Hybrid Deep Learning for Botnet Attack Detection Using LSTM and CNN

[1]Mr.Ganeshkumar S, [2]Manicka Mathavan M, [3]Muniyasamy V, [4]Vijayaprabakar V

[1]Associate Professor, Dept of Computer Science Engineering, Sree Sowdambika College of Engineering, Virudhunagar-626101, Tamil Nadu, India.[2]Student, Dept of Computer Science Engineering, Sree Sowdambika College of Engineering, Virudhunagar-626101, Tamil Nadu, India.
[3]Student, Dept of Computer Science Engineering, Sree Sowdambika College of Engineering, Virudhunagar-626101, Tamil Nadu, India.
[4]Student, Dept of Computer Science Engineering, Sree Sowdambika College of Engineering, Virudhunagar-626101, Tamil Nadu, India

ABSTRACT:

Deep Learning is an efficient method for the botnet attack detection. Usually the volume of network traffic data and its required large memory space. Principal Component Analysis is one of the common linear methods like kernel methods; DL and Spectral methods employ non-linear transformation techniques. It is impossible to implement the DL method in memory constrained IOT devices. We reduce the large-scale IOT network traffic data using to reduction techniques. Bot dataset is the most common dataset that is publicly available for the network based botnet attack detection. These traffic attack samples can be categorized into four botnet scenarios namely: DOS, DDOS, Information theft and Reconnaissance. Auto Encoder is an unsupervised method that produces lack of space representation of input data at the hidden layer. Different auto encoder architectures have been used to reduce the feature dimensionally in most popular intrusion datasets. We have to avoid the long short term memory and to implement the convolutional Neural Network. Finally the results show that the performance likes more accuracy.

Keywords: **Botnet Iot Attack, LSTM, CNN, Auto encoder.**

Introduction:

To bring the expected result for Bot-Net attacks in the network devices a new develop dataset is applied. The dataset contains that normal traffic flows and several numerous of cyber-attacks traffic flows in botnets attacks. For the accurate traffic and for the develop effective dataset, the realistic test bed is used for to develop this dataset with the effective information features and also for the improvement of ML model performance and effective prediction model, mostly it were extracted and added it with the extracted features datasets. However, for the best results, the extracted features datasets are labelled as attack flow, categories, and subcategories. Nowadays, the Internet technology is growing up in day to day, and the varies devices are connected with this technology.

By introducing this technology, daily life becomes more comfortable and well-organized. On the one side, these technologies are developing numerously but with this rapid development and popularization of internet devices causes the increasing number of cyber-attacks in the desktops is called the Bot-Net Attacks. Still it lacks the security in their internet connection software because most of them have not enough storage and computational resource for robust security mechanisms. In this project, it proposed a machine learning (ML) based botnet attack detection mechanism with sequential detection architecture. Its approach is adopted to implement a lightweight detection system with a high performance. Botnet is a network software bots designed to do malicious activities on the target network which are controlled using command and control protocol by the single unit is called the bot master. Bots are infected computers which is controlled remotely by bot master without any sign of being hacked and are used to perform malicious activities. Botnet size varies from small botnet to the large botnets where small botnet consists of few hundred bots and large botnets consists of 50,000 bots. Hackers can attack the system data without any noticeable indication of their presence.

To secure these devices against botnet attacks, ML algorithms have been applied to develop Network Intrusion Detection Systems (NIDS). This NIDS can install at the strategic points within a network. Specifically, Deep Learning, an advanced ML approach, gives the unique capacity for automatic extraction of data from large-scale and high-speed network traffic integrated by interconnecting heterogeneous computer devices. Considering the resource constraints in these devices, NIDS techniques applied in classical networks devices which are not efficient for botnet detection in networks because of high computation and memory requirements. In order to produce an efficient Deep Learning method for botnet attack detection in networks, large network traffic information is required to accept efficient classification performance.

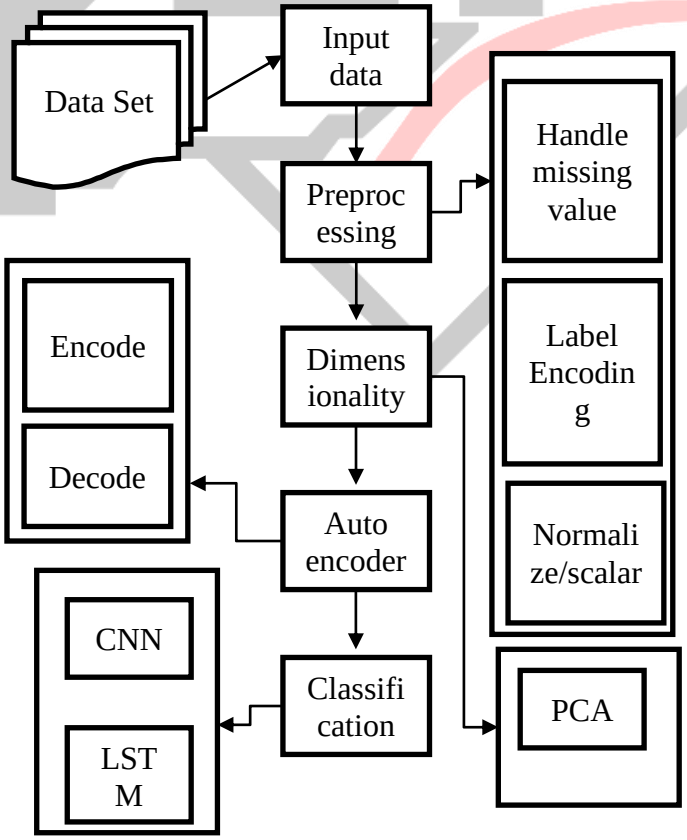
The processing and analysing the high-dimensional network traffic data can make the course of dimensionality. Also, training Deep Learning models with high-dimensional data can cause Hughes phenomena. High-dimensional of data process is complex and it requires huge computational resources and memory capacity. Some of the devices do not have sufficient space to store big network traffic data required for Deep Learning. Therefore, there is a need for end-to-end DL-based botnet detection method to reduce the high dimensionality of big network traffic and also detect the complexity and recent botnet attacks accurately based on

low-dimensional network traffic information. Currently, Bot-Net dataset is the most relevant publicly available dataset for botnet attack detection in networks. The original feature dimensionality of the Bot-Net dataset is 43, and the memory space required for this network traffic data is 1.085 GB. So far, feature dimensionality reductions methods are applied to the Bot-Net dataset were all based on feature selection techniques.

RELEATED WORKS:

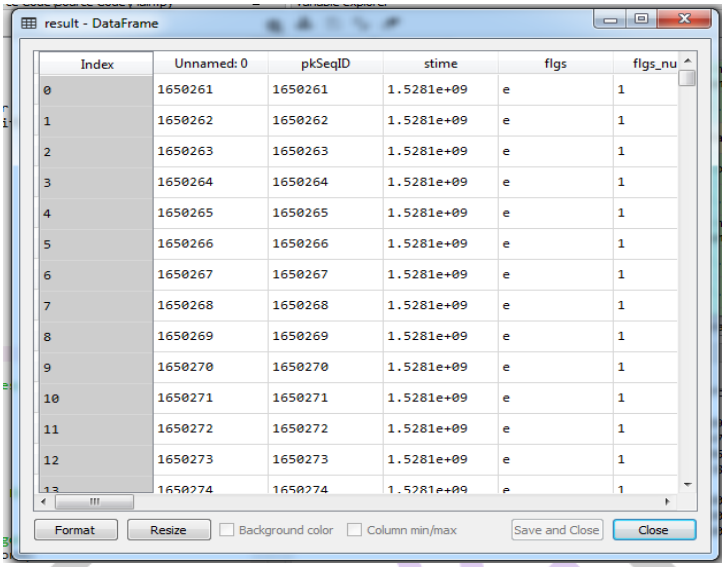
Although several types of datasets are available in network intrusion detection. They have several challenges, like lack of reliable labels, redundancy of network traffic, low attack diversity, and missing ground truth. For instance, NSL-KDD and KDD Cup99 datasets are most popularly applied, but they are out dated, and they are not reflect the current normal and the attack scenarios. The usage of the DEFCON-8 dataset is limited because of low number of benign traffic models. These attack scenarios in UNIBS dataset are limited to DOS, Bot-Net dataset is the most related dataset which is publicly available for network devices botnet attack detection in the systems. To realise this dataset, an network test bed was set up to generate benign and malicious network traffic using heterogeneous communication protocols like User Datagram Protocol (UDP), Reverse Address Resolution Protocol (RARP), Transmission Control Protocol (TCP), Internet Control Message Protocol (ICMP), Address Resolution Protocol (ARP), Internet Protocol version6 ICMP (IPv6-ICMP) and Internet Group Management Protocol (IGMP).
The test bed setup comprised a variety of IOT devices, including a weather station, smart fridge, remotely-activated garage door and smart thermostat. Also, millions of botnet attack traffic samples were included in Bot-Net. These attack traffic samples can be categorized into four IOT botnet scenarios, namely: DDOS, DOS and information theft. To ensure a fair comparison, feature dimensionality reduction methods that do not include benign network traffic traces and all the four botnet attack scenarios in the Bot-IOT dataset were not included in this paper. For instance, did not consider the DOS attack scenario. Also, the performance of the method in detecting benign network traffic was not reported. In a similar work, the authors did not evaluate the procedure of the proposed method. In another work did not evaluate the procedure of the proposed method with the network traffic data in the BOTNET dataset.
In summary, the state-of-the-art methods in the related work focused on the selection of specific features from available network traffic information available in the Bot-IOT dataset. However, this approach may likely affect the efficiency of botnet attack detection in IOT networks because the classifiers will not have access to some relevant network information during training, validation, and testing. Consequently, the feature selection approach may lead to low botnet attack detection accuracy and a high false alarm rate in IOT networks. On the other hand, LAE decreases the dimensionality of big IOT network traffic data and produces a low-dimensional latent space feature representation at the hidden layer without losing useful intrinsic network information.

SYSTEM ARCHITECTURE:



DATA SELECTION:

Data selection is the process of identifying the malicious traffic and the input data was collected from the main data set, In our progress the BOT-IOT dataset is used. In the dataset includes the normal traffic flows as well as several cyber-attacks traffic flows of botnet attacks. To track the accurate traffic and to train the dataset effectively, the test bed is used for the development of this dataset.



Index	Unnamed: 0	pkSeqID	stime	flgs	flgs_nu
0	1650261	1650261	1.5281e+09	e	1
1	1650262	1650262	1.5281e+09	e	1
2	1650263	1650263	1.5281e+09	e	1
3	1650264	1650264	1.5281e+09	e	1
4	1650265	1650265	1.5281e+09	e	1
5	1650266	1650266	1.5281e+09	e	1
6	1650267	1650267	1.5281e+09	e	1
7	1650268	1650268	1.5281e+09	e	1
8	1650269	1650269	1.5281e+09	e	1
9	1650270	1650270	1.5281e+09	e	1
10	1650271	1650271	1.5281e+09	e	1
11	1650272	1650272	1.5281e+09	e	1
12	1650273	1650273	1.5281e+09	e	1
13	1650274	1650274	1.5281e+09	e	1

DATA PRE-PROCESSING:

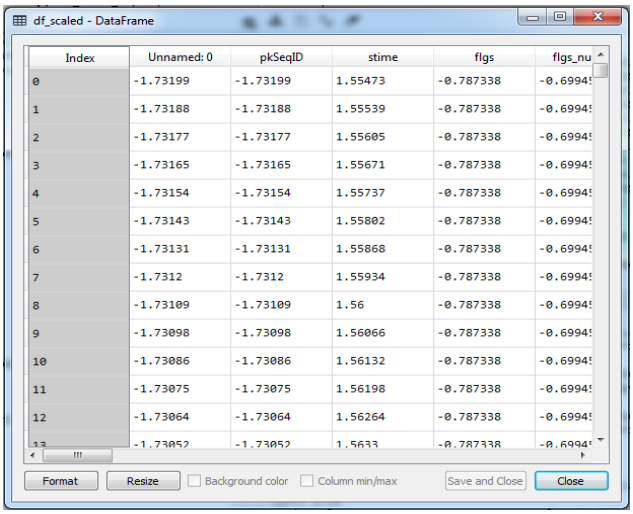
It is the process of deleting or removing the unwanted data from the main dataset. The main purpose is to transform the dataset into a suitable structural format. Using this method to remove the duplicated data it helps to clean the dataset and will get the more space. The Encoding categorical data is a variables with a finite set of values. The machine learning algorithms needs a numerical 0 and 1 input and output variables.

```
-----------------Checking Missing Values-------------------------

Unnamed: 0      0
pkSeqID         0
stime           0
flgs            0
flgs_number     0
proto           0
proto_number    0
saddr           0
sport           0
daddr           0
dtype: int64
```

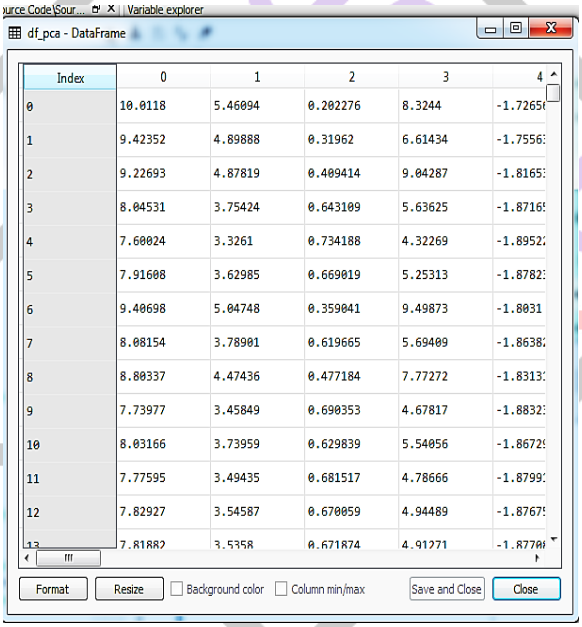
DATA NORMALIZATION:

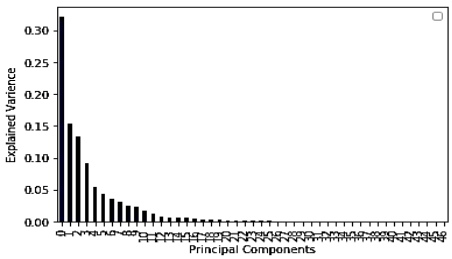
It contains normalize the data by using the normalize function. It takes an array as an input value and to normalize the values like 0 and 1. Standardization means to input value will be separated equally and one shift of distribution have a zero and a standard deviation of one.



DIMENSIONALITY REDUCTION:

The Number of input like variables, or columns present in a main dataset is called as dimensionality, and to reduce these features is known as a dimensionality reduction





AUTOENCODER:

Auto Encoder can be used to encode the raw data. It encodes the encoder and decoder of sub models. The Encoder compresses the input data and decoder recreates the input from the compressed encoder. Once the training is finished the encoder model will be saved and decoder will be discarded.

```
Train on 381 samples, validate on 96 samples
Epoch 1/10
381/381 [==============================] - 3s 7ms/sample - loss: 0.2540 - val_loss: 0.2822
Epoch 2/10
381/381 [==============================] - 1s 2ms/sample - loss: 0.2534 - val_loss: 0.2815
Epoch 3/10
381/381 [==============================] - 0s 210us/sample - loss: 0.2527 - val_loss: 0.2808
Epoch 4/10
381/381 [==============================] - 0s 173us/sample - loss: 0.2521 - val_loss: 0.2800
Epoch 5/10
381/381 [==============================] - 0s 123us/sample - loss: 0.2514 - val_loss: 0.2792
Epoch 6/10
381/381 [==============================] - 0s 126us/sample - loss: 0.2507 - val_loss: 0.2785
Epoch 7/10
381/381 [==============================] - 0s 121us/sample - loss: 0.2500 - val_loss: 0.2777
Epoch 8/10
381/381 [==============================] - 0s 134us/sample - loss: 0.2493 - val_loss: 0.2769
Epoch 9/10
381/381 [==============================] - 0s 123us/sample - loss: 0.2485 - val_loss: 0.2761
```

CLASSIFICATION:

Long Short Term Memory is an artificial neural network. It is standard for Feed Forward Neural Networks. It process entire sequences of data not only in single data points. These networks are well formed to classify, processing and making prediction it's depend on the time and data. Convolution Neural **Network** is a deep learning algorithm which can compare the input until it will find out the best output. It follows as a hierarchical model.

```
--------------------------------------------------
Long Short term Memory
--------------------------------------------------
PERFORMANCE METRICS

1.Confusion Matrix [[  390    0    0    0]
 [  186    0    0    0]
 [11607    0    0    0]
 [   57    0    0    0]]

2.Accuracy 67.70833333333334 %

3.Precision for LSTM 100.0 %

4.Recall for LSTM 67.70833333333334 %
```

```
--------------------------------------------------
Convolutional Neural Network
--------------------------------------------------
PERFORMANCE METRICS

1.Confusion Matrix [[  390   148 10662    57]
 [    0    38   945     0]
 [    0     0     0     0]
 [    0     0     0     0]]

--------------------------------------------------
2.Accuracy 100.0 %

3.Precision  72.4907063197026 %

4.Recall 100.0 %
```

RESULT GENERATION:

The Expected Result will be generated based upon the prediction and classification and two important things to increase the result like accuracy and precision.

Accuracy: It refers to the ability of the classifier to predict the class correctly and also the accurate of the predictor. Ac= (TruePositive+TrueNegative)/ (TruePositive+TrueNegative+FalsePositive+FalseNegative)

Precision: It refers to the Number of True positive divided by the total number of true positive plus the total number of false positives. P=True Positive/ (TruePositive+FalsePositive)

Recall: Recall is the number of correct output divided by the number of outputs that should have been returned. R=True Positive/ (TruePositive+FalseNegative)

CONCLUSION:

This system was proposed for efficient botnet detection in networks using deep learning algorithms such as LSTM and CNN. The effectiveness of this method was validated by performing extensive experiments with the most relevant publicly available dataset in binary and multi-class classification scenarios. By using this CNN the result will get more accurate and precision also comparing to the LSTM.

REFERENCES:

1. A. O. Akmandor, Y. Hongxu, and N. K. Jha, "Smart, secure, yet energyefficient, internet-of-things sensors," IEEE Transactions on Multi- Scale Computing Systems, vol. 4, no. 4, pp. 914–930, 2018.
2. D. E. Denning, "An intrusion-detection model," IEEE Transactions onsoftware engineering, no. 2, pp. 222–232, 1987.
3. J. Qiu, L. Du, D. Zhang, S. Su, and Z. Tian, "Nei-tte: Intelligent traffic time estimation based on fine-grained time derivation of road segments for smart city," IEEE Transactions on Industrial Informatics, 2019.
4. J. Qiu, Z. Tian, C. Du, Q. Zuo, S. Su, and B. Fang, "A survey on access control in the age of internet of things," IEEE Internet of Things Journal, 2020
5. X.-G. Luo, H.-B. Zhang, Z.-L. Zhang, Y. Yu, and K. Li, "A new frameworkof intelligent public transportation system based on the internet of things," IEEE Access, vol. 7, pp. 55 290–55 304, 2019.
6. Z. Tian, X. Gao, S. Su, and J. Qiu, "Vcash: A novel reputation framework for identifying denial of traffic service in internet of connected vehicles," IEEE Internet of Things Journal, vol. 7, no. 5, pp. 3901– 3909, May 2020.