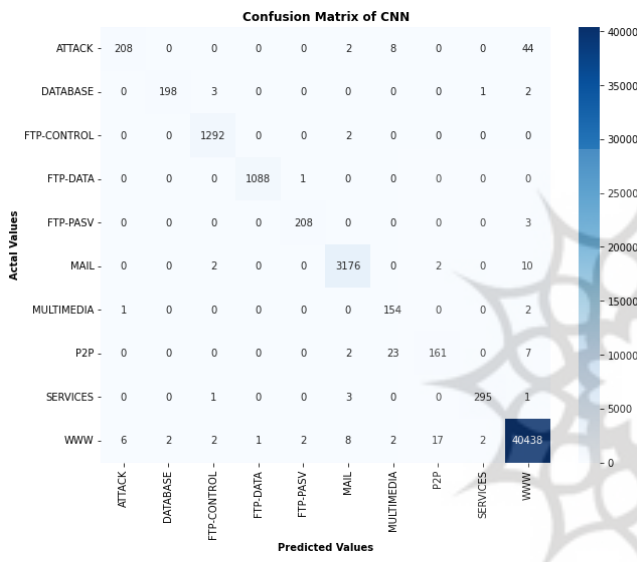


Figure. 10. Confusion matrix related to the CNN model

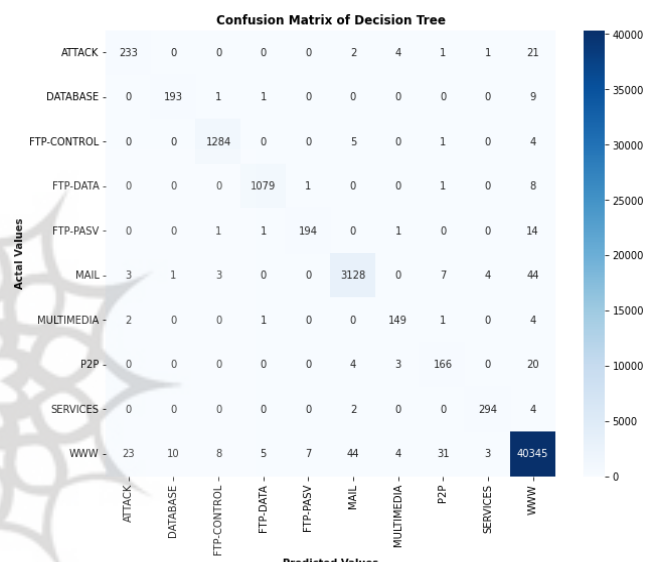


Figure. 12. Confusion matrix of decision tree algorithm

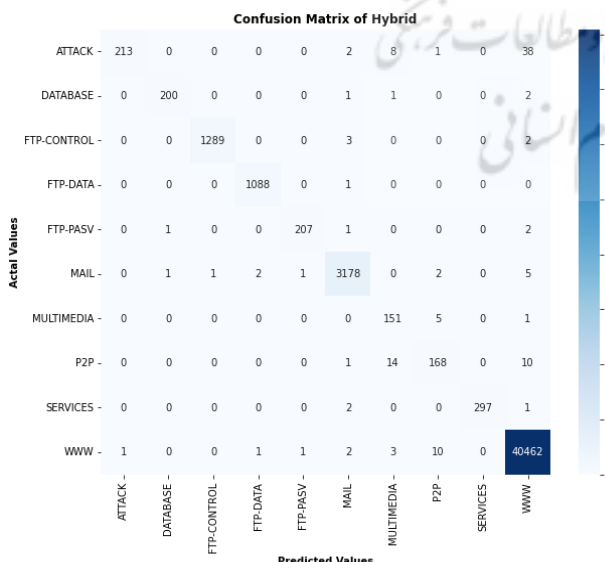


Figure. 11. Confusion matrix of CNN+LSTM model

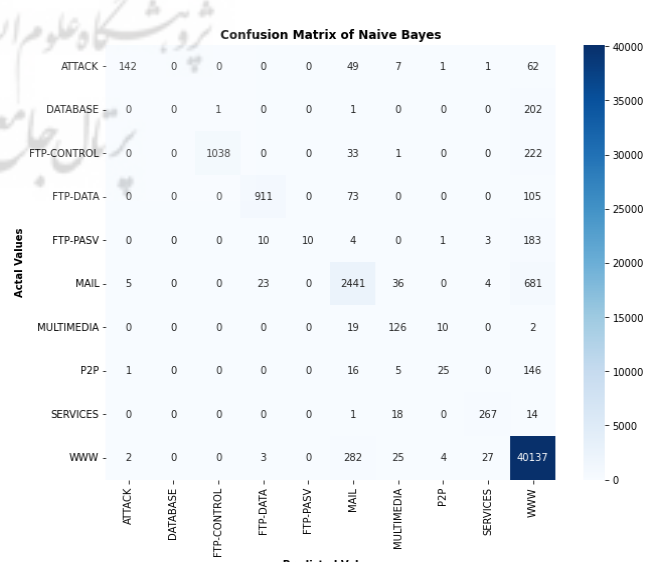


Figure. 13. Confusion matrix of Naïve Bayes algorithm

data classification. It has been able to separate the samples of each class with good accuracy and obtain results close to the combined model. The most error of the model in recognizing samples belongs to data from MAIL, ATTACK, and P2P classes with the number of 31, 27, and 26 data respectively, which are wrongly classified. Next, the Confusion Matrix of the AdaBoost algorithm is shown in Figure 15.

As shown in Figure 15, the AdaBoost algorithm has the weakest result among all algorithms. And like the Naïve Bayes algorithm, in some classes such as SERVICES, ATTACK, and DATABASE, it does not even have one correct prediction. In the following, algorithms are compared based on the AUC criterion.

4.6. Evaluation and comparison based on the AUC criterion

The evaluation results of the proposed combined CNN+LSTM method and comparative algorithms based on the AUC criterion are given in this section. The greater the area

under the graph and the AUC criterion, the better the performance of the model in data classification and recognition. Next, the AUC diagram of the LSTM model is shown separately for each class in Figure 16.

According to the evaluation results shown in Figure 16, the LSTM model has an AUC criterion value equal to 1 for FTP-CONTROL class data. which indicates the 100% correct classification of the model for this class of data. That is, the model was able to correctly recognize almost all samples belonging to this class. The lowest value of the AUC criterion in the LSTM model is also obtained for data from the P2P class with an AUC value equal to 0.79. Next, Figure 17 shows the AUC diagram of the CNN model separately for each class.

According to the obtained results shown in Figure 17, the CNN model has an AUC criterion value equal to 1 for FTP-CONTROL, FTP-DATA, and MAIL class data, which indicates the classification with high accuracy for these classes of data. The lowest value of the AUC criterion in the CNN model is also obtained for data from the P2P class with an AUC value equal to 0.92. Next, Figure 18 shows the AUC diagram of the combined CNN+LSTM model separately for each class.

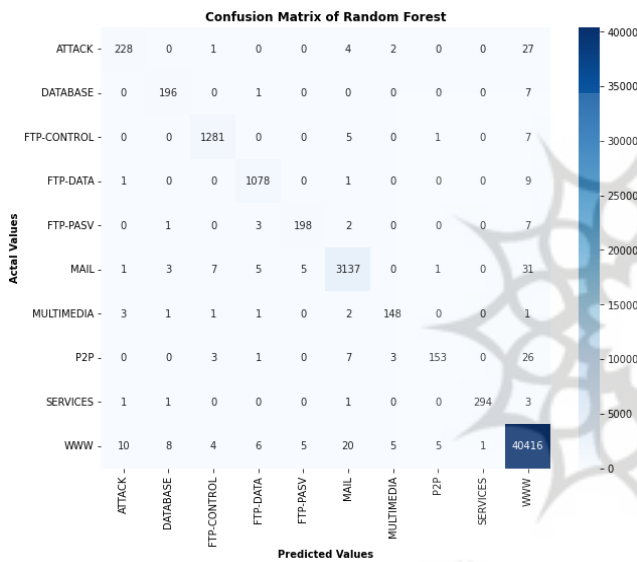


Figure. 14. Confusion matrix of random forest algorithm

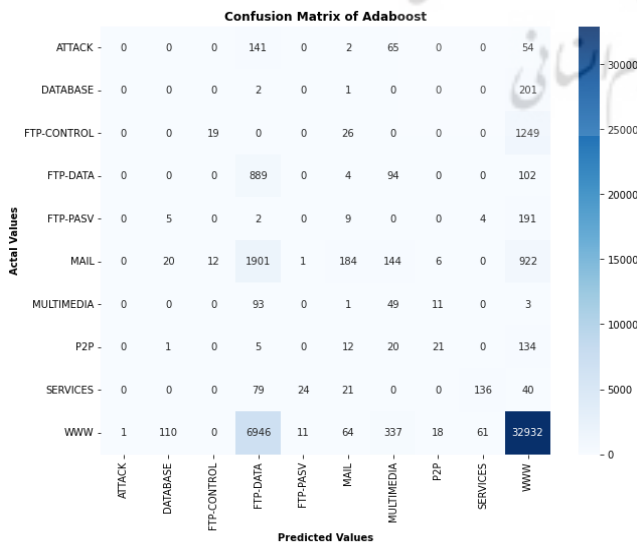


Figure. 15. The Confusion matrix of the AdaBoost algorithm

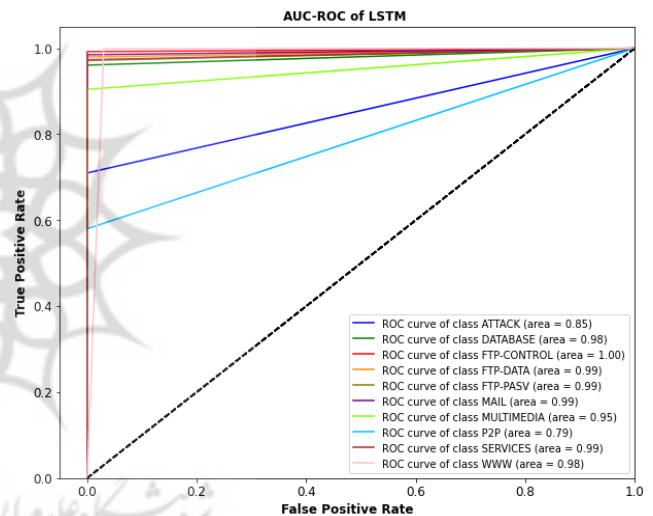


Figure. 16. Evaluation of the LSTM model based on AUC criterion

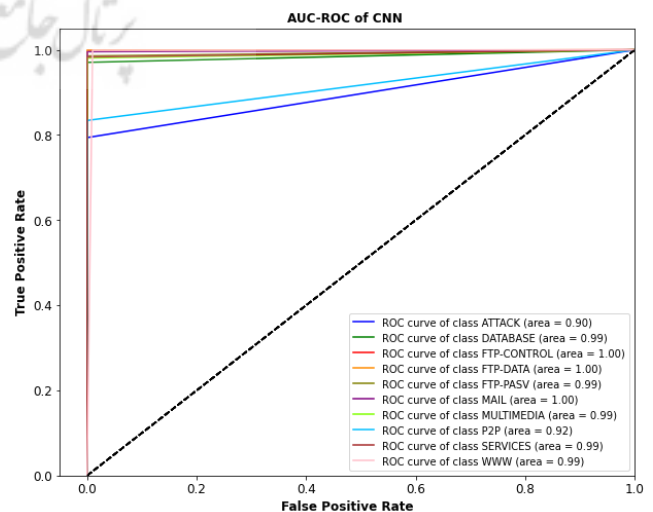


Figure. 17. Evaluation of CNN model based on AUC criterion

Figure 18 shows the results obtained by the combined CNN+LSTM prediction method. According to the results, this model has an AUC criterion value of 1 and 0.99 for most of the classes, which indicates a better classification for these classes of data. It means, the model has been able to classify samples belonging to most classes with high accuracy. The lowest value of the AUC criterion in the combined method was also obtained for data from the ATTACK class with an AUC value of 0.91. Next, in Figure 19, the AUC diagram of the decision tree algorithm is shown separately for each class.

According to the results obtained from the implementation of the algorithm shown in Figure 19. The decision tree algorithm has an AUC value equal to 1 for data from the FTP-CONTROL and FTP-DATA classes. The lowest value of the AUC criterion in the decision tree was also obtained for data from the P2P class with an AUC value equal to 0.93. Next, in Figure 20, the AUC diagram of the Naïve Bayes algorithm is shown separately for each class.

According to the evaluation results shown in Figure 20, the Naïve Bayes algorithm has an AUC criterion value below 0.5

for most data classes. And it is a very poor performance in the field of traffic classification and detection. The lowest AUC value is also for data from the SERVICES class with a value of 0.08. Next, Figure 21 shows the AUC diagram of the random forest algorithm separately for each class.

According to the evaluation results shown in Figure 21, the random forest algorithm has an AUC criterion value of 0.99 for most classes, and has obtained a performance close to the hybrid model. The lowest value of the AUC criterion in the random forest algorithm was also obtained for data from the P2P class with an AUC value equal to 0.90. Next, Figure 22 shows the AUC diagram of the AdaBoost algorithm separately for each class.

According to the evaluation results shown in Figure 22, the AdaBoost algorithm, like the Naïve Bayes algorithm, has an AUC criterion value close to 0.5 for most classes. The lowest value of the AUC criterion in the AdaBoost algorithm was also obtained for data from the FTP-CONTROL class with an AUC value equal to 0.51.

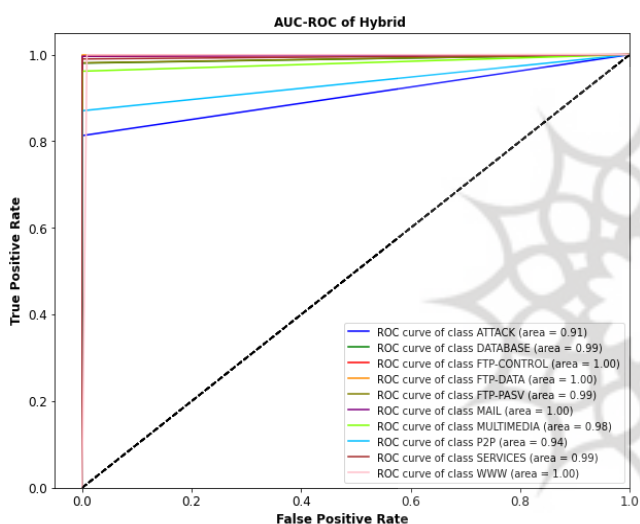


Figure. 18. Evaluation of CNN+LSTM model based on AUC criterion

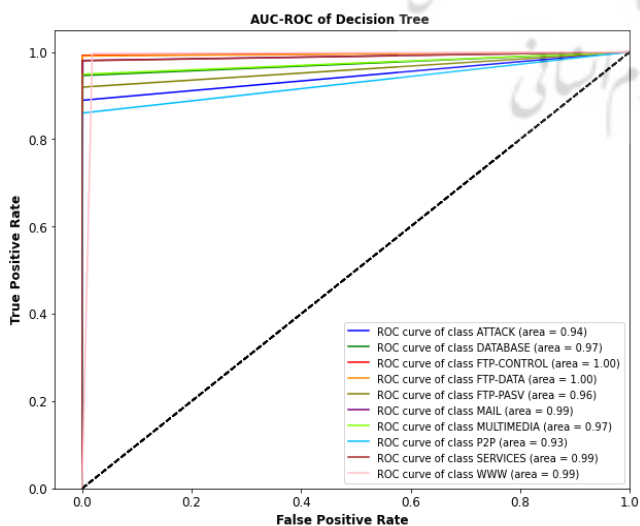


Figure. 19. Evaluation of the decision tree algorithm based on the AUC criterion

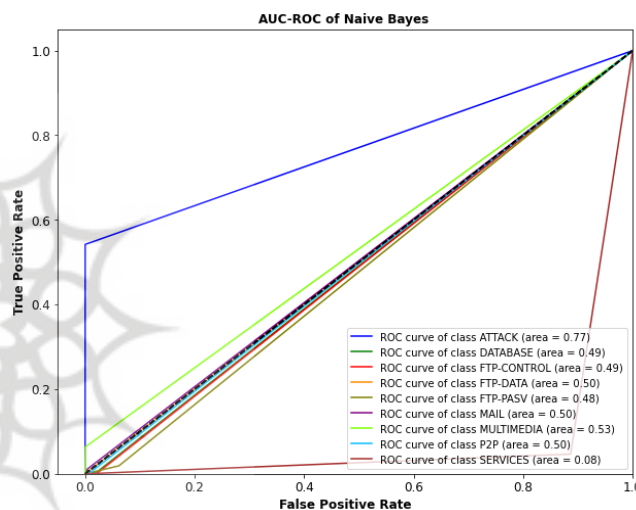


Figure. 20. Evaluation of Naïve Bayes algorithm based on AUC criterion

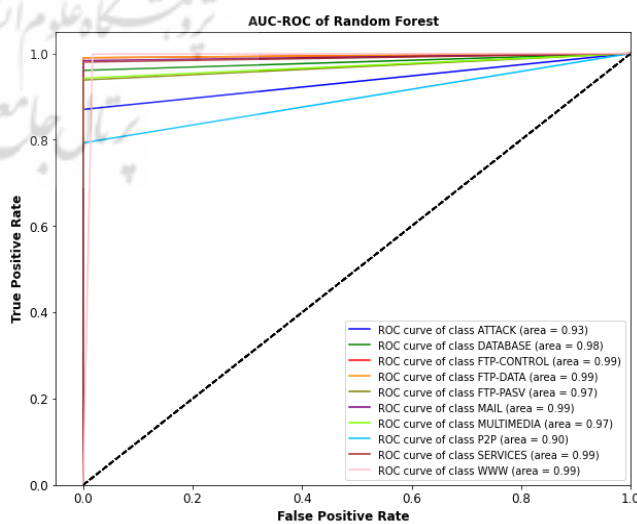


Figure. 21. Evaluation of random forest algorithm based on AUC criterion

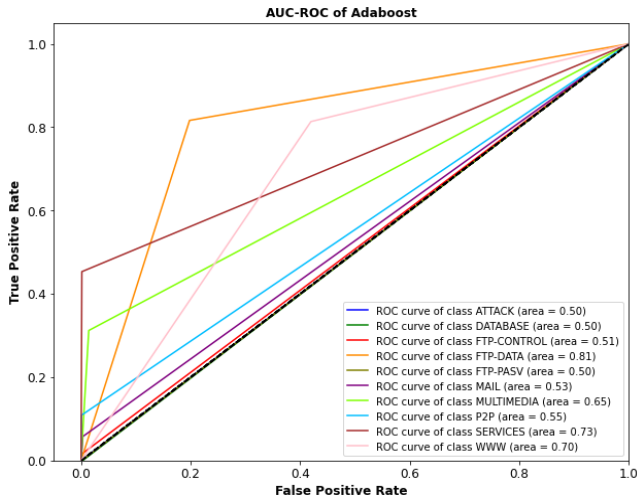


Figure. 22. Evaluation of AdaBoost algorithm based on AUC criterion

In the following, the result of the comparison of the proposed combined CNN+LSTM method with comparative algorithms including CNN and LSTM deep learning algorithms and machine learning algorithms including decision tree, Naïve Bayes, random forest, and AdaBoost are discussed based on accuracy, precision, recall, and F1 score criteria. Table 3 shows the results of implementation and execution.

As can be seen from Table 3, in the proposed two-step method after implementation on the Moore data set, the proposed CNN+LSTM model obtained values of 0.997, 0.972, and 0.959. 0.964 for accuracy, precision, recall, and F1 score, respectively and has the best performance among all compared algorithms. Among the comparative algorithms, the CNN algorithm performs better than other algorithms by obtaining values of 0.997, 0.964, 0.954, and 0.957 respectively for the criteria of accuracy, precision, recall, and F1 score. The weakest results belong to the AdaBoost algorithm with values of 0.722, 0.331, 0.258, and 0.198 respectively for accuracy, precision, recall, and F1 score.

5. Conclusion

In this research, a two-step method based on feature extraction with a deep autoencoder and classification with a hybrid method based on CNN and LSTM deep learning algorithms were presented to detect and reduce security anomalies in software-based networks using network traffic classification. The implementation results were analyzed and compared based on the criteria of accuracy, precision, recall, and F1 score and the results of the proposed method were compared with CNN and LSTM deep learning algorithms and decision trees, simple Bayes, random forest, and AdaBoost machine learning algorithms. In the proposed two-step method, the proposed CNN+LSTM model obtained values of 0.997, 0.972, and 0.959. 0.964 has the best performance among all algorithms for accuracy, precision, recall, and F1 score, respectively. Among the comparative algorithms, the CNN algorithm performs better than other algorithms by obtaining values of 0.997, 0.964, 0.954, and 0.957 respectively for the criteria of accuracy, precision, recall, and F1 score. The weakest results belong to the AdaBoost algorithm with values

Table 3. Results of evaluation and comparison of algorithms

Algorithm	Accuracy	Precision	Recall	F1 score
Decision tree	0.993	0.946	0.95	0.948
Naïve Bayes	0.952	0.778	0.581	0.611
Random forest	0.995	0.967	0.945	0.955
AdaBoost	0.722	0.331	0.258	0.198
CNN	0.997	0.964	0.954	0.957
LSTM	0.993	0.96	0.906	0.928
CNN+LSTM	0.997	0.972	0.959	0.964

of 0.722, 0.331, 0.258, and 0.198 respectively for accuracy, precision, recall, and F1 score. Therefore, it can be concluded that the proposed method is a practical and operational method with high accuracy, which can be applied in the real world and used in the detection of security anomalies in networks using traffic classification and network data.

Declarations

Funding

This research did not receive any grant from funding agencies in the public, commercial, or non-profit sectors.

Authors' contributions

All authors contributed equally to the generation and analysis of data, and the development of the manuscript.

Conflict of interest

The authors declare that there is no conflict of interest

References

- [1] J. Naughton, "The evolution of the Internet: from military experiment to General Purpose Technology," *Journal of Cyber Policy*, vol. 1, no. 1, pp. 5-28, 2016. <https://doi.org/10.1080/23738871.2016.1157619>.
- [2] U. Cisco, "Cisco annual internet report (2018–2023) white paper. 2020," *Acessado em*, vol. 10, no. 01, 2021.
- [3] N. Al Khater and R. E. Overill, "Network traffic classification techniques and challenges," in *2015 Tenth international conference on digital information management (ICDIM)*, IEEE, 2015, pp. 43-48. <https://doi.org/10.1109/ICDIM.2015.7381869>.
- [4] Y. Xue, D. Wang, and L. Zhang, "Traffic classification: Issues and challenges," in *2013 International Conference on Computing, Networking and Communications (ICNC)*, IEEE, 2013, pp. 545-549. <https://doi.org/10.1109/ICCNC.2013.6504144>.
- [5] J. Xie, F. R. Yu, T. Huang, R. Xie, J. Liu, C. Wang, and Y. Liu, "A survey of machine learning techniques applied to software defined networking (SDN): Research issues and challenges," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 393-430, 2018. <https://doi.org/10.1109/COMST.2018.2866942>.
- [6] A. Mestres et al., "Knowledge-defined networking," *ACM SIGCOMM Computer Communication Review*, vol. 47, no. 3, pp. 2-10, 2017. <https://doi.org/10.1145/3138808.3138810>.
- [7] A. Shirmarz and A. Ghaffari, "Performance issues and solutions in SDN-based data center: a survey," *The Journal of Supercomputing*, vol. 76, no. 10, pp. 7545-7593, 2020. <https://doi.org/10.1007/s11227-020-03180-7>.
- [8] K. Kalkan, L. Altay, G. Gür, and F. Alagöz, "JESS: Joint entropy-based DDoS defense scheme in SDN," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 10, pp. 2358-2372, 2018. <https://doi.org/10.1109/JSAC.2018.2869997>.
- [9] N. A. Lima and M. P. Fernandez, "Towards an efficient DDoS detection scheme for software-defined networks," *IEEE Latin America*

- Transactions*, vol. 16, no. 8, pp. 2296-2301, 2018. <https://doi.org/10.1109/TLA.2018.8528249>.
- [10] P. Kumar, M. Tripathi, A. Nehra, M. Conti, and C. Lal, "SAFETY: Early detection and mitigation of TCP SYN flood utilizing entropy in SDN," *IEEE Transactions on Network and Service Management*, vol. 15, no. 4, pp. 1545-1559, 2018. <https://doi.org/10.1109/TNSM.2018.2861741>.
- [11] H. Peng, Z. Sun, X. Zhao, S. Tan, and Z. Sun, "A detection method for anomaly flow in software defined network," *IEEE Access*, vol. 6, pp. 27809-27817, 2018. <https://doi.org/10.1109/ACCESS.2018.2839684>.
- [12] X.-D. Zang, J. Gong, and X.-Y. Hu, "An adaptive profile-based approach for detecting anomalous traffic in backbone," *IEEE Access*, vol. 7, pp. 56920-56934, 2019. <https://doi.org/10.1109/ACCESS.2019.2914303>.
- [13] Y. Xu, H. Sun, F. Xiang, and Z. Sun, "Efficient DDoS detection based on K-FKNN in software defined networks," *IEEE access*, vol. 7, pp. 160536-160545, 2019. <https://doi.org/10.1109/ACCESS.2019.2950945>.
- [14] R. Kokila, S. T. Selvi, and K. Govindarajan, "DDoS detection and analysis in SDN-based environment using support vector machine classifier," in *2014 sixth international conference on advanced computing (ICoAC)*, IEEE, 2014, pp. 205-210. <https://doi.org/10.1109/ICoAC.2014.7229711>.
- [15] T. Dang-Van and H. Truong-Thu, "A multi-criteria based software defined networking system Architecture for DDoS-attack mitigation," *REV Journal on Electronics and Communications*, vol. 6, no. 3-4, 2017. <http://dx.doi.org/10.21553/rev-jec.123>
- [16] P. Baldi, "Autoencoders, unsupervised learning, and deep architectures," in *Proceedings of ICML workshop on unsupervised and transfer learning, 2012: JMLR Workshop and Conference Proceedings*, vol. 27, pp. 37-49. <https://proceedings.mlr.press/v27/baldi12a.html>.
- [17] S. Misra, S. Thakur, M. Ghosh, and S. K. Saha, "An autoencoder based model for detecting fraudulent credit card transaction," *Procedia Computer Science*, vol. 167, pp. 254-262, 2020. <https://doi.org/10.1016/j.procs.2020.03.219>.
- [18] M. Zamini and G. Montazer, "Credit card fraud detection using autoencoder based clustering," in *2018 9th International Symposium on Telecommunications (IST)*, Tehran, Iran, IEEE, 2018, pp. 486-491. <https://doi.org/10.1109/ISTEL.2018.8661129>.
- [19] S. Saha, "A comprehensive guide to convolutional neural networks—the ELI5 way," *Towards data science*, vol. 15, 2018
- [20] S. Khan, H. Rahmani, S. A. A. Shah, and M. Bennamoun, *A guide to convolutional neural networks for computer vision, Synthesis Lectures on Computer Vision*, vol. 8, no. 1, pp. 1-207, 2018. <https://doi.org/10.1007/978-3-031-01821-3>
- [21] M. Samadzadeh and N. F. Ghohroud, "Evaluating Security Anomalies by Classifying Traffic Using Deep Learning," *2023 9th International Conference on Web Research (ICWR)*, Tehran, Iran, 2023, pp. 135-141, <https://doi.org/10.1109/ICWR57742.2023.10138963>.



Najmeh Farajipour Ghohroud is an assistant professor at the Computer Engineering Department of Iranian University. She received the Ph.D. degree in computer engineering from Sharif University of technology. Her research interests include hardware security and trust, SDN networks, Real-time system design and embedded system design.



Mohammadreza Samadzadeh received BSc degree in Computer Software Engineering from Islamic Azad University, He received his MSc degree in Information Security, He is now a Ph.D. candidate of Information Technology Engineering at the University of Tehran, His research interests include Machine Learning, Deep Learning, Information Security and recommender systems.