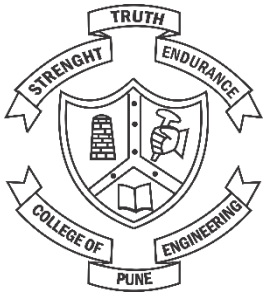
COLLEGE OF ENGINEERING

PUNE



REPORT

ON

"DDOS Attack Detection Using Deep Learning"

INFORMATION SECURITY

Department of computer science and information technology

SUBMITTED BY

Name	**MIS No.**
Mr. Jayesh V Jawade	121942010

Under the Guidance of

DR. V. K. PACHGHARE

Year

2020

INDEX

Sr. No.	Title	Page No.
01	**Introduction ……**	**01**
02	**DDOS Attack Working and Countermeasure's ……..**	**02**
03	**Working of project ……..**	**02**
	3.1 Feature Extraction and Transformation	
	3.2 Bidirectional Recurrent Neural Network	
04	**Implementation Steps ……..**	**04**
05	**Output ……**	**04**
12	**Conclusion ……**	**05**
13	**References ……..**	**05**

DDOS Attack Detection Using Deep learning

1.1 Introduction

A Distributed Denial of Service (DDOS) attack is an attempt to make a resources of a targeted system or bandwidth unavailable by overwhelming it with multiple false requests or traffic from various sources or bots. In simple words, DDOS attack is simply flooding the network bandwidth with false protocol requests from various bots. DDOS attack is usually done by launching the coordinated DOS attack from various systems or bots.

DDOS attack has become most widely used attack in its earlier days and is very critical to detect and even today's world also the DDOS attack is very difficult to detect.

According to the survey conducted by Radware, DDOS attack is currently the largest threat to most of the organizations. The targets to the DDOS attack is mainly from the online financial, gaming, online resources, etc. domains. The duration of DDOS attack is not defined and it may be conducted for large number of time to do more harm to targeted system.

DDOS attack mainly includes requests such as HTTP flood, SYN flood, DNS flood, ICMP flood, UDP flood, etc.

Our main aim for this paper is to find the defense mechanism for DDOS attack. DDOS attack detection is one of the main defense for DDOS attack. But at the same time, DDOS attack is pretty hard to detect with the traditional methods since most of the times hackers manage to launch the attack in such a way that it looks like legitimate traffic. Most of the times machine learning approach is not capable to detect the attack because of requirement of updating the machine learning models.

We propose a deep learning approach to detect the DDOS attack more accurately. For this approach, we used UNB ISCX Intrusion Detection Evaluation 2012 Dataset for training the deep learning model. This dataset consist of the network traffic of two days during the DDOS attack. We use different methods such as Long Short Term Memory (LSTM) and Recurrent Neural Network (RNN) to greatly improve the accuracy while training the model using large dataset. By using such big dataset, the error rate gets reduced compared to other models. This proves that the approach we used is capable of learning from its past experiences.

Deep learning is used for processing such big data in a nonlinear approach by using some neural network architectures or algorithms. In this approach, we use Recurrent Neural Network. Recurrent Neural Network is one kind of neural network in which the output of the network from previous step is feed as the input for next step. Because of this reason, our model is capable of detecting the DDOS attack from previous as well as current packet information. The main problem is this approach is of storing the memory for such long sequence. To solve this problem, we used the Long Term Short Memory (LSTM) model. LSTM has feedback connection with the capability of processing the long sequence of data. In this approach we used the bidirectional LSTM network.

bidirectional_1(lstm_1): Bidirectional(LSTM)	input:	multiple
	output:	multiple

Figure 1: Architecture of bidirectional LSTM

1.2 DDOS Attack Working and Countermeasure's

DDOS attack is simply flooding the network or bandwidth with large traffic or false request so that the resources cannot operate even to legitimate user. This attack is mainly launch from different bots or machines. DDOS attack is very easy to launch and all you need is only the machines to send the false requests to servers or websites or resources.

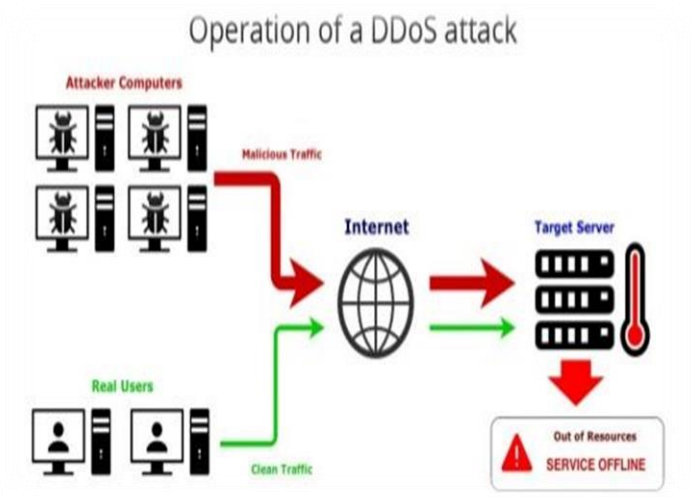


Figure 2: Operation of a DDOS attack

Suppose, we have to launch DDOS attack on a web server as shown in figure. Then the bunch of attacker computer or bots collectively sends the large number of requests to server. The web server is able to handle some definite amount of requests at a time according to its processing power. And after some time if the limit of such false requests increases then the server is not able to respond to new requests as the server is already busy in handling the older requests. Because of this, the new requests are rejected even if they are from legitimate user. Because of such distributed nature of this attack, it's very difficult to identify the legitimate traffic and false traffic and hence it's very commonly used attack.

This type of attack is not considered as a security breach because it doesn't steals any data rather it only overwhelm the network to force it to go out of resource or offline. DDOS attack are possible because of easily compromised devices used by victims or default settings.

Some preventive measures for such type of attacks are as follows:

a. Secure your WiFi router as it is the gateway to your network.
b. Change all the default passwords to connected devices.
c. Use firewalls or intrusion detection systems to detect such attacks.

1.3 Working of project

DDOS attack is very easy to detect when the network traffic is at high rate, but it's equally hard to detect when the traffic rate is low and seems to be similar to legitimate traffic. If the DDOS attack takes place with such low traffic rate and generated over specific period of time, then it is very difficult to detect with traditional methods. Hence, it is very important to consider the historical information about the packet traffic in network. It is not enough to consider only the single packet or forward direction only to accurately detect attack.

Therefore, we implement the approach to detect DDOS attack based on Recurrent Neural Network (RNN).RNN is achieved using LSTM, GRU algorithms. In our approach, the historical information is fed into the RNN model. It is used to detect the repeated patterns representing DDOS attack and find them in a long traffic sequences.

The main advantage of RNN is that it is independence of input window size. In all the previous approaches, the input window size is usually dependent on the task. Because of such dependent nature, it is difficult to detect different types of attacks. Another advantage of RNN is to remember the long sequences of data which is not possible for other machine learning approaches. The accuracy increases with the amount of dataset i.e. length of the history of network packets.

1] Feature Extraction and Transformation

The actual dataset i.e. ISCX 2012 contains total 29 traffic fields. But we first of all extract 20 network traffic fields from that dataset. Because lesser the traffic fields, lesser the processing time. But for extraction we have to keep in mind that only less important fields must be only extracted. We use this 20 traffic fields for training our deep learning model as a training features. We do not have to select the statistical features in our model.

In our dataset, we classify three main types of data fields as Boolean, Numerical and Text fields. In this dataset, we transform the dataset by applying some encoding such as we encode the TCP and UDP port number to 16 bit binary list and for text field we transform the fields into Bags of Words method. For conversion of Bag of Words, we used hashing methods.

Field	Field Example	Field Type
frame.encap_type	1	Boolean
frame.len	805	Numerical
frame.protocols	eth:ip:tcp:http:data: data:data: data-text-lines	Text
http.time	0	Numerical
icmp.length	203	Numerical
icmp.type	3	Boolean
irc.request.command	PONG	Text
irc.response.command	462,JOIN,353,366	Text
tcp.ack	2.692e+03	Numerical
tcp.analysis.ack_rtt	0	Numerical
tcp.analysis.bytes_in _flight	1.460e+03	Numerical
tcp.analysis.duplicate _ack_num	1	Numerical
tcp.dstport	2090	Boolean
tcp.flags.urg	0	Boolean
tcp.len	751	Numerical
tcp.srcport	80	Boolean
tcp.window_size	12864	Numerical
udp.dstport	47666	Boolean
udp.length	97	Numerical
udp.srcport	47521	Boolean

Table 1: Network Traffic Fields

After feature transformation, we get m x n' matrix, where m indicates the number of packets and n' indicates the number of new features after transformation.

In order to learn patterns in both long and short term, we use a sliding window to separate continuous packets and reshape the data into a series of time windows with window size T. The label y in each window illustrates the last packet.
After reshaping,we have a three-dimensional matrix with shape $(m-T) \times T \times n'$. Figure illustrates the workflow of feature extraction, transformation, and reshaping.

In this way, we change the features from conventional packet-based to window-based, by which we can learn network patterns from both previous $(T-1)$ packets and current packet.

2] Bidirectional Recurrent Neural Network

In our approach, we use Bidirectional Recurrent Neural Network (RNN) using Long Short Term Memory architecture. RNN is used to trace the history from previous network traffic packets.

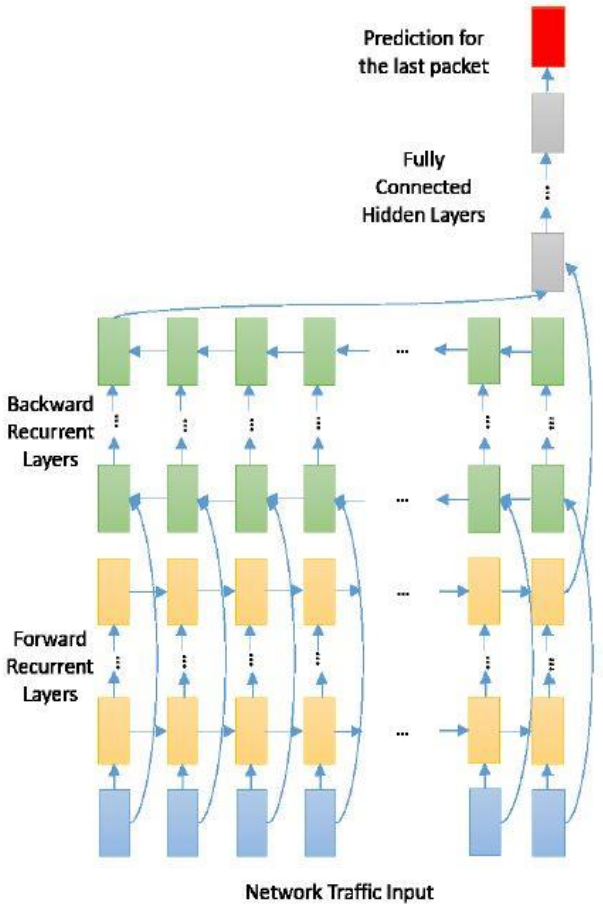


Figure 3: Overall Network Architecture

In this model, each direction includes two sequence-to-sequence Recurrent Neural Layers. The deep learning model use sigmoid function to indicate the prediction of the last packet in the whole sequence. To capture the local information and simplify the deep neuron network, we try to stack 1-Dimensional Convolutional Neural Layers before Recurrent Neural layers with Rectified Linear Unit (RELU) as activation function.

1.4 Implementation Steps

Step 1: Importing the required libraries

Import the required libraries for data processing, performing operations, etc. The libraries such as Numpy library for matrix operations, Pandas library for dataset operations, Matplotlib and Seaborn libraries for graphs plotting,etc. The most important library for deep learning is Tensorflow which is required for creating neural network and this is done very easily by using Keras library which works on top of Tensorflow library.

Step 2: Specify number of sample

Decide the number of samples in dataset which is to be considered for further operations. It is used to select the desired number of rows only to reduce the processing power requirement.

Step 3: Read data from normal and attack dataset

Read the selected dataset files using Pandas library.

Step 4: Feature Extraction

This is most important part of project implementation since it will reduced the unwanted data processing. This includes dropping the unwanted columns from the dataset.

Step 5: Standardize the data

Standardize the dataset for better results.

Step 6: Create a model

Divide the training and testing dataset. Create the model based on the below architecture.

bidirectional_1(lstm_1): Bidirectional(LSTM)	input:	multiple
	output:	multiple
dense_1: Dense	input:	multiple
	output:	multiple
dense_2: Dense	input:	multiple
	output:	multiple

Figure 4: Model Architecture

Step 7: Train the model with dataset

Train the created model by specifying number of epochs, validation_split and verbose.

Step 8: Plot accuracy and loss graphs and confusion matrix

Plot the required graphs and confusion matrix using matplotlib library.

Step 9: Save the model

Save the trained model if required for further use.

1.5 Output

1] Accuracy Model

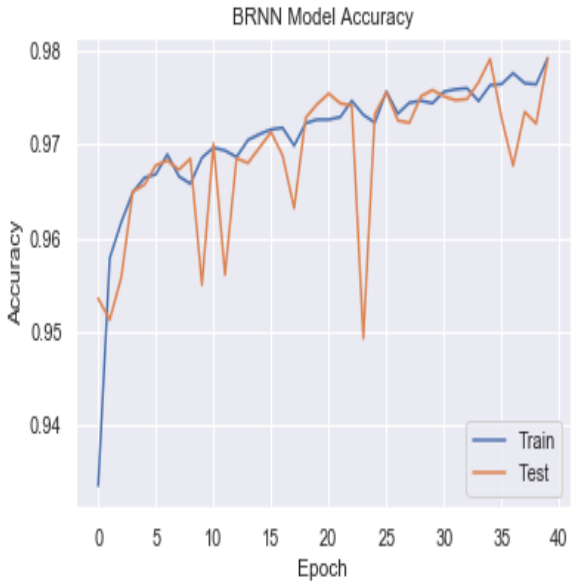


Figure 5: Accuracy of Model

2] Loss Model

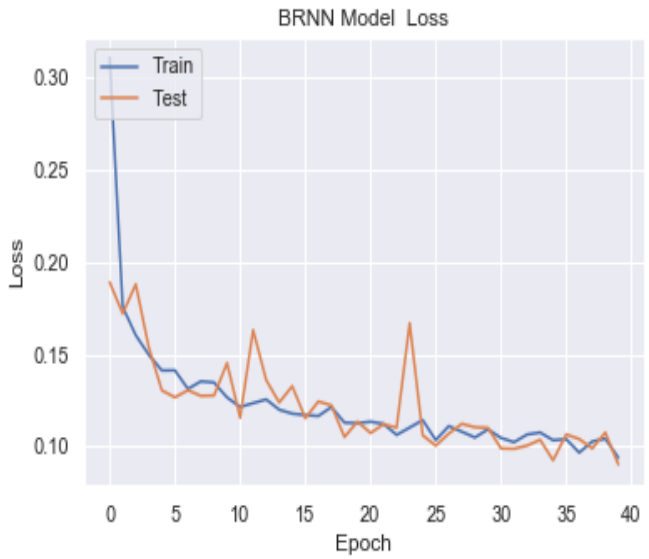


Figure 6: Loss of Model

3] Confusion Matrix

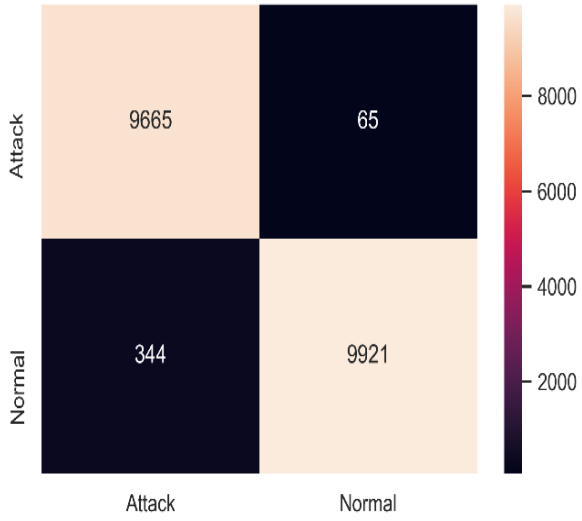


Figure 7: Confusion Matrix

From the figure 7, we conclude that the true positive values are 9665, true negative values are 9921, false positive values are 344 and false negatives are 65.

The definition of above terms are as follows:-

a) True Positive (TP) : Attack is performed and is correctly detected as attack.
b) True Negative (TN) : Attack is not performed and is correctly detected as normal.
c) False Positive (FP) : Attack is not performed but falsely detected as attack.
d) False Negative (FN) : Attack is performed but falsely detected as normal.

1.6 Conclusion

In this paper, we propose the deep learning model to improve the performance accuracy to detect the DDOS attack traffic. Recurrent Neural Network is used to learn longer sequence of packets rather than other machine learning algorithms. Hence, we used this RNN model in our paper. The Recurrent Neural Network using bidirectional LSTM gives the accuracy of 98%.

The accuracy of DDOS detection is increased by using the bidirectional LSTM model (Recurrent Neural Network in deep learning) by training the model on limited dataset (packets).

Because of using limited dataset, the computational power required for operation is less, which in turn reduce the time required for training without affecting its accuracy.

1.7 References

[1] X. Yuan, C. Li and X. Li, "DeepDefense: Identifying DDoS Attack via Deep Learning," *2017 IEEE International* Conference on Smart Computing (SMARTCOMP), Hong Kong, 2017, pp. 1-8.

[2] Ali Shiravi, Hadi Shiravi, Mahbod Tavallaee, Ali A. Ghorbani, “Toward developing a systematic approach to generate benchmark datasets for intrusion detection” 2012 Elsevier Ltd.