Computer Systems Science & Engineering
DOI: 10.32604/csse.2023.032078
Article

A Novel Framework for DDoS Attacks Detection Using Hybrid LSTM Techniques

Anitha Thangasamy*, Bose Sundan and Logeswari Govindaraj

Department of Computer Science and Engineering, College of Engineering Guindy, Anna University, Chennai-600025, Tamilndau, India
*Corresponding Author: Anitha Thangasamy. Email: anithathanagasamy123@gmail.com
Received: 06 May 2022; Accepted: 10 June 2022

Abstract: The recent development of cloud computing offers various services on demand for organization and individual users, such as storage, shared computing space, networking, etc. Although Cloud Computing provides various advantages for users, it remains vulnerable to many types of attacks that attract cyber criminals. Distributed Denial of Service (DDoS) is the most common type of attack on cloud computing. Consequently, Cloud computing professionals and security experts have focused on the growth of preventive processes towards DDoS attacks. Since DDoS attacks have become increasingly widespread, it becomes difficult for some DDoS attack methods based on individual network flow features to distinguish various types of DDoS attacks. Further, the monitoring pattern of traffic changes and accurate detection of DDoS attacks are most important and urgent. In this research work, DDoS attack detection methods based on deep belief network feature extraction and Hybrid Long Short-Term Memory (LSTM) model have been proposed with NSL-KDD dataset. In Hybrid LSTM method, the Particle Swarm Optimization (PSO) technique, which is combined to optimize the weights of the LSTM neural network, reduces the prediction error. This deep belief network method is used to extract the features of IP packets, and it identifies DDoS attacks based on PSO-LSTM model. Moreover, it accurately predicts normal network traffic and detects anomalies resulting from DDoS attacks. The proposed PSO-LSTM architecture outperforms the classification techniques including standard Support Vector Machine (SVM) and LSTM in terms of attack detection performance along with the results of the measurement of accuracy, recall, f-measure, precision.

Keywords: Cloud computing; distributed denial of service; particle swarm optimization; long short-term memory; attack detection

1 Introduction

Cloud Computing has been recently used by most of the researchers, due to its widespread benefits and applications. Cloud Computing entirely relies on internet for providing services and its distributed nature possesses several challenges regarding security. The most serious challenge is Distributed Denial of Service (DDoS) attack which totally deactivates all the services. A Distributed Denial of Service attack

may cause immense harm to real users of resources and the available resources. Because of its comparatively low expertise and wide storage, the provided defending system could not be easily used for cloud computing. In 2015, the number of DDoS attacks globally increased by 25% and that raised by 2020 to 17 million by 2.6 times. Multiple rich cloud computing characteristics have increased the adoption of cloud computing system by several organizations. The platform of cloud computing expands the advantages to both cloud customers and cloud service providers.

Cloud computing makes access without expensive computer hardware to various resources easily with cost efficient. Furthermore, the cloud's on-demand feature has been helpful in reducing cloud customers' operating costs, once the demand for cloud-based services becomes highly speculative [1]. The simplicity of the cloud pricing model enables the customers to pay for their estimates. Many cloud features assist service providers to reduce operating costs and also obtain better performance. The reason behind DDoS attacks is to interrupt a server's normal operations [2,3]. A successful DDoS attack has been launched by three entities: Attacker, Handler, and Bot.

The attacker is the party that permits the attack, even though the handlers and Bots damage the computing machines. The installation of a malicious programmer on such machines compromises such computers. Next, the attacker enlists some attack handlers to enlist the handlers and then, the attacker scans the web for carrying out DDoS attacks. It is displayed in Fig. 1.

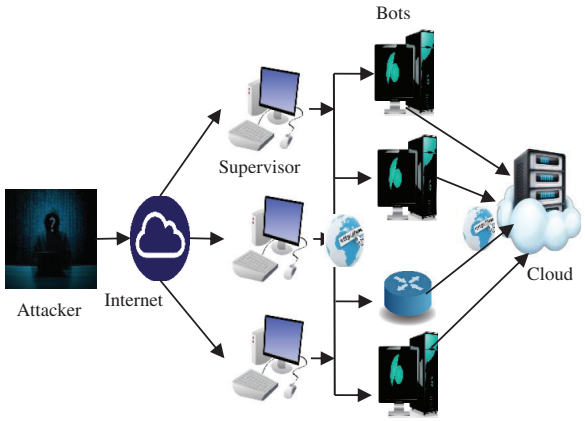

Figure 1: The general mechanism to perform DDoS attacks in Cloud

Potentially vulnerable computers install a handler program on those computers. A Bot Program is installed through these computers to build a botnet instead after supervisors begin to scan for more vulnerable computers. The attacker sends the appropriate instructions for the attack to the supervisors to start the attack. Supervisors then send instructions through the target cloud server to the Bots to initiate the DoS attack. This prevents DDoS attacks to detect various kinds of DDoS attacks using different network flow features, while DDoS attacking strategies are delayed, due to the complexity of the algorithm.

Types of available DDoS attacks are:

a. Volumetric-dependent attacks

It acquires targeted server's bandwidth for every second to lower down the services. Eg: TCP, UDP and ICMP flooding attacks.

b. Protocol-dependent attacks

It consumes the targeted server's resources for every second to pull down the services. Eg: ping of death, SYN flood and Smurfing attack.

c. Application Layer attacks

This attack makes the application to be not available for the real users. It crashes targeted server. Eg: attack by Hash DoS and attack by Teardrop.

Currently, numerous studies have been carried on DDoS attacks. For designing the architecture to predict DDoS attacks, it needs to know its advantage and disadvantage. From the studies, it is clear that DDoS attack has closest structure to other kinds of attacks and it results in wrong classification of attacks. Thus, selecting appropriate classification function is very critical task. To overcome the above said issues, in the present paper, it has been planned use deep learning technique like PSO which is integrated with LSTM to predict DDoS attacks in cloud environment.

The main contributions of the work are as follows:

a) To make dataset concise to be preferred for further activity and pre-processing has to be carried out.
b) Depending on the analysis of characteristic of DDoS attack, feature of DDoS is extracted using Deep Belief Network (DBN) method
c) To choose the best optimal weight for Neural Network (NN), PSO algorithm is used to train the model
d) To retain memory for long duration of time, LSTM has been chosen for effective classification of attacks.
e) To classify and predict DDoS attack, PSO-LSTM method has been proposed and used.

The remaining sections of the paper have been organized as follows: Section II describes the prediction of DDoS attack with various methods like PSO, LSTM etc. In Section III, the proposed architecture is discussed briefly for predicting DDoS attacks with PSO-LSTM method. In Section IV, the experimental results and the comparison analysis are discussed briefly and finally, the conclusion of the proposed work is provided in Section V.

2 Related Work

This section presents a detailed analysis of most recent intrusion detection and preventative systems designed to reduce potential DDoS attacks. At the beginning of null networks, Aneetha et al. [4] proposed updated self-organisation map algorithms that formed the original data size as the original weight vector and successful updating of neighbourhood regulations as well as learning rates to regulate the fixed architecture and random assignment of simple SOM weight vectors. The new procedure evaluated through performance measures such as detection rate and false alarm rate, resulting in an impressive rise in the detection rate and a 2% false alarm rate in comparison with other individual methods in the neural network. Bahman et al. [5] analysed network virtualization-based DDos defense mechanism. In DDoS attack system, excessive network traffic was diverted to various other collaborative domains for filtration. It enabled domains network to assist other domains to handle large volume of DDoS attack traffic through sharing resources.

Kim et al. [6] introduced a long shorter-term memory and persistent neural network focused on machine learning in order to detect DDoS flooding attacks, and the second generation Tensor flow architecture was used by Google. Productivity and precision were determined by seven parameters, the accuracy of detection was 99.968 percent. Here, both the CPU and the GPU were used.

CoFence a collaborative network presented by Bi et al. [7] suggested User action anomaly detection method for Discrete-Time Markov Chains (DTMC). Normal user features were obtained based on the user behaviour. This model compared the normal user and the detected user behaviour. In this model, a

default value was set when identified user features surpassed the threshold value and it was assessed as an intruder for DDoS.

Braga et al. [8] implemented feature extractor module which stored steams disentangle features that were crucial to detect DDoS flooding attack, as well as accrued them in 6 tuples to elapse the classification. SOM is a system of classification and it is informed by traffic flow features. The module scans 6-tuples and they are compared with approved traffic or DDoS flood attack.

Chen et al. [9] introduced TSDNN (End-End Trainable Tree Shaped Deep Neural Network) which was used to categorize data. For the case of minority classes, the author proposed QDBP (Quantity Dependent Backpropagation) which used the concept of disparity among classes. An experiment was conducted to evaluate the proposed work on imbalanced datasets and the results showed that proposed work outperformed the existing system.

Wang et al. [10] proposed Sky Shield program that managed large DDoS flood attacks. Bloom filters, and CHAPCHA made it more powerful. Sky Shield would request rate for floods to a reasonable level and Sky Shield effectively would detect attacks that happened in a flash crowd case. The author stated that the attack by AL-DDoS rapidly alleviated legitimate users with a restricted power. Joseph et al. [11] implemented a scalable method of cloud-based detection against DDoS attack. They suggested reserving backup resource to exercise and extended to a federated virtual machine cloud design. The architecture was legitimized by counter measures from DDoS being guided by the attack stratagem.

Luo et al. [12] suggested D-PID, a mechanism that also adaptively changes path identifiers (PID) and PID will be used as inter domain routing object. D-PID protects huge-scale networks from attacking DDoS. In this research, they have concluded that DDoS attack is effectively blocked using D-PID, and it increases the cost of DDoS attack launch.

Ma et al. [13] developed a new approach which allowed detail analyzes for the identification of network anomalies. Two ISCX-IDS-2012 and CIC-IDS-2017 datasets were implemented to test the performance of the proposed methodology. The present study shows that for other algorithms, the comprehensive output of the proposed method is better.

Sarra et al. [14] programmed a resource-based DDoS mitigation solution which was called as a multi-level collaborative DDoS mitigation for the service cloud. This device tracked resource consumption, Virtual Machine and hypervisor level, as well as managed the detection of attacks.

Shui et al. [15] suggested a technique known as quick DDoS mitigation focused on the scaling of Cloud resources. The network monitoring server filtered the attack packet for the cloud during most of the attack period, depending on idle resource allocation, and hence, this approach assured high quality services. Yang et al. [16] implemented a method namely SBTA and it was placed in front of web Brower depending on the use of SOA. It was used for tracking back and finding the DDoS attack source address. Detection of DDoS was successfully done through the combination of SBTA and Cloud-Filter. Subsequently, choosing an appropriate classification feature is very important. To keep the cloud available to users all the time, threats that affect the cloud availability need to be detected and prevented. This work offers effective swarm-based as well as deep learning methods for DDoS attack for reorganization and prevention.

3 Proposed Methodology

The NSL-KDD dataset, which has been used to experiment and to classify the phase, would be a basic data set for evaluating the benefits of the conceptual approaches in the development of network intrusion detection. Fig. 2 illustrates the proposed DDoS attack detection based on PSO-LSTM Learning method. In the first step, the network set of data has been designed and traffic functions (e.g., packet rate, protocol type, etc.) will be collected in the next step when using DBN. Such features can be computerised to ease

up a training phase. The training data would be fitted out with PSO-LSTM learning algorithms. The above packet could be labelled as a relevant DDoS attack in real network traffic. The NSL-KDD dataset has been categorized into five classes by such a method as Normal, Remote to local (R2L), Probing, Denial of service (DoS) and User to Root (U2R) as part of features and also the SVM for classification. Based on the suggested definition, the set of data will decrease from 41 to 5 characteristics using the DBN and then, add SVM is added to these decreased data as well as categorization would be achieved.

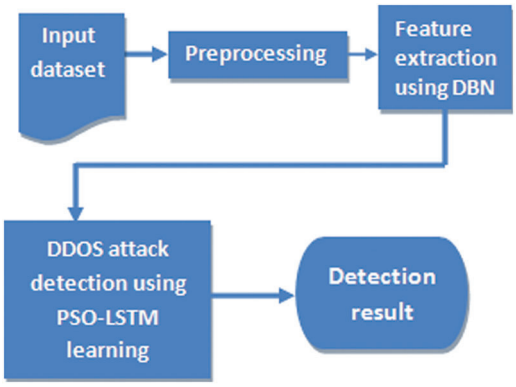

Figure 2: Architecture of DDoS attack detection based on PSO-LSTM learning

3.1 Pre-processing the Dataset

String values have to be substituted with integer values until the dataset could be used and the replicated records will have to be deleted. String value field would have a set of integer variables equal to the number of string values; for instance, the "protocol form" field will contain integer values of 1, 2 or 3. Although the database includes about half a million documents, it is a very time-consuming and tedious method to compare and record all the records and delete the duplicated ones. To decrease the complexity and accelerate the process, the dataset is divided into subsets with no common records. Hence, the data are divided into records of 'attack type,' 'protocol type,' 'flag' and 'operation' logged in fields. The "data registered" are binary data with a field number of 12.

After the data are segregated, there are large numbers of records in certain subsets which increase the probability that the duplicated records are present in the subset. It calculates the total integer of the fields in all records to solve this problem. If the summation of the integer values of two record fields is unique, then the two records are classified differently. Otherwise, the two documents that have similar summation values are tested more carefully to decide whether they are duplicates or not. The total number of records decreases to 145,586 when all the duplicated records are eliminated.

3.2 DDoS Feature Extraction Using Deep Belief Network

Single DDoS attack data packet has been typically valid in protocol and content with the advancement of attack technologies. Various DDoS attackers can be downloaded directly from the Internet to potentially compromise data protection for any internet user. DDoS attacks are becoming easier to use, more frequent, alarming, more complicated, more dynamic, more robust and more difficult to follow. Each NSL-KDD record set contains 41 features (e.g., protocol type, service, flag) to mark them as standard, different types of attack.

The attack types are Denial of service, Root to Local Probing and User to Root. Neptune, Smurf, Pod, and Teardrop are the examples for Denial of Service attacks (DoS). Ftp-write, imap and Phf Guess-Password

are the examples of a remote computer to a local (R2L). Rootkit, Load-module, Buffer overflow and Perl are the examples of User to Root (U2R). Satan, Port-sweep, ip-sweep and nmap are the examples of probing.

Depending on analyzing the features of DDoS attack, the above work recommends a method for extracting features from DDoS premised on a deep-belief network. Deep belief network is a learning model method implemented by Hinton [17], with a deep structure. It may be one of the first non-convolutionary patterns and it successfully applies profound training in architecture.

In comparison with the Conventional neural network, it has better modelling and representation capacity, until it can total a complex approximation. A collection of Boltzmann platform modules, stacked in a restricted fashion, can be seen in the deep belief network. The front layer of the cached layer and the data of the second cached layer are next transparent layer. Then, the Restricted Boltzmann (RBM) is a layer of measurable and latent variables. This is a model focused on energy and a recurrent neural network. Fig. 3 demonstrates the restricted structure of the Boltzmann system. It has been demonstrated to be a two-part graph. Just two neurons are located in RBM. The name of one layer is the hidden layer used to enter a training data and there is an interconnection between the layers but no link occurs in the layer between the units. The visible layer may be used to access training information and the hidden layer is employed to capture greater association of data within the visible layer. In the restricted Boltzmann system, neurons tend to be binary and only two states continue to be used: 0 and 1. State 1 is activation and State 0 is suppression.

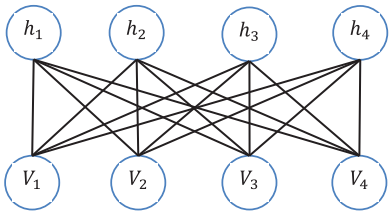

Figure 3: RBM composition diagram

The deep belief network can be seen as a set of modules and it is enclosed and stacked in Fig. 4 of Boltzmann. The visible units are the first level, which corresponds with the analysis components, i.e., one visible unit for each input sequence characteristic. Therefore, the dependence of hidden unit model between two elements of the findings is the dependence of the characteristics.

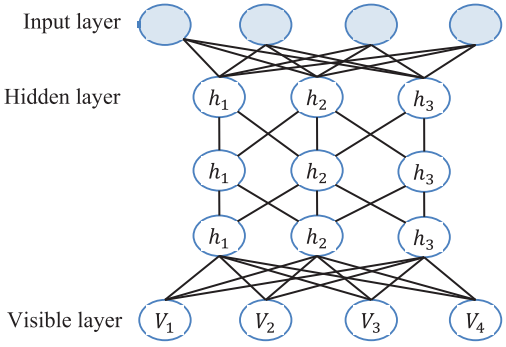

Figure 4: DBN composition with RBM structure

The energy dependence model ensures that all variables v and h are normally distributed through this entropy function. This function is defined by Eq. (1):

$$E(m,\ n;\ \theta) = E(v,\ h;\ \theta) = -\sum_{i=1}^{m}\sum_{j=1}^{n} w_{ij}v_i h_j - \sum_{i=1}^{m} b_i v_i - \sum_{j=1}^{n} a_j h_j \tag{1}$$

Here, the visible layer is $v_1,\ldots, v_m$ where m denotes the number of neurons in the visible layer and the hidden layer is $h_1,\ldots, h_n$ where n denotes the number of neurons in the hidden layer where the two bias vectors are a and b, θ = w, a, b and the weight of the matrix is mentioned in the above Eq. (1).

In order to use Entropy, each pair of the network neurons can be allocated with a probability, one in the visible layer, and one in the hidden layer as mentioned in Eq. (2)

$$\mathrm{P}(v,\ h;\ \theta) = \frac{e^{-E(v,h;\theta)}}{\sum_{v,h} e^{-E(v,h;\theta)}} \tag{2}$$

3.3 DDoS Attack Detection Using PSO-LSTM Learning Method

Long-term-short-term memory [18] has become a special RNN model meant to solve the RNN model problems, including the internal RNN network status that reveals the dynamic subsequent behaviour. RNN uses its internal memory to manage arbitrary time series input pieces in contrast to neural networks to enable the processing of such a model. There are also two problems in RNN and they are vanishing gradient and explosive gradient. The original purpose of LSTM design is to resolve long-term RNN dependence, because retrieving long-term information is not a great deal of learning rather the standard behaviour of the neural network. The LSTM model substitutes RNN layer cells with LSTM cells for long-term memory functionality. The popular LSTM model structure uses the forgotten gate as well as the input gate for introducing parameters.

LSTM: In contrast to traditional Machine Learning (ML), Long Short-Term Memory (LSTM) networks could be used to identify attack pattern that repeats inside a long packet sequence regardless of window size. The forward calculation method [19] can be demonstrated in the LSTM neural network model as described in Eq. (3):

$$\begin{aligned}
\mathrm{f_t} &= \sigma(\mathrm{W_f}\ \ [\mathrm{h_{t-1}},\ \mathrm{x_t}] + \mathrm{b_f})\\
\mathrm{i_t} &= \sigma(\mathrm{W_{\mathbb{i}}}[\mathrm{h_{t-1}},\ \mathrm{x_t}] + \mathrm{b_i})\\
\tilde{\mathrm{C}}_\mathrm{t} &= \tanh(\mathrm{W_c}[\mathrm{h_{t-1}},\ \mathrm{x_t}] + \mathrm{b_C})\\
\mathrm{C_t} &= \ \ \mathrm{f_t}\ *\ \mathrm{C_{t-1}} + \mathrm{i_t} * \tilde{\mathrm{C}}_\mathrm{t}\\
o_t &= \sigma(\mathrm{w_o}[h_{t-1},\ X_t] + b_o)\\
h_t &= o_t * \tan h(C_t)
\end{aligned} \tag{3}$$

The matrix and terms for the bias are w and b where forgetting gate is f, input gate is i, cell gate is C and output gate is o. Hence, tan h is sigmoid, and as a result, the tangent activation functions in hyperbolic order. The implementation of the formula shows that LSTM is addressing the long-term RNN dependence problem. Consequently, a neural network prediction model based on LSTM can be set up and detected by derived functions in DDoS attacks identified and detected. PSO-LSTM neural network models with input dataset architecture of NSL-KDD have been employed to improve the predictive accuracy by optimizing the weights of LSTM neural network.

Algorithm 1: LSTM based attack detection

1. Initial step (check→first time) Parameters w and b are initialized by the coordinator node
2. Else
   Receive upgrade parameters from coordinator
   node by Eq. (5)
3. Get DATA local data input
4. Obtain DATA as sample data and break into sub DATA
5. For threads on node n, done in parallel
6. Train instance i∈subDATA
7. Update w and b.
8. Return updates of w and b
9. Send w and b to the node of coordinator
10. Apply phase (2) till termination criteria is satisfied

The process of detection starts with data collection per node. Parallel preparation and updating algorithm of parameters (Algorithm 1), which are dispersed in nature, have been done from the sequential stochastic gradient (SGD). DATA must be the information provided from all distributed n nodes, given preliminary training weights w, a learning rate a and b bias parameter. That DATA local data into the given node might be distributed to samples of subDATA. DATA samples could also be split into subDATA on processor threads at each node. As shown in the above algorithm, the coordinator node starts sending the initial random parameters to the rest of the nodes and the models are trained with distributed nodes. The modified parameters, optimizer and SGD for the node are forwarded to the coordinator node. The Coordinating Node quantifies each node's aggregate update parameters and transmits those parameters to the distributed nodes. The LSTM prediction model has very few parameters but the weights are the most important. To further demonstrate those parameters, the particle swarm algorithm has been used to produce better results.

3.4 Proposed PSO-LSTM Based Attack Detection

PSO prevents the convergence of the network to an optimum nearby solution. LSTM hidden layer weights are considered as the input of the particle swarm. The original LSTM output error is used for measuring the fitness of particle swarm and the efficiency of particles. The unusual initial particle swarm has refreshed the individual extremum and global extremum with its own parameter. The formula for updating the particle's speed and position is given in every iteration process in Eq. (4):

$$\mathcal{V}_{\mathrm{i}}^{k+1} = \omega \mathcal{V}_{\mathrm{i}}^{k} + l_1 r_1 (P_i^k - X_i^k) + l_2 r_2 (P_g^k - X_i^k)$$

$$X_i^{k+1} = X_i^k + \mathcal{V}_{\mathrm{i}}^{k+1} \tag{4}$$

where, $k = 1,\ 2,\ \ldots,\ D$ be the current iteration times $i = 1,\ 2,\ \ldots,\ n$; From Eq. (4), the velocity $\mathcal{V}_{\mathrm{i}}^{k}$ and X_i^k are the finest locations on the particles, and the two learning dimensions, l_1 and l_2, respectively, are l_1 and l_2 equal to 2 due to specific improved convergence. P_i^k is individual extremum, P_g^k is global extremum, w is weight, and k is the current iteration. In the [0,1] range, r_1 and r_2 are random numbers. In the LSTM network, determining the optimum particle position vector in the particle swarm is used as an initial weight sequence before the LSTM algorithm is improved by the PSO algorithm. When formulating the neural network model, the size of each particle can be determined too. The fitness of the particle swarm on the given training set is

considered as the medium square error of each output neuron. Then, the smaller the fitness value is according to the fitness function, the smaller the network performance error is. It also means the efficiency of the respective particles is improved. The particle location is then refreshed continuously to reduce the error in the network output layer. The least amount of error particle is drawn in all phases as the finest current particle.

The input layer of LSTM manages the NSL-KDD to fulfill the requirements of the network data. The hidden layer is designed using LSTM cells as shown in Fig. 5 to build a recurrent neural network of one layer that uses its functions. The output layer provides the effects of the expected results of the attack. The network training uses the PSO optimization mechanism that is used for network training and the iterative mechanism is used to estimate point to point network prediction.

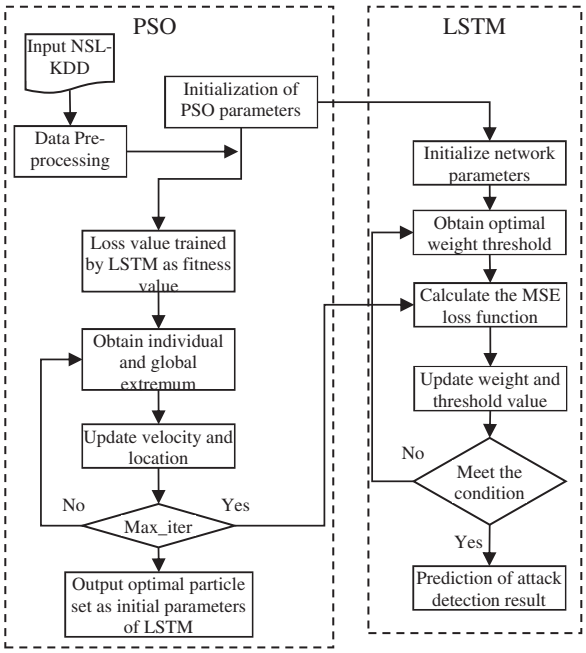

Figure 5: The PSO-LSTM prediction model framework

Using this approach for f'$_t$ data segmentation, the break length value to L is set. As defined in Eq. (5), the segmented prototype is as follows:

$$X = (X_1,\ X_2,\ \ldots,\ X_L)$$

$$X_t = (\mathrm{f}'_t + \mathrm{f}'_{t+1},\ \ldots,\ \mathrm{f}'_{m-L+t-1});\ 1 \le t = L;\ p,\ L \in N \tag{5}$$

Then, X is inserted into the hidden layer of h_t, with L homogeneous. Further, forward and back-linked LSTM cells and X output could be represented after the hidden layer as in Eq. (6):

$$h(h_1,\ h_2,\ \ldots,\ h_L)h_t LSTM_{fo}(X_t,\ \mathfrak{c}_{t-1},\ h_{t-1}) \tag{6}$$

The C_{t-1} and h_{t-1} 1 are the position and output of previous LSTM cells; the cell state vector size is S; the sizes of C_{t-1} and h_{t-1} are S, and here, the hidden layer performance can be seen as ht as well as the model inputs X. Then, chooses the medium square error as the error calculation formula and defines the loss function as in Eq. (7):

$$LSTM_{Loss} = \sum_{i=1}^{L(m-L)} \frac{(h_t - \mathfrak{v}_t)^2}{(L(m-L))} \tag{7}$$

Therefore, a lowest loss function is set as the optimization goal of the PSO and the arbitrary number of particles, velocity V and position are set in the network. Then, PSO is used to optimize the network weight continuously for the final network secret layer PSO. To project the qualified LSTM with $PSOLSTM^*_{\text{net}}$, the assault prediction system is predicated on an iterative approach. First, in Eq. (8), the last line information of the theoretical output Y is provided as

$$y_{\mathfrak{f}} = (\mathfrak{f}'_{m-L+1} + \mathfrak{f}'_{m-L+2}, \ \ldots, \ \mathfrak{f}'_m) \tag{8}$$

Replace the Eq. (8) into $\text{LSTM}^*_{\text{net}}$ and the result is described in Eq. (9);

$$P_{train} = PSOLSTM^*_{net}(y_{\mathfrak{f}}) \tag{9}$$

Then, the last prediction sequence of the test set is shown in Eq. (10) by the means of z-score anti-standardization of X (dez):

$$P_{test} = dez(P_{seq}) \tag{10}$$

The key combination stages of the two algorithms will be described in this process:

1. The PSO parameters specify the correct LSTM network parameters;
2. The latter's performance error is called the former fitness value;
3. Optimized approach is performed by PSO and used as neural network training parameters.

The proposed PSO-LSTM learning method phases follow as given below:

Step by Step procedure for PSO-LSTM Learning Method:

Step 1: Build, standardize and finally evaluate the input and test samples for each LSTM network layer;

Step 2: Identify PSO metrics and then, determine the correlating LSTM network parameters;

Step 3: Generate the starting part position and velocity at random;

Step 4: Train and measure initial performance on the neural network; initialize and normalize the samples for input training and then, finally evaluate the number of neurons on each LSTM network layer;

Step 5: Get the individual extremum and global extremum;

Step 6: Applying the velocity and the position for each of these particles as shown in Eq. (4);

Step 7: If PSO is needed to avoid repetition, save other results and move to step 8 if possible, then proceed to step 4;

Step 8: Taking the global optimal PSO value as the LSTM Neural network weight as in Eq. (7).

Step 9: Using the LSTM algorithm, the neural network will be trained, and if LSTM fulfills the requirements, otherwise repeating steps between 4 and 9, save the results as in Eqs. (8)–(10) and proceed to phase 10.

Step 10: Set the sampling period, Tolerance Factor and Size of the time window

Step 11: While time window←sampling period Analyze the cloud-network traffic

Step 12: Feature extracted from packet header using DBN {SrcIP, DstIP, SrcPort, DstPort, Protocol, Flag, No. of Packet (NP)}

Step 13: Compute entropy value using Eqs. (1) and (2).

Step 14: Find the loss function using Eq. (10)

Step 15: If NP> Tolerance Factor, then Cloud-network traffic may be High-Rate DDoS or legitimate

Step 16: If ID> Tolerance Factor, then Declare High-Rate DDoS Else Declare legitimate

Step 17: End if End if

Step 18: Change (increases) the Tolerance Factor value and repeat the process from step 2.

Step 19: Simulate the test sample and get the prediction result.

The PSO-LSTM model has enhanced both global search and better local search capabilities to achieve the advantages of two algorithms. The experimental comparison shows the size of the particle, the number of 100 iterations, the inertia weight at 0.5, 11, 12 at 1.5, [−5,5] location restrictive interval and velocity limit interval at [−1,1]. As the iteration time increases, the function value decreases, and an individual's condition is that, the neural network model of LSTM uses the fitness function of the particulate swarm optimization.

4 Experimental Results

The experiments of PSO-LSTM have been performed on 64 GB of RAM and 16 core nectar cloud processors. A Keras on Theano package is used to implement LSTM networks' Deep Learning functions and a Cloud computing platform is provided by Apache Spark. Experiments are carried out to detect DDoS attacks using LSTM-RBM which consists of 7 layers, 100 neurons, and 38 visible neurons and NSL-KDD data set contains five classes i.e., normal, dos, R2L, U2R and probe. Column-wise data set normalization is between the intervals of [0,1]. Five epochs are used to train the Network. Weights are selected at random for the 100 used hidden neurons. NSL-KDD and 20 Percent data set are used for training and testing the network. Thus, the number of training data rows contains 25,194 samples and the test data include 4,508 samples. Experiments are carried out on the datasets with 38 features. 3 Features which do not significantly affect the effectiveness of DDoS attack detection are not taken into consideration during the classification process. This proposed model uses the sigmoid activation function.

In the validation process, the proposed DDOS attack detection algorithm PSO-LSTM is compared with the state of the art intrusion detection methods like standard SVM and LSTM. For the purpose of comparisons, the 'Precision', 'Recall', 'F Measure' and 'Accuracy' measures which are commonly used in the literature, are relied to evaluate each method.

All these evaluation metrics are basically derived from the confusion matrix's four basic attributes which depict the actual and predicted classes. These confusing matrix components are:

True Negative (TN): number of correctly predicted instances as non-attacks.

False Negative (FN): number of incorrectly predicted instances as non-attacks.

False Positive (FP): number of wrongly predicted instances as attacks.

True Positive (TP): number of correctly predicted instances as attacks.

The comparison measures are defined as:

Precision: The ratio of truly classified samples to sum of predicted positive samples is as:

$$Precision = \frac{TP}{FP + TP} \tag{11}$$

Recall: The ratio of possible positive to the sum of true positive as well as false negatives is:

$$Recall = \frac{TP}{TP + FN} \tag{12}$$

F-measure: F_1-score denotes the harmonic mean of precision and recall as follows:

$$F - measure = \frac{2 * (Recall \ * \ Precision)}{(Recall + Precision)} \tag{13}$$

Accuracy: performance of classification is considered as accuracy and it is denoted as proportion of samples of correctly classified and sum of samples:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{14}$$

In the context of intrusion detection, the 'Precision', 'Recall' and 'F Measure' and 'Accuracy' for each method and the various attacks are reported in Tabs. 1–4, respectively. Figs 6–9 show that the number of data in the specified data sets is precisely comparable with data rate 25. The results of PSO-LSTM are denoted in bold which corresponds to the best assessment values.

Table 1: Comparison of accuracy in %

Classes	SVM	LSTM	PSO-LSTM
Normal	85	87	91
Dos	92	94	98
R2L	81	83	90
U2R	89	90	93
Probing	86	88	92

Table 2: Comparison of precision in %

Classes	SVM	LSTM	PSO-LSTM
Normal	88	89	95
Dos	85	90	97
R2L	82	85	87
U2R	83	87	88
Probing	82	92	95

Table 3: Comparison of F-measure in %

Classes	SVM	LSTM	PSO-LSTM
Normal	83	88	94
Dos	88	92	96
R2L	80	84	89
U2R	86	81	90
Probing	89	91	93

Table 4: Comparison of recall in %

Classes	SVM	LSTM	PSO-LSTM
Normal	85	87	92
Dos	91	93	95
R2L	84	87	88
U2R	80	83	85
Probing	84	88	91

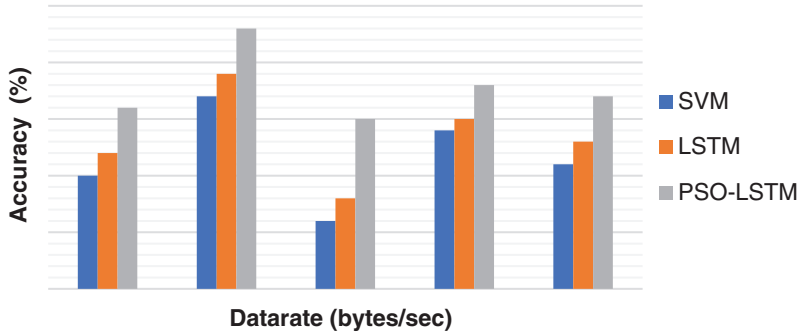

Figure 6: Result of accuracy

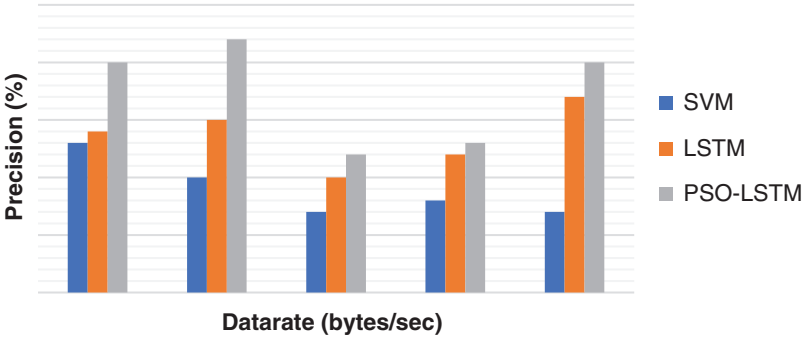

Figure 7: Result of precision rate

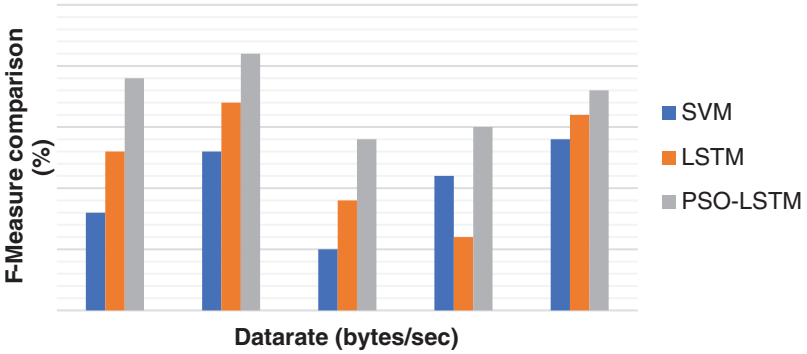

Figure 8: Result of F-measure rate

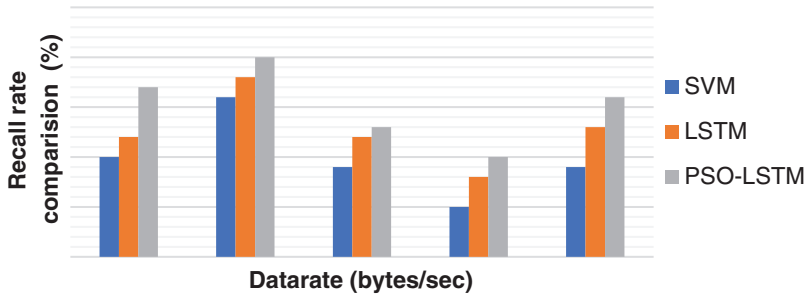

Figure 9: Result of recall rate

According to the Tab. 1 and Fig. 6, one can note that PSO-LSTM method leads to the best accuracy results followed by LSTM and SVM methods. Tab. 2, and Fig. 7 depict that PSO-LSTM obtains the highest precision results and in particular, the PSO has a high convergent rate that speeds up the LSTM method whereas the PSO-LSTM is a better solution for the detection of attacks.

Tab. 3 and Fig. 8 represent that PSO-LSTM obtains the highest F-measure. The reason is that the LSTM parameter is optimized for the entropy function and it reduces the computational time. Hence, it improves it.

Finally, according to the recall measure presented in Tab. 4, Fig. 9 shows that the PSO produces optimum LSTM parameters with a higher probability and the efficient global optima will improve the results of the attack detection. PSO-LSTM leads to the best results which are followed by LSTM and SVM algorithms.

5 Conclusions

For the research of DDoS attacks detection, this work has proposed a method of DDoS attack detection based on deep belief network feature extraction and it is combined with PSO-LSTM model. Primarily, the proposed method mines the feature of IP packets from the data with the use of DBN and it establishes PSO-LSTM model for predicting network traffic to discover DDoS attacks. Here, PSO has been chosen to select the best weight for NN in a way to attain the best accuracy level in classification and prediction. Likewise, LSTM has been chosen to retain memory for long duration of time to make the best classification of attacks. Experiments are conducted on NSL-KDD dataset to evaluate the performance of the proposed method. The result illustrates that the proposed method outperforms all other existing DDoS attacks prediction systems in terms of accuracy, precision, recall and F-measure. It also discovers the DDoS attacks with high accuracy to avoid adverse effect caused by the attack for the services in cloud environment.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] S. Aanjankumar and S. Poonkuntran, "An efficient soft computing approach for securing information over gameover ZEUS botnet with modified cpa algorithm," *Soft Computing*, vol. 24, no. 1, pp. 16499–16507, 2020.

[2] K. Bhushan and B. Gupta, "Security challenges in cloud computing: State-of-art," *International Journal of Big Data Intell*, vol. 5, no. 3, pp. 81–107, 2017.

[3] J. Idziorek, M. Tannian and D. Jacobson, "Attribution of fraudulent resource consumption in the cloud," in *Proc. of IEEE 5th Int. Conf. on Cloud Computing (CLOUD)*, Honolulu, HI, USA, pp. 99–106, 2012.

[4] A. S. Aneetha and S. Bose, "The combined approach for anomaly detection using neural networks and clustering techniques," *Computer Science & Engineering*, vol. 1, no. 1, pp. 37–45, 2012.

[5] R. Bahman, F. Carol and B. Elisa, "A collaborative DDOS defence framework using network function virtualization," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 5, pp. 1162–1175, 2017.

[6] J. Kim, J. Thu and H. Kim, "Long short term memory recurrent neural network classifier for intrusion detection," in *2016 Int. Conf. on Platform Technology and Service (PlatCon)*, Jeju, Korea (South), pp. 1053–1062, 2016.

[7] M. Bi, A. Wang and X. Jian, "DDOS attack detection system based on analysis of users, behaviors for application layer," in *IEEE Int. Conf. on Embedded and Ubiquitous Computing (EUC)*, Guangzhou, China, pp. 582–596, 2017.

[8] R. Brag, E. Mota and A. Passito, "Lightweight DDoS flooding attack detection using nox/openflow," in *Proc. 35th IEEE Conf. Local Computer Networks (LCN)*, Denver, CO, USA, pp. 408–415, 2010.

[9] Y. C. Chen, A. Tseng and T. Lin, "Deep learning for malicious flow detection," in *IEEE 28th Annual Int. Symp. on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Montreal, QC, Canada, pp. 1–7, 2017.

[10] C. Wang, T. T. Miu and X. Luo, "SkyShield: A sketch-based defense system against application layer DDoS attacks," *IEEE Transactions on Information Forensics and Security*, vol. 31, no. 8, pp. 559–573, 2018.

[11] L. Joseph, M. Philippe and N. Syed, "Scalable cloud defenses for detection, analysis and mitigation of ddos attacks," *Future Internet Assembly*, vol. 9, no. 5, pp. 127–137, 2010.

[12] H. Luo, Z. Chen and V. Athanasios, "Preventing distributed denial-of-service flooding attacks with dynamic path identifiers," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 8, pp. 1801–1815, 2017.

[13] C. Ma, X. Du and L. Cao, "Analysis of multi-types of flow features based on hybrid neural network for improving network anomaly detection," *IEEE Access*, vol. 7, pp. 148363–148380, 2019.

[14] A. Sarra and G. Rose, "DDoS attacks in service clouds," in *48th Hawaii Int. Conf. on System Sciences (HICSS)*, Kauai, HI, USA, pp. 5331–5340, March 2015.

[15] Y. Shui, T. Yonghong and G. Song, "Can we beat ddos attacks in clouds?," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 9, pp. 688–673, 2014.

[16] L. Yang, T. Zhang and J. Song, "Defense of DDoS attack for cloud computing," in *IEEE Int. Conf. on Computer Science and Automation Engineering (CSAE)*, Zhangjiajie, China, 2012.

[17] G. E. Hinton, "A practical guide to training restricted boltzmann machines," *Neural Networks Tricks of the Trade Lecture Notes in Computer Science*, vol. 7700, no. 25, pp. 599–619, 2012.

[18] A. Graves, *Long Short-Term Memory*, Berlin: Springer, pp. 1735–1780, 2012.

[19] S. Kai, S. Richard and D. Christopher, "Improved semantic representations from tree-structured long short-term memory networks," in *Proc. of the 53rd Annual Meeting of the Association for Computational Linguistics*, Beijing, pp. 1556–1566, 2015.