Journal of King Saud University – Computer and Information Sciences xxx (xxxx) xxx

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

A deep learning based intelligent framework to mitigate DDoS attack in fog environment

Rojalina Priyadarshini *, Rabindra Kumar Barik

KIIT University, Bhubaneswar, India

ARTICLE INFO

Article history:
Received 28 September 2018
Revised 16 April 2019
Accepted 17 April 2019
Available online xxxx

Keywords:
Fog computing
Deep learning
DDoS attack
Software defined network
Openflow network

ABSTRACT

Fog computing (FC) is a contemporary computing paradigm that gives additional support to cloud environment by carrying out some local data analysis in edge of the devices, facilitating networking, computing, infrastructure and storage support as backbone for end user computing. Still enterprises are not convinced to use this as security and privacy are most of the open and challenging issues. Availability among the security requirements is the one which is about rendering on demand service to different client applications without any disruptions. It can often be demolished by Denial of service (DoS) and distributed denial of service (DDoS) attacks in fog and cloud computing environment. In this paper we propose a novel Source based DDoS defence mechanism which can be used in fog environment as well as the cloud environment to mitigate DDoS attacks. It makes use of Software Defined Network (SDN) to deploy the DDoS defender module at SDN controller to detect the anomalous behavior of DDoS attacks in Network/Transport level. The proposed work provides deep learning (DL) based detection method which makes use of the network traffic analysis mechanisms to filter and forward the legitimate packets to the server and can block the infected packets to cause further attacks.

1. Introduction

In Cloud computing (CC) security is a challenge and has been a major concern for both industry and academia. Many researchers are working hard in last couple of years to cater to different security needs in CC. FC has been evolved as a new computing paradigm which is useful for giving some extra support to cloud environment (Buyya and Dastjerdi, 2016) which is formally introduced by Cisco (C.G.C. Index, YYYY). It functions in a similar way as cloud do, but is not centralized as cloud. Fog systems can be leveraged with doing some local data analysis in edge devices, facilitating networking, computing, infrastructure and storage support as backbone for end user computing (Khan et al., 2017). FC is a distributed paradigm that provides cloud-like services to the edge of network (Xiao and Xiao, 2013; Mahmud et al., 2018). Among the security requirements of FC, availability is one of the core requirements which intends to provide on demand service to different client applications. DoS and DDoS attacks are the kind of attacks which can demolish the availability (Yan et al., 2016). The intention behind DoS attacks and DDoS attacks are making a machine or network resource unavailable to its target clients. When these attacks are performed by more than one persons, or bots, it is called as DDoS and DoS, in case attacks are performed by a single person or a system (Silva et al., 2013). A bot is a victimized machine created when a computer is injected through some software as a malware code. So, DoS attacks could be considered as a particular type of DDoS attacks. According to the source of DDoS launch, these can be of two types. The attacks are either launched by TCP, UDP, ICMP and DNS packets to disturb the target clients by exhausting their network resources or could be launched to exhaust the server resources like server's socket, port, memory, database and input output bandwidth. In the former case the attack is network level flooding and in the later case it is known as application level DDoS flooding which is usually performed on a HTTP webpage (Yi et al., 2015). SDN is an emerging technology and its architecture is a novel way to manage networks (Sahoo et al., 2016). SDN architecture separates control plane from the switches and provide its functionality in Controller, which is programmable and is used to process the incoming packets of the switches. The packets are first matched in the forwarding table

* Corresponding author.
E-mail address: priyadarshini.rojalina@gmail.com (R. Priyadarshini).

Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.jksuci.2019.04.010
1319-1578/

Please cite this article as: R. Priyadarshini and R. K. Barik, A deep learning based intelligent framework to mitigate DDoS attack in fog environment, Journal of King Saud University – Computer and Information Sciences, https://doi.org/10.1016/j.jksuci.2019.04.010

of switch, if not present then it is sent to controller for processing. During the Distributed Denial of Service(DDoS) attack, the online services are made unavailable by overwhelming the same with unwanted data traffic from multiple sources. The attack occurs in the network layer or the application layer of the compromised system that are connected to the network. Controller, being the central point of a SDN network is extremely vulnerable to such cyber attack and may affect the entire network(Shin and Gu, 2013). Still some of the characteristics of SDN like traffic analysis capability, possessing a centralized control logically, view of global state of the network, and dynamic updation of forwarding rules makes it a suitable choice to detect and to defend against DDoS attacks in both cloud and fog environments (Kreutz et al., 2013). The SDN controller has to operate in a centralized way. So there are ample chances for having DDoS attacks on the SDN controller for leveraging the cloud to fog. For designing a secure and reliable SDN controller some methodologies can be imposed to defend against these attacks. SDN based DDoS defence mechanisms could be categorized into three classes basing on their deployment locations. These could be (1) Source based mechanism (2) Network based mechanism and (3) Destination based mechanism (Mirkovic and Reiher, 2004). In this paper we are trying to build upon a source based defence mechanism to handle DDoS attacks where the SDN controller detects the anomalous data traffic, recognize the malicious packets and validate the source IP in the vicinity of the ingress network.

The contributions towards this paper are listed below:

1. We propose a novel Source based DDoS defence mechanism which can be used in fog environment as well as the cloud environment to mitigate DDoS attacks.
2. It uses SDN technology where the DDoS defender module is deployed to defend against Network/Transport level DDoS attacks.
3. The proposed work provides a deep learning (DL) based detection method which successfully detect the DDoS infected packets and can block the same packet from being propagated to cloud.

The remaining paper is arranged as follows. Section 2 will describe work done so far in this context. Section 3 will portray the system model and the necessary conditions for DDoS attack. Section 4 will present the proposed model to design the defender module. The experimental setup is provided in Section 5. The results and their analysis is given in Section 6. Section 7 will conclude the paper, which also depict the future work.

2. Related work

For designing solutions against DDoS attack prevention and detection, both statistical and machine learning based methods are used (He et al., 2017). This section is discussed with numerous noteworthy works and investigations made with the aim of resolving the DDoS attack problem in accordance with SDN and machine learning. The key utility of DDoS revealing and alleviation schemes which use SDN are implemented with the help of a centralized SDN controller. The behaviour and power of centralized SDN controller is exploited to design intrusion detection mechanisms for DDoS detection. On the other hand the widely used machine learning methods that are used for DDoS attacks are Naive bayes, K-Nearest neighbourhood, Support Vector Machine (SVM), neural networks, random forest models and decision tree. A hidden markov model is used along with reinforcement learning to isolate DDoS packets from normal packets by Xu et al. (2007). Their model was based on computing the probability of the incoming IP address sequence. Berral et al. have used the intermediate data traffic, Naive Bayes algorithm was used to identify the DDoS packets with the help of source, destination IP address and some other shared information from the network nodes (Berral et al., 2008). Genetic Algorithm (GA), SVM are used by Shon et al. where the features from data traffic are selected by GA, and SVM is used as a pure classifier to detect DDoS packets (Shon et al., 2005). Zecheng He et al. use machine learning algorithms such as SVM, Naive Bayes to propose a DDoS detection system. They have employed the statistical features of virtual machines and cloud server to impede the packets to move outside of the network (He et al., 2017). Neural network and biological danger theory could also be employed in SDN to alleviate the DDoS attack. In these cases the associated risk is computed for every host and the same is sent to the very VM that keeps an eye on incoming data flow. If the computed risk factor of the in-flow traffic is more than some predefined assessment then some commands and orders are propagated to the SDN controller to offer some defence mechanisms (Mihai-Gabriel and Victor-Valeriu, 2014) to act on. Manikopoulos and Papavassiliou have used a combination of neural network with a statistical method to defend against DDoS (Manikopoulos and Papavassiliou, 2002). Klomogrov-Smirnov test was carried out on the data traffic to obtain the similarity measures and then Neural networks are used to classify the packets. Seufert and O'Brien extract the features from different protocol layers and resources of the system (Seufert and O'Brien, 2007). Their idea was that, by taking the behaviour of system resources, along with network characteristics will be meaningful; because at the time of attack, the system resources are overwhelmed. Kumar et al. used a resilient back propagation algorithm to build the defense system (Kumar and Selvakumar, 2011). All these methods are using shallow algorithms to build the DDoS attack defense mechanisms. These shallow algorithms have their limitations like (1) An extensive experimental analysis is required to capture the relevant statistical features which will increase the generalization performance of the learning algorithm. (2) The models need to be trained regularly to learn the new characteristics of incoming traffic. Yuan et al. (2017) used a deep learning based solution named as DeepDefence to identify DDoS packets. They have tested several DL models to categorize the normal traffic and DDoS traffic. They have used Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Long Short-Term Memory Neural Network (LSTM), and Gated Recurrent Unit Neural Network (GRU) in their work and achieved a significant reduction in error rate compared with the conventional shallow methods. Li et al. (2018) also have used DL along with SDN to mitigate the DDoS attacks and got an accuracy of 99% in training phase and 98% for test data. In this piece of work a variation of DL algorithm, Long Short Term Memory (LSTM) is being imparted to design a solution for DDoS mitigation which is especially tested in a Fog and Cloud environment. The reason behind choosing LSTM is its suitability to handle sequential and time dependent data. Further more the LSTM model is improvised by using dropout probability model, which can avoid over-fitting problem. It also uses a Mini Batch(MB) gradient descent algorithm which is mainly used to fight against vanishing gradient problem. The proposed model is tested on the standard dataset ISCX 2012 along with some real data. For getting these data, a test bed is built in a simulated environment, where the DDoS attacks are created through an open source tool- HPing-3. The attack is initiated by some random virtual machine. For these attacked packets, the common patterns are found out by our model which helps them to segregate between the attacked and normal packets. Further more, our model is enhanced by using various dropout probability,which in turn is able to give a prediction accuracy of 98.88% on the test data.

Please cite this article as: R. Priyadarshini and R. K. Barik, A deep learning based intelligent framework to mitigate DDoS attack in fog environment, Journal of King Saud University – Computer and Information Sciences, https://doi.org/10.1016/j.jksuci.2019.04.010

3. System model

Definition 1. Network $N = \{FNd, Sw, L_n, R_t\}$, the network where DDoS attack is happened. $FNd = \{fnd_1, fnd_2, \ldots..\}$ is the fog environment consisting of fog nodes represent in network N. $S = \{Sw_1, Sw_2, \ldots\}$ set of switches, Ln={lnk1, lnk2, ...s}, set of links exist between S and FN. Rt paves the path, consisting of a rote $P < fnd_{souce}, fnd_{dest} >$ from source fog node fnd_{source} to fnd_{dest}.

Definition 2. Target Fog-Controller (TF):TF_nd is the target node where SDN controller has been deployed as a centralized controller. At the time of attack, the controller's resources would be exhausted, CPU utilization is raise causing a degradation in overall throughput of the system.

Definition 3. Botnet $BtNt = \{bt_1, bt_2, \ldots..\}$ set of zombies causing the attack by invoking the function $Attack(TFi, t)$ which means that bt_i sends infected request packets to TF_ith target fog controller.

3.1. Necessary condition for DDOS attack

In the presented system model, there are two necessary conditions for DDoS attack which are depicted below:

Condition 1. There must exist a data path in between Bt and Tf depicted as $P < fnsouce, fndest >$. The connecting path might be direct or indirect. It is direct if there is a single link (ln) between BN and TF i,e ln = 1 and indirect if it contains multiple links $(ln > 1)$.

Condition 2. To perform the attack, the Botnet may contain a large number of zombies. e,g $BN = \{bn_1, bn_2, \ldots.., bn_n\}$, where $(n > th)$, (Here th is the threshold value of number of attackers so that the resources of the FN is exhausted more than 50%).

Assumption 1. The transmission path followed for data forwarding from the controller and the cloud server is secured.

Assumption 2. The model is trained in a regular basis with the new incoming data traffic entering into the edge network to build the model more robust.

4. Deep learning model

To identify the DDoS attack, the goodness of Deep learning is used in this work. A Long Short Term Memory (LSTM) network is used as it works well for time dependent sequential data (Hochreiter and Schmidhuber, 1997). It is independent of window size and it retains the knowledge of previous packet's effect on the current packet. For a particular time t continuous network packets are captured to form an input window. The pretrained model which is already learned the generalized pattern exhibited in legitimate and malignant packets from some historical data can be able to make difference between the benign and malignant incoming packets (Diro and Chilamkurti, 2018).

Let's consider there are 'S' number of switches and 'P' number of packets are traversing at t_th instance of time. The controller is connected with switches at t_th instance is represented as $C[i,j]$ where, ith switch is transmitting jth packet. Each packet $C[i,j]$ is characterized by several feature captured during packet analysis is given by a matrix $[f_1, f_2, \ldots, f_n]$, All the 192 features are collected and stored as a matrix of 192 columns, each column represents a feature. So the final set of all transmitted packets (P) through switches (S) are stored in a form of a multi dimensional list $List_L$; which is given by:

$$\{List_L\}_d^n = \{C_{1,1}, C_{1,2}, \ldots, C_{S,P}\} \tag{1}$$

where, d = 192 representing the number of dimensions of the data and n = number of data packets, where each packet represents an instance. All these features contain a mixture of text value, numerical value and Boolean value. The Boolean values are encoded into binary values. The text values are also encoded to 16 bit of binary values. For converting the text values to binary bits a vector space model is used. The $[d * n]$ matrix is sliced into windows of a particular time slots δt. Each window is labeled either as 0 or 1. 0 indicates normal window and 1 indicates the packets captured during attack.

In an LSTM there are three gates and one Cell state, named as Forget gates, Input and Output gates. These are used to get rid of vanishing gradient problem which may encounter in RNN. The components of LSTM are represented by the following equations. Here, f_t represents the forget gate and i_t represents the input gate. For each gate different weight sets are used in LSTM which are represented by matrices given by Wf, Wi, Wc and Wo. The input at a given time 't' is given by Xt, Ct' is the intermediate state and Ct is the cell state. The non-linear activation function sigmoid is used to generate the output of forget, input and output gate. This is given by Eq. (2). i_t, f_t and o_t are the input, forget and output gate respectively. The non-linear function $tanh()$ activation is used to generate the intermediate state is given by Eq. (3). The current state is presented as St whereas its previous state is given by $St - 1$. The architecture of LSTM is provided in Fig. 1.

$$f(X_t) = \frac{1}{1 - \exp^{\alpha Xt}} \tag{2}$$

where, α is a constant and said to be as learning rate parameter.

$$f(X_t) = tanh(X_t) \tag{3}$$

$$f_t = \sigma(W_f S_{t-1} + W_f X_t) \tag{4}$$

$$i_t = \sigma(W_i S_{t-1} + W_i X_t) \tag{5}$$

$$o_t = \sigma(W_o S_{t-1} + W_o X_t) \tag{6}$$

$$C_{t'} = tanh(W_c S_{t-1} + W_c X_t) \tag{7}$$

where $C_{t'}$ is the intermediate cell state.

$$C_t = (i_t * C_{t'}) + f_1 * C_{(t-1)} \tag{8}$$

$$H_t = O_t * tanh(C_t) \tag{9}$$

The input to the proposed model is a multidimensional matrix. There are 32 neurons in each cell, which are connected in a forward direction. In the suggested model lawful and unlawful machines dispatch request to get admittance the cloud server and these messages are propagated to cloud via the intermediate fog devices. The intermediate fog device present in the fog network is responsible for detecting the unlawful packets and handles them with proper defined schemes, allowing only the lawful messages to get access to the cloud server. This causes a reduction of irrelevant traffic and checks the unwanted traffic to reach the cloud and thereby avoiding superfluous utility of cloud resources.

In this proposed work, both benign and malignant packets are transmitted from the client sites which may request to gain access to cloud services. But the entire data traffic before reaching to the cloud service has to pass through the fog layer. The fog layer is constituted with a number of fog devices and a fog server, where the SDN controller is installed. The SDN controller works as a central

Please cite this article as: R. Priyadarshini and R. K. Barik, A deep learning based intelligent framework to mitigate DDoS attack in fog environment, Journal of King Saud University – Computer and Information Sciences, https://doi.org/10.1016/j.jksuci.2019.04.010

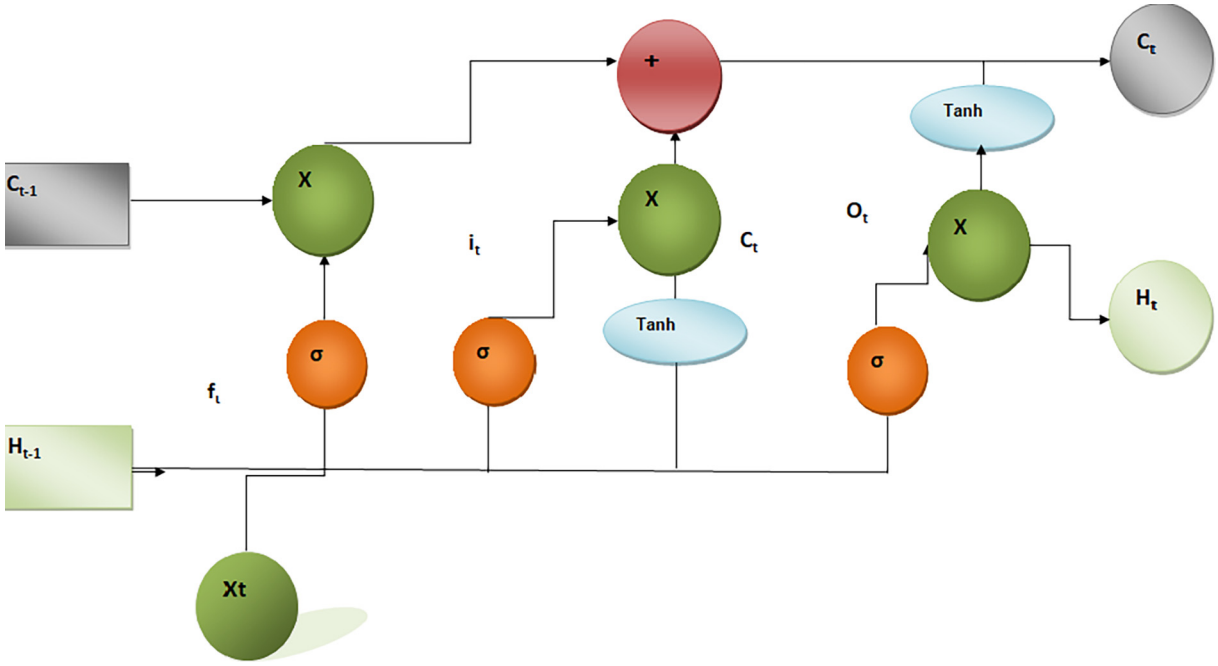

Fig. 1. Architecture of Long Short Term Memory neural network model.

controller that is responsible to inspect all the incoming packets from various nodes. There, the data traffics are filtered and some specific characteristics are captured by the Fog server, which are the deciding features to identify that whether the incoming packet is legitimate or malicious. The attacks are generated from different source machines by means of a variety of tools and scripts. The Fog server is pre-trained with the deep learning algorithm. The algorithms are trained with the captured characteristics of the incoming data traffic. In other words it can be said that, the server is deployed with classifier models to characterize the incoming packets whether are legitimate or malicious. Upon an incoming request, the packets are passed through the classifiers, to decide the requests are legitimate or not. If the packet is found to be the legitimate one, it is forwarded to the cloud server. If it is found to be the suspicious, then the IP address of the corresponding packet is moved to the blocked list of the flow table of switches from SDN controller. Sufficient programming is done at switch level to prevent the packet from being forwarded to the cloud server.

The detail mechanism of the discussed scheme is given in the sequence diagram given in Fig. 2. The user system is consisting of more than one virtual machine. The virtual machines are responsible for transmitting both normal and infected packets, in a given time slot t. The middle layer is the fog network which is built on software defined network (SDN) architecture. The SDN architecture has a controller, an open-flow switch and hosts. The responsibility of SDN controller is to manage the whole network by maintaining the network forwarding table. The flow table present in the switches are updated with each new entry coming to the network, and the network forwarding table updates itself according to the change of state taking place in the flow tables of each switch. Basing on some rules, the SDN controller can forward the packets or drop them. In this work, the controller is capturing the network characteristics, which are the features mentioned in section-1. These features are then passed through a Deep learning detector module. This module is pre-trained with some historical data and is learned to differentiate among DDoS packets and normal packets. Upon receiving the new incoming data traffic, the features are collected and, the detector model takes a decision whether to drop or forward the packet to the cloud sever. For example, lets consider an incoming data packets entered to a network. They may be normal or malignant packets. If a packet enters to our network, its characteristics are captured through HPing-3, and added to the deep defense model as a test input. Then it can predict, whether it is a safe packet. If it is not, then all its information are sent to the SDN Controller. The controller in turn takes some action to prevent this packet to enter into the network. One of the action could be by blocking the IP address in the flow table of the controller.

5. Experimental setup

The system model is configured by setting up a cloud environment, where all the necessary conditions are satisfied. The whole environment is constituted by three layers. The top most layer is comprised of a set of open source software to build the cloud environment. 'Owncloud' is a cloud storage connected with 'Apache' web-server. To setup 'Owncloud' we need PHP, MySQL as pre- requisite. The cloud server is deployed on Cent OS7. The database used in MySQL is 'MariaDB'. MariaDB is a fork of MySQL which is commonly used by linux distribution like CentOS. The fog layer is comprised of few virtual machines, a SDN with an Apache server, where the SDN controller is installed. The application layer includes varieties of legitimate and attacker virtual machines installed with linux, windows OS. The main attacks are performed on TCP, UDP and ICMP protocols through random VMs by using HPing-3. Mininet emulator is used to create a topology for multiple VMs in application layer.

We have used 'FloodLight' controller in the fog server as an SDN controller. It is an Apache licensed Java based controller, which is used to establish the connection links Ln = ln1, ln2, …. between fog nodes Fn and client machines. A DDoS Defence module is developed and configured inside the controller.

The deep learning model is built by using an open source Python library 'Keras' which runs on a 'Tensorflow' background (Priyadarshini et al., 2018). LSTM is explored with 128 hidden neu-

Please cite this article as: R. Priyadarshini and R. K. Barik, A deep learning based intelligent framework to mitigate DDoS attack in fog environment, Journal of King Saud University – Computer and Information Sciences, https://doi.org/10.1016/j.jksuci.2019.04.010

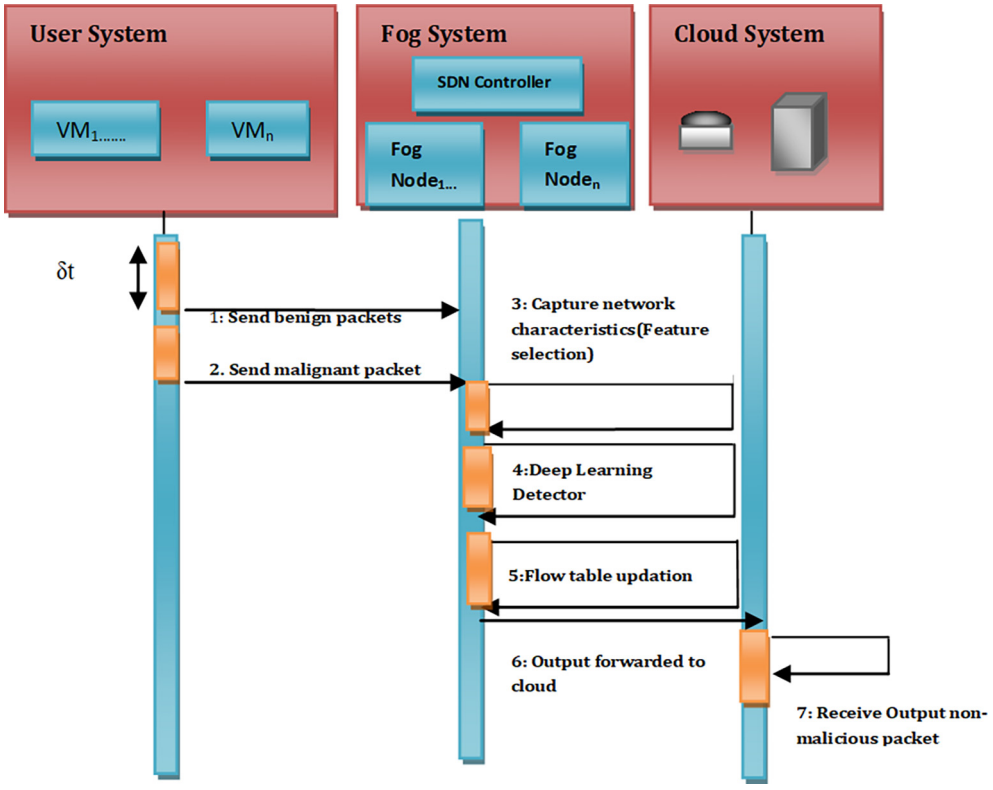

Fig. 2. State-Chart of the proposed model.

rons, the loss function used is 'binary cross-entropy', and 'Adam's' optimizer used with a dropout probability as 0.2. The model uses 2 hidden layes with 128 hidden neurons, which uses Sigmoid, where as the output layer uses a tanh() activation function. As because, the model is using a Sigmoid activation function, it may suffer from vanishing gradient problem. To avoid this, without using batch gradient descent, a Mini-batch gradient descent (GD) algorithm is used. In Mini-batch GD, after iterating up to a fixed mini-batch size, the learning restarts with a new mini-batch, there by reducing the chance of exploding gradient. Here, the mini-batch size is chosen as 512 iterations. The python execution environment is run on Anaconda distribution on a Windows-10 operating system.

6. Results and analysis

6.1. Training of DDoS defence model

For training the Deep learning based DDoS defence module, Hogzilla Dataset is used in this work to train and validate the proposed model. This dataset is extracting data from CTU-13 Botnet (Garcia et al., 2014) and the ISCX 2012 IDS (Shiravi et al., 2012) datasets. In these data, each flow has 192 behavioural characteristics. The dataset CTU-13 botnet carries all the features concerning the attacked packets and ISCX 2012 IDS dataset contain information regarding normal packets. The dataset contains three types of fields. They are numerical, categorical and Boolean. The categorical fields are represented as binary strings by using One-hot encoding scheme(Cassel and Lima, 2006). In this scheme each categorical attribute is converted into an equivalent 16 bit binary strings. Table 1 depicts the attribute details regarding this.

The results produced in this section are produced by running the deep learning model on the CTU-13 Botnet and the ISCX 2012 IDS datasets. The split among training and test sample is 90:10. It means that, 90% of entire data sample is used as training and remaining 10% data are taken as validation data as well as test data. We have used a 10-cross validation scheme for validating the output. During this, the total data samples are divided equally into 10 divisions, from which 9 divisions are randomly chosen as the training samples and remaining one division will go for testing. This process is repeated for 10 times and then average of all iterations is taken as the final result. The model's parameters are also tried to be changed. The model is tried with 1 hidden layer, 2 hidden layers then again by changing the number of hidden nodes, initially from 32 nodes to 64 and then 128 nodes. Similarly the drop out probability is initially set as 0.1 and later on settled at 0.2. The dropout probability is used to avoid over-fitting problem and for quick response in recurrent neural network; where the visible and hidden units of the neural networks are removed temporarily along with their incoming and out-going connections (Srivastava et al., 2014). Initially the network is trained with dropout as 0, and then tried with 0.1 to 0.3. But the model was tuned with 0.2. It can be observed from the results that, the model of LSTM with 2 hidden layers with dropout rate 0.2 is performing well. The parameters of the model are fine tuned with analyzing the results after changing the parameters and repeating the experiments. Fig. 3 represents the percentage of accuracy with respect to training and testing instances. Fig. 4 considers only the test data and draws a comparative graph of the error rate on variants of LSTM.

Table 1
Details of the dataset attributes used in the proposed model.

Total No. of fields	192	No. of classes	3
No. of Categorical fields	4	No. of bits required to represent categorical fields	16
No. of Numerical field	9	No. of bits required to represent numerical fields	Nil
No. of Boolean fields	179	No. of bits required to represent Boolean fields	2

Please cite this article as: R. Priyadarshini and R. K. Barik, A deep learning based intelligent framework to mitigate DDoS attack in fog environment, Journal of King Saud University – Computer and Information Sciences, https://doi.org/10.1016/j.jksuci.2019.04.010

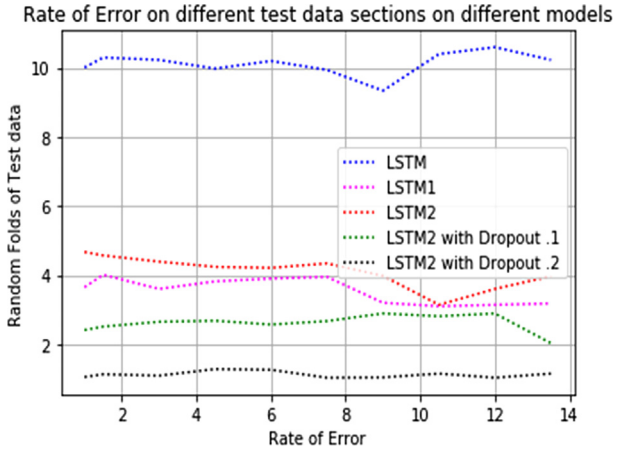

Fig. 3. Comparison of Training and Testing accuracy of LSTM-2.

Fig. 5. Rate of Error for different LSTM models.

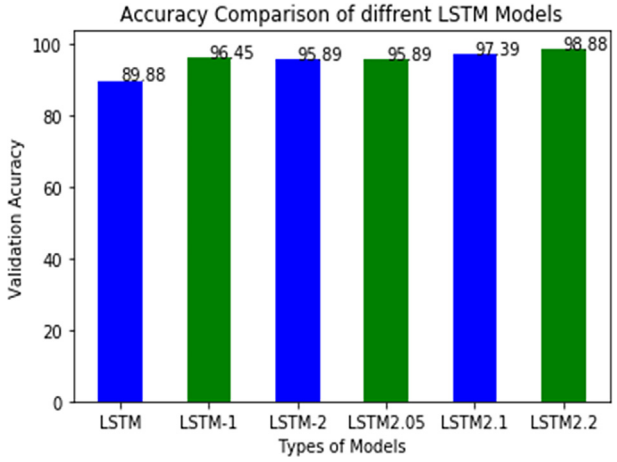

Fig. 4. Validation accuracy of different LSTMs.

Fig. 5 shows the accuracy of LSTM variants on the test data. It can be observed that, LSTM2.2 is outperforming in terms of accuracy over others. The performance on training data sample is almost 99.12% and on the test sample is also promising, which is coming as 98.88%.

6.2. DDoS defence model test

The Testing of built model is carried out with two types of data. At first, the model is tested on the Hogzilla dataset, from which 10% of the total dataset is exploited as the test samples. Along with this, some real DDoS attacks are performed and a test bed is produced to validate the model. The network traffic is extracted using TCPDump. TCPDump is an automated tool to monitor network statistics. To simulate the DDoS attack a tool known as Hping3 is being used by which both malicious and non-malicious data are collected separately. DDoS attacks are performed on TCP, UDP and ICMP protocols through some random virtual machines with the help of Hping3 open source tool. These attacked packets are then passed through the Deep learning model, the results are presented in Table 2, where the performance measures are chosen as accuracy percentage on test data on all LSTM variants with varying number of hidden neurons. Again these LSTMs are tested without dropout probability and with dropout probability as 0.1 and 0.2. It can be observed that, the LSTM model with 3 hidden layers, consisting 128 number of input nodes and with drop-out rate as 0.2 is

Table 2
Types of Models with different parameters and their performance.

Model Type	LSTM with no hidden layers	LSTM with 1 hidden layers (LSTM-1)	LSTM with 2 hidden layers (LSTM-2)	LSTM with 3 hidden layers (LSTM-3)
Activation Function	Sigmoid, Tanh	Sigmoid, Tanh	Sigmoid, Tanh	Sigmoid, Tanh
	Validation Accuracy Percentage			
No hidden neurons = 32	87.67	91.33	94.68	93.67
No hidden neurons = 64	87.96	96.98	95.67	97.45
No hidden neurons = 128	89.88	96.45	95.89	97.21
Dropout = 0.0	89.88	96.45	95.89	93.29
Dropout = 0.1	90.13	91.57	97.39	96.78
Dropout = 0.2	90.98	92.89	98.88	98.34

Table 3
Performance Comparison of DDoS Defence Model with other existing DL Models.

Model Type	Training Accuracy	Testing Accuracy	Dataset Used	Used in Cloud and Fog
Stacked Auto Encoder (Niyaz et al., 2016)	NA	95.65	Captured Data	No
LSTM (Yuan et al., 2017)	99.00	98.00	ISCX 2012	No
LSTM-2 Dropout = 0.2	99.48	98.88	ISCX 2012, Real Data	Yes

Please cite this article as: R. Priyadarshini and R. K. Barik, A deep learning based intelligent framework to mitigate DDoS attack in fog environment, Journal of King Saud University – Computer and Information Sciences, https://doi.org/10.1016/j.jksuci.2019.04.010

out performing then the others. Table 3 represents the comparison of DDoS Defence model with the existing models which used DL along with SDN in past. It can be found out, LSTM 2.2 is giving some promising results for test data.

7. Conclusion

In this work we have designed a deep learning based model to protect a Fog network from DDoS attacks. We used SDN technology to control the whole Fog network. The open flow based SDN network is exploited and is equipped with a DDoS defense module which makes use of deep learning technology. LSTM model is chosen among all the other deep learning varieties. Because, LSTM works well for sequential data, and the data packets used for DoS detection are time collected in. The deep learning model is trained with the historical data and tested with both simulated and real DDoS attack packets. The model has experimented on different parameters to get a set of optimized performance tuners. LSTM with 3 hidden layers, one dense layer, 128 input nodes and where a dropout rate is 0.2 for all the hidden layers is giving a good performance indicator in terms of growing accuracy and reduced error rate. For setting up the fog environment we have used a real cloud setup wih some open source cloud platform, SDN controller is configured with 'FloodLight' controller. The controller node is equipped with a DL model which is trained with Hogzilla dataset and is tested on some real time DDoS attack. For causing the DDoS attacks, some open source tools are used. The model is showing 98.88% of accuracy on testing data set. Upon detecting the incoming data packet as suspicious malicious packets, the openflow switch present in SDN can prevent the packets to further propagation to the cloud server. The infected packet is denied for being forwarded to the server, which can prevent the entire fog network from being affected by the DDoS attacks.

Conflict of interest

None.

References

Berral, J.L., Poggi, N., Alonso, J., Gavalda, R., Torres, J., Parashar, M., 2008. Adaptive distributed mechanism against flooding network attacks based on machine learning. Proceedings of the 1st ACM workshop on Workshop on AISec. ACM, pp. 43–50.

Buyya, R., Dastjerdi, A.V., 2016. Internet of Things: Principles and paradigms. Elsevier.

Cassel, M., Lima, F., 2006. Evaluating one-hot encoding finite state machines for seu reliability in sram-based fpgas. On-Line Testing Symposium, 2006. IOLTS 2006. 12th IEEE International. IEEE, p. 6.

C.G.C. Index, Forecast and methodology, 2015–2020 white paper, Retrieved 1st June.

Diro, A.A., Chilamkurti, N., 2018. Distributed attack detection scheme using deep learning approach for internet of things. Future Generation Comput. Syst. 82, 761–768.

Garcia, S., Grill, M., Stiborek, J., Zunino, A., 2014. An empirical comparison of botnet detection methods. Comput. Security 45, 100–123.

He, Z., Zhang, T., Lee, R.B., 2017. Machine learning based ddos attack detection from source side in cloud. Cyber Security and Cloud Computing (CSCloud), 2017 IEEE 4th International Conference on. IEEE, pp. 114–120.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9 (8), 1735–1780.

Khan, S., Parkinson, S., Qin, Y., 2017. Fog computing security: a review of current applications and security solutions. J. Cloud Comput. 6 (1), 19.

Kreutz, D., Ramos, F., Verissimo, P., 2013. Towards secure and dependable software-defined networks. Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking. ACM, pp. 55–60.

Kumar, P.A.R., Selvakumar, S., 2011. Distributed denial of service attack detection using an ensemble of neural classifier. Comput. Commun. 34 (11), 1328–1341.

Li, C., Wu, Y., Yuan, X., Sun, Z., Wang, W., Li, X., Gong, L., 2018. Detection and defense of ddos attack–based on deep learning in openflow-based sdn. Int. J. Commun Syst 31, (5) e3497.

Mahmud, R., Kotagiri, R., Buyya, R., 2018. Fog computing: A taxonomy, survey and future directions. Internet of Everything. Springer, pp. 103–130.

Manikopoulos, C., Papavassiliou, S., 2002. Network intrusion and fault detection: a statistical anomaly approach. IEEE Commun. Mag. 40 (10), 76–82.

Mihai-Gabriel, I., Victor-Valeriu, P., 2014. Achieving ddos resiliency in a software defined network by intelligent risk assessment based on neural networks and danger theory. Computational Intelligence and Informatics (CINTI), 2014 IEEE 15th International Symposium on. IEEE, pp. 319–324.

Mirkovic, J., Reiher, P., 2004. A taxonomy of ddos attack and ddos defense mechanisms. ACM SIGCOMM Comput. Commun. Rev. 34 (2), 39–53.

Niyaz, Q., Sun, W., Javaid, A.Y., 2016. A deep learning based ddos detection system in software-defined networking (sdn), arXiv preprint arXiv:1611.07400.

Priyadarshini, R., Barik, R.K., Panigrahi, C., Dubey, H., Mishra, B.K., 2018. An investigation into the efficacy of deep learning tools for big data analysis in health care. Int. J. Grid High Performance Comput. (IJGHPC) 10 (3), 1–13.

Sahoo, K.S., Mohanty, S., Tiwary, M., Mishra, B.K., Sahoo, B., 2016. A comprehensive tutorial on software defined network: The driving force for the future internet technology. Proceedings of the International Conference on Advances in Information Communication Technology & Computing. ACM, p. 114.

Seufert, S., O'Brien, D., 2007. Machine learning for automatic defence against distributed denial of service attacks. Communications, 2007. ICC'07. IEEE International Conference on. IEEE, pp. 1217–1222.

Shin, S., Gu, G., 2013. Attacking software-defined networks: a first feasibility study. Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking. ACM, pp. 165–166.

Shiravi, A., Shiravi, H., Tavallaee, M., Ghorbani, A.A., 2012. Toward developing a systematic approach to generate benchmark datasets for intrusion detection. Comput. Security 31 (3), 357–374.

Shon, T., Kim, Y., Lee, C., Moon, J., 2005. A machine learning framework for network anomaly detection using svm and ga. Information Assurance Workshop, 2005. IAW'05. Proceedings from the Sixth Annual IEEE SMC. IEEE, pp. 176–183.

Silva, S.S., Silva, R.M., Pinto, R.C., Salles, R.M., 2013. Botnets: a survey. Comput. Netw. 57 (2), 378–403.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15 (1), 1929–1958.

Xiao, Z., Xiao, Y., 2013. Security and privacy in cloud computing. IEEE Commun. Surveys Tutorials 15 (2), 843–859.

Xu, X., Sun, Y., Huang, Z., 2007. Defending ddos attacks using hidden markov models and cooperative reinforcement learning. In: Pacific-Asia Workshop on Intelligence and Security Informatics. Springer, pp. 196–207.

Yan, Q., Yu, F.R., Gong, Q., Li, J., 2016. Software-defined networking (sdn) and distributed denial of service (ddos) attacks in cloud computing environments: A survey, some research issues, and challenges. IEEE Commun. Surveys Tutorials 18 (1), 602–622.

Yi, S., Qin, Z., Li, Q., 2015. Security and privacy issues of fog computing: a survey. In: International Conference on Wireless Algorithms, Systems, and Applications. Springer, pp. 685–695.

Yuan, X., Li, C., Li, X., 2017. Deepdefense: identifying ddos attack via deep learning. 2017 IEEE International Conference on Smart Computing (SMARTCOMP). IEEE, pp. 1–8.

Please cite this article as: R. Priyadarshini and R. K. Barik, A deep learning based intelligent framework to mitigate DDoS attack in fog environment, Journal of King Saud University – Computer and Information Sciences, https://doi.org/10.1016/j.jksuci.2019.04.010