A New Intrusion Detection and Prevention System Using a Hybrid Deep Neural Network in Cloud Environment

Subalakshmi Mani, Bose Sundan, Anitha Thangasamy, and Logeswari Govindaraj

Abstract Cloud computing has become an innovative technology, with distributed on-demand services; it has an attractive target for potential cyber-attacks by intruders. Intrusion detection systems (IDS) and intrusion prevention systems (IPS) are the most commonly used mechanisms to detect and prevent large-scale network traffic and any type of online attacks. In this paper, a novel framework for a deep long short-term memory (LSTM)-based intrusion detection system is proposed to detect network traffic flow patterns as either malicious or normal in a cloud environment. The proposed IPS prevents malicious attacks received from IDS by increasing the detection rate of malicious attacks and reducing computational time. The experimental results for the overall performance of the intrusion detection and prevention system were evaluated with 99% accuracy, precision, recall, and F-score. The evaluation results prove that the proposed system is suitable for effective attack detection and prevention in resource-constrained operational in cloud computing environments.

Keywords Intrusion detection system (IDS) · Intrusion prevention system (IPS) attacks · Security · Deep long short-term memory model · Cloud computing

1 Introduction

An intrusion detection system (IDS) is the most commonly used method to detect any specified type of attack. Many network-based attacks may particularly attack cloud security at the level of the network layer. It includes IP spoofing, port scanning, a man in the middle of the attack spoofing, denial-of-service attacks (DOS), and the distributed denial-of-service (DDOS) attacks. Generally, an intrusion detection system (IDS) is a software package that is responsible for detecting threats across

S. Mani · B. Sundan · A. Thangasamy (✉) · L. Govindaraj
Department of Computer Science and Engineering, College of Engineering, Guindy, Anna University, Chennai, India
e-mail: ani.astt18@gmail.com

B. Sundan
e-mail: sbs@annauniv.edu

A. P. Pandian et al. (eds.), *Computer Networks, Big Data and IoT*, Lecture Notes on Data Engineering and Communications Technologies 117,
https://doi.org/10.1007/978-981-19-0898-9_73

the network or system, while an intrusion prevention system (IPS) is the software responsible for stopping all the events. Nowadays, systems run both as intrusion detection and prevention systems. Among that, it can define some of the open-source software such as Snort, Suricata, and Bro.

Due to the increase in a newer form of attacks and malware type of detection was introduced. It was primarily introduced to detect newer attacks that were signature-based. The only problem with this type of detection approach might be suffering from false positives. Now from the above concepts, it clearly understands how an intrusion detection and prevention system are work. It gives precise information ranging from installation of machines such as victim, attacker, and snort to deploying it and performing attacks to see the working of snort to prevent the attacks that are detected. Despite the fact that the number of cloud projects has expanded rapidly in recent years, assuring the availability and security of project data, services, and resources remains a critical and demanding research subject. Attacks can deplete the cloud's resources, take the majority of its bandwidth, and devastate an entire cloud project in a short amount of time. Cloud computing is the preferred choice of every organization since it provides flexible and pay-per-use-based services to its users. However, security and privacy are a major hurdle in its success because of its open and distributed architecture that is vulnerable to intruders. The most popular approach for detecting attacks is an intrusion detection system (IDS). It examines the type, positioning, detection time, detection technique, information sources, and threats that existing cloud-based intrusion detection systems can detect.

A snort is an open-source tool used for intrusion detection and prevention systems. This paper discusses all the elements of snort [1, 2] that must prevent malicious attacks using snort to configure it as a full-fledged IPS. It also gives an overview of the functioning of snort by performing several attacks such as cross-site scripting and SQL injection attack. On the whole, it presents a complete step-by-step process of deploying a complete package of snort along with the configuration of both the attacker and victim.

The following three types of modes in snort [3] are mentioned below:

1. Sniffer mode: Sniffer mode examines each packet from the network that collects and shows it on the snort console.
2. Packet logger mode: To write all of the logs to disk, packet logger mode is utilized. It will capture all the incoming packets and store them in the database.
3. Intrusion detection mode: This mode will monitor and analyze all network traffic according to user-defined rules, and it prevents attacks whenever it come.

The main contribution of the work is as follows: A novel deep LSTM detection system is proposed with the preprocessing techniques to make the dataset more concise. Based on the analysis, the feature is extracted using the combined algorithms of PCA and LDA for dimensionality reduction and to avoid over-fitting.

1. The proposed deep LSTM-based IDS uses a grid search algorithm to choose the best hyperparameter weight, along with seep LSTM to remain a memory for a long time to detect network traffic flows as either malicious or normal in

a cloud environment. The detected type of malicious attack is prevented using a proposed prevention system for increasing the detection rate and reducing computational time.

2. The overall performance of the integrated intrusion detection and prevention system is compared with the existing work, and it provides 99% accuracy, precision, recall, F-score, and malicious packets dropped at a high rate.

The other section of the paper is organized as follows. Section 2 describes the detection of any kind of attack with various deep learning techniques. Section 3 discusses the proposed deep LSTM intrusion detection and prevention system for the attacks. Section 4 is about the evaluation metrics, and the experimental results are discussed in Sect. 5.

2 Related Works

Millar et al. [4], in today's Internet, distributed denial-of-service (DDOS) attacks are one of the harmful threats that disrupt essential services. The most compromising challenge in DDOS detection is the volume of live traffic that is to be analyzed is coupled with the attack approaches. LUCID [4] presents a practical lightweight deep learning [3–6] detection system that uses CNN to sequentially classify live traffic flow as malicious or benign. Zengguang et al. [7] largely focus on edge computing and machine learning, and it gives the novelty of the weight update method, which is similar to the principle of distributed machine learning [8], and here federated learning has some more similarities over local data. Secondly, FL uses the process of model training. İt deals with the effective measure of data privacy protection. The traditional protection methods are k-anonymity and diversification to specifically reduce noise in the data. In this experiment, CNN-based intrusion detection models were analyzed [9]. The self-organizing map algorithms that successfully updated community rules and learning rates proposed by Aneetha et al. [10] to govern the original data size as the original weight vector, as well as the static structure and random assignment of simple SOM weight vectors. The new technology is evaluated using performance metrics such as detection rate and false alarm rate. Aqeel et al. [11] the most common DDOS attacks with multiple comprised systems in clouds that act like a victim. From the traffic pattern, real-time attack alerts are obtained. It also demonstrates the alternative path to the origin source. So, it can block the path source of the attack. Gupta, Sharma et al. [2], an intrusion prevention system (IPS) is an IDS with the capability of blocking intrusions. It will drop the malicious packets, blocking the specific IP address or it will reset the connection with the system. An IPS acts faster on the malicious one whereas an IDS only gets a copy of the network traffic. Patel A et al. [12] propose the new IDPS system with alarm management and this prevention technique to investigate attacks in the cloud environment [13].

3 Proposed Methodology

This paper presents deep LSTM intrusion detection and prevention system that can be deployed in cloud environments. Our proposed system identifies and prohibits harmful activity TCP and UDP patterns of malicious attacks in a cloud environment. To identify cloud attacks, a novel deep LSTM intrusion detection and prevention system is proposed for network traffic in cloud environments. After receiving the packets from the proposed deep LSTM detection system the condition uses protocol and length attributes or features is used to check whether the received packets are normal or not. This section describes the deep LSTM IDS framework Fig. 1. The network traffic preprocessing method, feature extraction techniques, proposed intrusion detection system and prevention system.

3.1 Dataset Description

The dataset consists of online traffic flows which are collected as a packet by a tool called Wireshark. Live traffic is collected using a virtual machine in which Ubuntu is a target address (carried by LOIC) and Windows 8.1 as a source address. Table 1 shows the classification of the train and test dataset. The dataset consists of the source port, destination port, protocol, time, length, etc., of 32 features and 1 feature

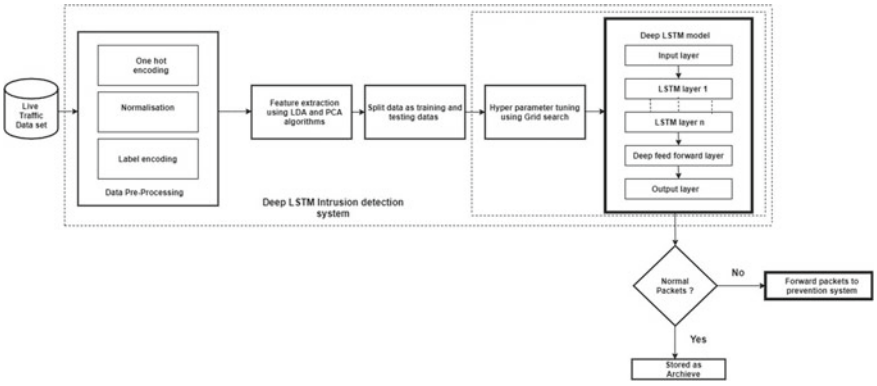

Fig. 1 Proposed deep LSTM intrusion detection and prevention system

Table 1 Train and test set classification of dataset

Dataset data type	Train set	Test set
Malicious	102,422	25,605
Normal	78,175	19,543
Total	180,597	45,148

as a label that shows the attack is either malicious or normal. The total parameters are above two lakh parameters. The features of the protocol contain 10 categories, whereas the label consists of two categories, i.e., malicious and normal, in which malicious counts up to 128,027, and normal counts to 97,718.

3.2 Data Preprocessing Techniques

In the data preprocessing step, the unbalanced data are converted to balanced data using normalization, label encoding, and one-hot encoding. Algorithm 1 provides all the steps that govern the data preprocessing for deep LSTM IDS.

3.2.1 Normalization

Feature scaling is an essential preprocessing step for our proposed framework. Normalization scales each feature separately to a fixed range. Usually, the range [0,1] is used. To scale these features to this range, the minimum and maximum of each feature in the dataset must be calculated.

3.2.2 Label Encoding

As the dataset, features of protocol and label are in an unbalanced state to convert it into machine-readable form. It mainly involves converting each value in a column to a number.

3.2.3 One-Hot Encoding

One-hot encoding is applied to categorical data to convert it into a binary vector representation for use in many deep learning algorithms.

Algorithm 1: Data preprocessing	
Input:	Unbalanced data
Output:	Balanced data
Step 1:	To convert the unbalanced data into a balanced one, it involves the following techniques: normalization, label encoding, and one-hot encoding
Step 2:	In normalization, it scales the feature values to the same specified range without distorting differences in the range of values. In normalization, selected standard scalar The value is normalized as follows, $Y = (x - -\text{mean})/(\text{standard\_deviation})$ Where x is the original feature vector

(continued)

(continued)

Algorithm 1: Data preprocessing	
Step 3:	Label encoding can be used to transform non-numeric to numeric labels
Step 4:	One-hot encoding changes the variable of binary representation as 0's and 1

3.3 Feature Extraction Methods

In dimensionality reduction algorithms, PCA and LDA are used to select the best features. As the dataset has too many features with too many attributes and to make our model more lightweight, the dataset is reduced with the combined algorithm of PCA and LDA. Combining attributes into a new reduced set of features. The feature extraction method involves.

1. Principal component analysis algorithm
2. Linear discriminant algorithm.

3.3.1 Linear Discriminant Algorithm

LDA is a supervised classification technique that provides more classes to the feature set while also reducing its dimensionality. And the dimensionality reduction property also makes our model more accurate. It aims to make the distance between data points of the same class more compact, which is shown in Algorithm 2.

Algorithm 2: Linear discriminant analysis algorithm	
Input:	Preprocessed data
Output:	Extracted Features
Step 1:	Compute the mean vectors for the input features dataset Mean : $X = \frac{1}{n} \sum X_i$ Where n is the number of values and X_i is the sum of the values of each input X
Step 2:	Calculate the scatter matrices, within the class (S_w) and between classes (S_b)
Step 3:	Find the linear discriminants by computing the eigenvalues $S_w - 1$, S_b
Step 4:	Select the eigenvectors W with the highest eigenvalues as the linear discriminants for the new feature set
Step 5:	The new feature set obtained from the linear discriminants is then used to obtain the transformed input dataset $Y = X.W$ where X is an n × d-dimensional matrix representing the n samples, and y is the transformed n × k-dimensional samples in the new subspace

3.3.2 Principal Component Analysis Algorithm

PCA uses a conversion method to convert the data into a lower-dimensional space while maximizing the data to analyze the covariance of each X and Y feature using the following Algorithm 3. It is a method for summarizing data. The resulting output features are the uncorrelated orthogonal basis set called principal components. The largest eigenvalues contain the most massive amounts of data.

Algorithm 3: Principal component analysis	
Input:	Preprocessed data
Output:	Extracted Features
Step 1:	Compute mean vectors for the input features dataset (x_i) Mean : $\overline{x}\frac{1}{n}\sum x_i$ Where n = number of values and x_i is the sum of values of each input x
Step 2:	Determine the scatter matrix – covariance matrix, two feature vectors x_j and x_k the covariance between them σ_{jk} can be calculated using the following equation: $\sigma_{jk} = \sum_{i=1}(x_j - \mu_j)(x_k - \mu_k)$
Step 3:	Compute eigenvectors and eigenvalues to compute the principal components
Step 4:	Eigenvectors to be sorted in descending order W
Step 5:	Integrate the principal components onto the input features

3.4 Splitting of Data

The random selection of 80% of the original data is split, and the testing set should be the remaining 20%. The parameters of the model of an optimization algorithm are adjusted during a training process. The test set must be separated from the training set. It is used to evaluate the accuracy of the trained model.

3.5 Hyperparameter Tuning Using Grid Search

The objective is to minimize the complexity of the time taken and the performance time of the deep LSTM model. To achieve this, a lightweight model is proposed. It has shared and reused parameters about the weight. This reduces the storage and memory requirements of our model. Hyperparameters are the setting parameters of the model that are not changed during the training process. The training parameters consist of batch size, number of epochs, learning rate, momentum, and dropout. To develop good LSTM models, the appropriate hyperparameters should be found. It is possible to determine the combinations of suitable hyperparameters. The grid search technique is used to find the best hyperparameters. Grid search will result in the most

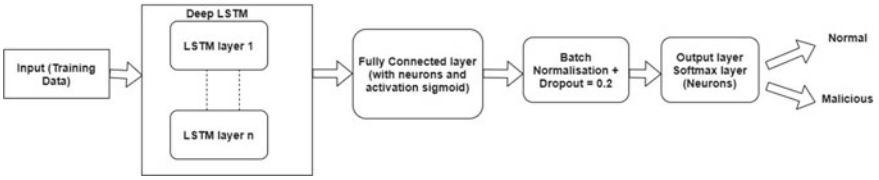

Fig. 2 Deep LSTM model for attack detection

'accurate' predictions. In automated research techniques, algorithms are used to find the best values. Steps involved in hyperparameter tuning using grid search algorithm are shown as Algorithm 4.

Algorithm 4: Hyperparameter tuning using grid search	
Input:	Extracted features from dimensionality reduction algorithms
Output:	Fit the model
Step 1:	Briefly give the parameters that you use to train Grid search along with the values that you wish to try (i.e., epochs = [10, 50, 100] Batch_size = [10, 50, 20] Dropout = [0.2, 0.3, 0.7] Learning rate = [0.001, 0.0002, 0.00001])
Step 2:	
Step 3:	Fit the grid search

3.6 Deep LSTM IDS Model for Attack Detection

Deep LSTM model Fig. 2 is an advanced updation to the LSTM model that has one or more hidden layers. Each layer contains multiple memory cells that will hold information for a long duration. More hidden layers can be added to the multilayer perceptron neural network to make it look deeper. It shows the hyperparameter tuning with deep LSTM model detection is shown as algorithm 5.

Algorithm 5: Deep LSTM for attack detection
Input: Extracted features as input
Output: Classify the type of attack
Step 1: 80% of the training data is taken as input to the deep LSTM model
Step 2: As it is deep LSTM, this model contains more hidden layers and each layer contains memory cells
Step 3: The next layer is the fully connected layer with the activation function as sigmoid
Step 4: Batch normalization is done along with the dropout rate

(continued)

(continued)

Algorithm 5: Deep LSTM for attack detection
Step 5: The final layer is the output layer, which is the softmax layer that classifies malicious or benign
Step 6: Train the chosen deep LSTM model on the training set (80%)
Step 7: Validate the chosen deep LSTM model on the evaluation set (20%)
Step 8: Test the chosen deep LSTM model on the test set
Step 9: Repeat steps until the desired results are reached

3.7 Proposed Intrusion Prevention System

The detection of malicious attacks employing deep learning algorithms is done in a deep LSTM intrusion detection system. To prevent detected malicious attacks, it employs proposed snort techniques to eradicate malicious attacks. Complete malicious attacks which are taken from the detection using the deep LSTM intrusion detection system are given to the input of the snort. Whenever the snort is activated, the port of TCP, UDP of attacks is identified concerning rules generated, using snort rule engine will identify and prevent the attacks.

The detected malicious attack is given as the input of the snort, which is manually generated or given as a dataset. The snort rules are generated. The snort with the given rules whenever the type of attack is seen, the snort will send the alarm of the attacks to the cloud administrator, and it will prevent accordingly. Figure 3 shows the proposed prevention system for attack prevention and the steps discussed below:

Packet Decoder: It is used to capture and analyze network traffic generated during packet capture. As packets are already captured and classified, the type of attack is classified from the detection phase.

Preprocessors: Packets are captured from the ports of TCP and UDP. İt can handle the data over multiple packets. Here, snort uses preprocessors for pattern matching.

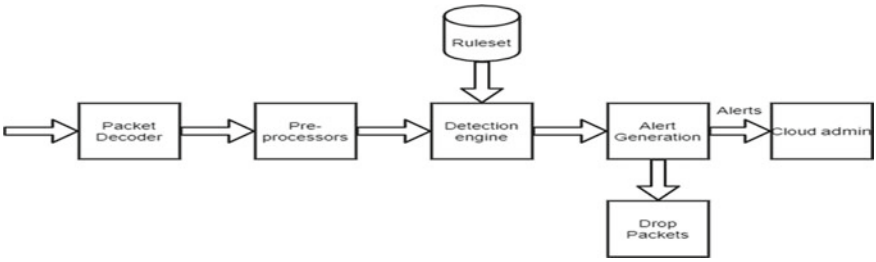

Fig. 3 Proposed intrusion prevention system

Detection Engine: It will prevent any malicious system attacks. From the rule set, detection engine will eliminate it by blocking the attack.

Ruleset: The rule sets are created according to the ports of TCP and UDP. The type of attack is predicted from the detection system.

Alert Generation: Whenever the type of attack is detected, based on the ruleset alerts are generated. From the detection engine, alerts are generated to the cloud administrator for dropping the packets.

Drop packets: Followed by the alert generation from the cloud admin, the detection engine will detect and display the packets that are dropped finally eliminate them.

Cloud admin: From the generated alert, the corresponding port attacks are displayed on the cloud admin.

4 Evaluation Metrics

In our model, the most important performance is accuracy, precision, recall, and f-score of intrusion detection is used to calculate the performance of the proposed deep LSTM model. Also, we discussed TP, TN, FP, FN as the predicted equals the actual, if the actual is positive and the model predicted a positive value then it is TP, if the actual is negative and the model predicted a negative value then it is TN, if the actual is negative and the model predicts a positive, then it is FP if the actual value is positive and the model predicted is negative then it is FN.

Accuracy: Accuracy represents the total percentage of correctly classified samples of both normal and malicious as shown in Eq. (1).

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{1}$$

Precision: PPV represents the ratio between the correctly detected malicious, and all detected malicious samples are shown in Eq. (2).

$$\text{PPV} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

Recall: Recall represents the percentage of malicious samples that are correctly classified as shown in Eq. (3).

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

F1-score: It represents the percentage of samples that are falsely classified as malicious as shown in Eq. (4).

Table 2 Confusion matrix for the two category experiments on the testing set

Predicted class actual class	Malicious	Normal
Malicious	19,394	11
Normal	25,744	0

$$F1 - \text{score} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{4}$$

Confusion Matrix: The confusion matrix is a kind of error matrix. It visualizes the prediction for a classification task. Table 2 shows the confusion matrix for the two category experiments on the testing set with predicted class and actual class of testing data.

Hence, the motivation was to obtain high accuracy, precision, recall, f-score for the proposed deep LSTM IDS model. From the detected phase, it focuses on dropping packets using snort at a high rate.

5 Experimental Results

The experiments on lightweight deep neural networks were performed by TensorFlow and Keras and are used to implement the proposed deep LSTM intrusion detection and prevention system in the cloud environment, which is provided by IBM Watson studio. This experiment was conducted to detect TCP, UDP attacks. The dataset is collected from a tool called Wireshark, which consists of 32 features and parameters that are more than 2 lakhs, and is split into 80% of training data and 20% of test data. In the proposed deep LSTM model, which consists of more than 3 layers, 100 + neurons are used. The classification of this model involves the logistic activation function of binary classification. The detection of TCP, UDP attacks with deep learning algorithms is done and sent to a prevention system to prevent malicious packets using proposed prevention techniques to drop malicious packets with high accuracy. The accuracy of the training model with 2 epochs is shown in Figs. 4 and

Fig. 4 Accuracy of training model of 23 epochs

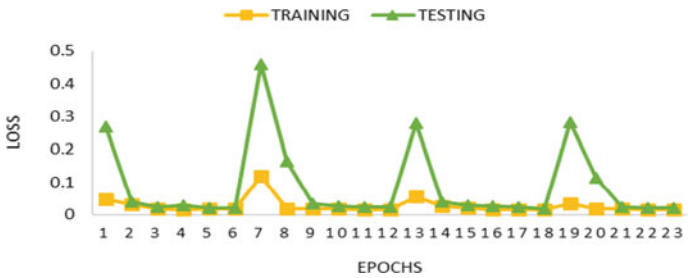

Fig. 5 Loss of training model of 23 epochs

5 that shows the loss of the training model. Here we compare the existing model [4] with the state-of-the-art comparison of proposed deep LSTM IDS model attack detection. The comparisons were held with other models on metrics like 'accuracy,' 'precision,' 'recall,' and 'F1 score' as shown in Table 3, and the comparison chart is shown in Fig. 6. The local snort rules that are generated for TCP UDP attacks are shown in Fig. 7 using the rule engine for the prevention system to prevent attacks received from the detection system. Figure 8 shows the comparison chart of the proposed prevention system and compared with existing methods [3]. It shows the dropping of packets with a high accuracy rate.

Table 3 Comparison results of the proposed deep LSTM IDS model

Model	Accuracy	Precision	Recall	F1-score
Proposed deep LSTM IDS model	0.9955	0.9926	0.9959	0.9897
Lucid [4]	0.9888	0.9827	0.9952	0.9889
Deep defense 3LSTM [5]	0.9841	0.9834	0.9847	0.9840
Mlp [14]	0.8634	0.8847	0.8625	0.8735
TR-IDS [15]	0.9809	0.0040	0.9593	0.8742
DeepGFL [15]	0.9624	0.7567	0.3024	0.4321

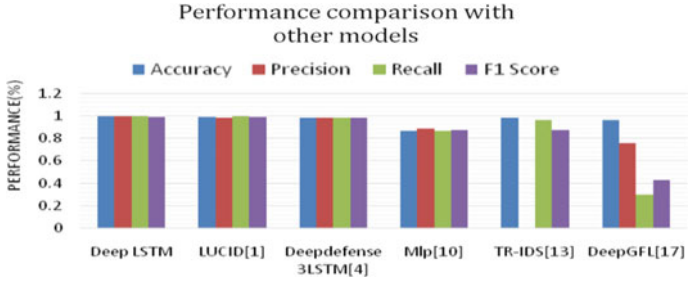

Fig. 6 Performance comparison of the proposed deep LSTM IDS model

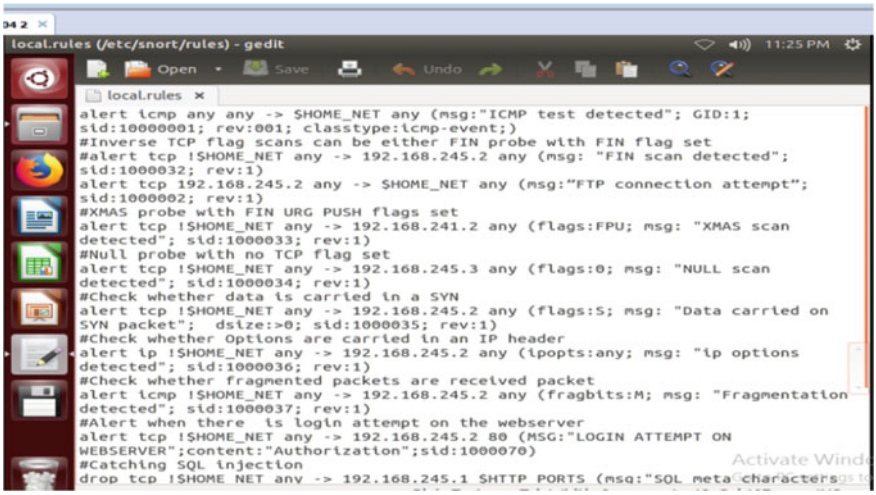

Fig. 7 Snort local rules for malicious attacks

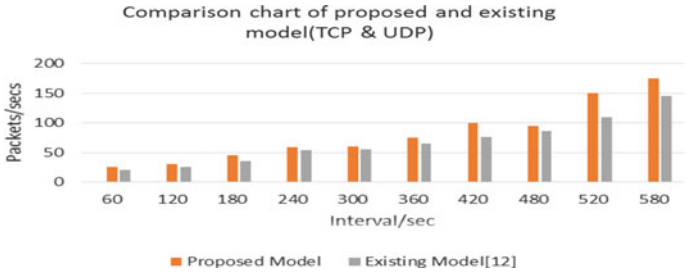

Fig. 8 Comparison chart of proposed prevention system [3]

6 Conclusion and Future Enhancement

Cloud computing in many sectors is becoming vast, as it uses to improve security in many aspects. In this paper, the incoming packets are classified to detect the behavior of the source whether the attack is malicious or normal. The results show that the novel deep LSTM-based intrusion detection system can accurately detect attacks. The detected attacks are passed to the proposed prevention mechanism using snort. The malicious packets are dropped with high accuracy. The overall proposed detection and prevention system have an accuracy of about 99% on the last iteration of best-chosen hyperparameter values under different attacks. The performance is validated and tested and is shown to be accurate. Thus, the proposed approach can efficiently improve accuracy using a security of data and will reduce the bandwidth usage and cut the over usage of resources. In the future, various types of DDoS attacks detection and prevention frameworks can be a study point along with deep learning algorithms.

References

1. Gupta, A., Sharma, L.S.: Mitigation of DoS and port scan attacks using snort. Int. J. Comput. Sci. Eng. **7**, 248–258 (2019)
2. Patel, A., Taghavi, M., Bakhtiyari, K., Junior, J.C.: Review: an intrusion detection and prevention system in cloud computing: a systematic review. J. Netw. Comput. Appl. **36**, 25–41 (2013)
3. Min, E., Long, J., Liu, Q., Cui, J., Chen, W.: TR-IDS: anomaly-based intrusion detection through text-convolutional neural network random forest. Security and Communication Networks (2018)
4. Doriguzzi Corin, R., Milla, S., Scott Hayward, S., Martnez Del Rincon, J., Siracusa ICT, D., Fondazione Bruno Kessler.: LUCID: a practical, lightweight deep learning solution for DDoS attack detection. IEEE Trans. Netw. Serv. Manage., **17**, 876–888 (2020)
5. Yuan, X., Li, C., Li, X.: Large-scale intelligent systems laboratory, deep defense: identifying DDoS attack via deep learning. Smart Comp. **1**, 1–8 (2017)
6. Saxena, R., Dey, S.: DDoS attack prevention using collaborative approach for cloud computing. Cluster Comput. **23**, 1329–1344 (2020)
7. Wu, K., Chen, Z., Li, W.: A novel intrusion detection model for a massive network using convolutional neural networks. School Control Comput. Eng. **6**, 50850–50857 (2018)
8. Kasongo, S.M., Sun, Y.: A deep learning method with filter based feature engineering for wireless intrusion detection system. IEEE Access **7**, 38597–38607 (2019)
9. Sahi, A., Lai, D., Li, Y., Dikyh, M.: An efficient DDoS TCP flood attack detection and prevention system in a cloud environment. Inf. Comput. Sci. **5**, 6036–6048 (2017)
10. Roopak, M., Yun Tian G., Chambers, J.: Deep learning models for cyber security in IoT networks. In: IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC) (2019)
11. Gupta, B., Badve, O.P.: Taxonomy of DoS and DDoS attacks and desirable defense mechanism in a cloud computing environment. Neural Comput. Appl. **28**, 3655–3682 (2017)
12. Yao, Y., Su, L ., Lu, Z.: Deep GFL: deep feature learning via graph for attack detection on flow-based network traffic. In: Proc. of IEEE Military Communications Conference (MILCOM) (2018)
13. Aneetha, A.S., Bose, S.: The combined approach for anomaly detection using neural networks and clustering techniques. Comput. Sci. Eng. **2**, 37–46 (2012)
14. Subba, B., Biswas, S., Karmakar.: A neural network based system for intrusion detection and attack classification. In: Twenty Second National Conference on Communication (NCC) (2017)
15. Shah, S. A. R., Isaac, B.: Performance comparison of Intrusion detection systems and application of machine learning to snort system. Future Gener. Comput. Syst., Elsevier, **80**, 157–170 (2018)