Deep learning-Based Real-time malicious network traffic detection system for Cyber-Physical Systems

Pranjal Mestry[1], Ameya Rathi[1], Dr Gaurav Bansod[1]

[1]SCTR Pune Institute of Computer Technology, Pune, IN 411043

Corresponding Author: Dr Gaurav Bansod (e-mail: gvbansod@pict.edu)

ABSTRACT*:* Characteristics of IoT, such as large quantities and simple functions, have made it easy for IoT devices or servers to be attacked and converted into DDoS(Distributed Denial of service) attacks. The purpose of this paper is to provide an overview of security in the IoT sector and to discuss the classification of different attacks. In this paper, we have established a way to better analyze DDoS attack traffic from selected IoT devices under a local network route in real time using a hybrid deep learning model consisting of a convolutional neural network and a long term short memory neural network. Specifically, we used the CICFlowMeter tool for collecting features from real data traffic which we intercepted on the node and reduced the feature vector using feature selection algorithms.The feature selection algorithms that we used were random forest and p-value selection algorithm. The features extracted from feature selection algorithms were tested on the deep learning model and the features that gave best performance of the model were selected. We then integrate the CICFlowMeter tool with our designed model to extract the features from real traffic and classify the nature of the traffic into normal or malicious.The results show that applying random forest feature selection algorithm leads to increase in accuracy and decrease in latency when classifying the nature of network traffic .

Key Words: DDoS, IoT Security, IoT Attacks, Deep Learning,CNN

1 INTRODUCTION

The Emerging trends in embedded technology and the Internet have made things around us more connected. The Internet of Things (IoT) is an emerging communication paradigm that aims to connect various types of objects to the Internet, harvest sensory data, remotely control equipment and devices, monitor locations, vehicles, and buildings, and so on. The number and diversity of IoT devices have grown exponentially over the years. This will increase security as a major concern.

In order for commercial IoT devices to withstand cyber attacks, security should be considered from the design phase of new products. However, the great diversity of IoT devices hinders the development of well-established security mechanisms with IoT design. The challenge is also compounded by the complex limits in terms of power, connectivity, pricing, and storage capacity of many IoT devices[1]. Such restrictions severely limit the availability of standard security measures used on traditional Internet-connected devices and require new, yet unparalleled solutions.

In October 2016 a major DDoS attack on IoT devices was launched. It was done by building a great botnet called Mirai on Brian Krebs' famous security blog, krebsonsecurity.com. 100000 IoT devices (security cameras) were exploited by Mirai to launch a massive attack on DNS(Domain Name System) with excessive bandwidth of 1Tbps. Netflix, Amazon, Twitter and Github were some popular websites that were crippled for hours. IoT attacks can occur as devices on the IoT network are easy to access. Once they are in danger, hackers can control and perform dangerous tasks and attack other devices near the endangered area. The problem of protecting IoT devices is mainly due to their nature of resource limitations because solutions to reduce attacks and privacy protections used on traditional networks cannot be easily integrated into IoT networks.

Key contribution:
In this paper, we have proposed an efficient method to detect IoT attacks on networks. We validate our proposal by running testbed experiments. Our main contributions are as follows:

1) We have presented a deep learning model using CNN+LSTM (Convolutional Neural Network + Long Short Term Memory) to detect which attack took place in the network.
2) We have extracted 21 important features from data packets for our model to get trained on these features.
3) We have used CICFlowMeter for Ethernet traffic Bi-flow generator and analyzer for generating flows and extracting unique features from them. CICFlowMeter has been used in many Cybersecurity datasets.
4) Conducted experiments on our model in real-time data and achieved an accuracy of more than 99%.

Paper Structure:

The remainder of this paper is structured as follows. In section I, we gave information about IoT in Cybersecurity and some past attacks that took place in the world. In Section II we gave background information about past work related to attacks and the detection of IoT attacks. In section III we introduced our system design which includes data preprocessing, features Selection , and model architecture. Results are shown in section IV and finally, conclude in section V.

2 REVIEW OF IOT ATTACK AND COUNTERMEASURES

The network layer is the part of the Internet of Things that carries out various tasks such as transferring and storing data. It is typically responsible for storing and accessing data.In order to find the proper protocols for the Internet of Things, Yavuz[1] proposed a DL-model (DLP) to detect anomalous behavior in the networks. He used the Cooja IoT simulator with upto 1000 nodes within 16 networks to detect a number of version attacks, a decrease in level attack and a flood of hello attacks. DNN performed well in detecting three attacks. Their model achieved good accuracy (99%) and F1 score close to 99%

Bostani H. and Mansour Sheikhan[6] proposed hybrid architecture which consists of specification based and anomaly based intrusion detection system for detecting iot based routing attacks.Specification based module which are located in router nodes analyze the behaviour of host nodes and send the result to the root node. Anomaly detection module is deployed at the root node which analyzes the incoming data packets to and predicts the nature of the traffic using an unsupervised optimum path forest algorithm. They achieved 76.19 % true positive rate and 5.92 false positive rate when selective forwarding and sinkhole attacks were launched together.

Saied, Overill and Radzik[9] detected DDoS attacks based on specific characteristics patterns that separate traffic using ANN. The purpose of this paper is to detect known and unknown DDoS attacks in real time. The main objective of the Distributed Denial of Service (DDoS) attack is to integrate multiple Internet systems and create botnets. An attacker or Trojans controls infected systems remotely and launches packets into the system.

Yingxu lai, Jhingwen zhang and Zenghui Liu [11] proposed deep learning based convolutional neural network detection models for industrial traffic anomaly detection and attack classification. CNN based Intrusion detection system is capable of automatically extracting critical features thus giving high accuracy. The model was tested on real time attack data traffic using the SCADA system. This model performed better than classical machine learning algorithms like SVM, decision tree. However, CNN can only analyze one data packet at a time. Generally, when a large number of data packets are sent simultaneously in a small span of time it constitutes malicious traffic. CNN can only analyze spatial information from data traffic but fails to analyze sequential or time series information from data traffic.

Pramita Sree Muhuri, Prosenjit Chatterjee, Xiaohong Yuan ,Kaushik Roy and Albert Esterline [2] proposed deep learning based long term short memory recurrent neural network intrusion detection systems to classify network attacks. The model was trained on NSL-KDD dataset for binary classification (malicious or non-malicious network traffic) and multiclass classification to classify the nature of attack in categories such as normal, Dos, probe, U2R, R2L respectively. Genetic algorithm (GA) was used for optimal feature selection. The results obtained from the model after applying GA for optimal feature selection were compared withSVM and random forest. The results showed an increase in accuracy for multiclass classification and binary classification when compared with models based on SVM and Random forest. The limitation of using RNN is that it can extract sequential information, i.e the information in the temporal domain from streaming network traffic but fails to analyze the information from a single data packet of network traffic, i.e the information in spatial domain.

This drawback can be overcomed by introducing a layer which analyzes time series information of data traffic along with CNN. In [3] Sun, Pengfei & Liu, Pengju & Li, Qi & Liu, Chenxi & Lu, Xiangling & Hao, Ruochen & Chen, Jinpeng proposed a hybrid DL-IDS which consisted of convolutional neural network and long term short memory (RNN) to classify the attacks based on extracted features from spatial as well as temporal domain .The model was trained and tested on CICIDS2017 network traffic data set which simulates real network traffic.They achieved an overall accuracy of 98.67%.

3 System Design

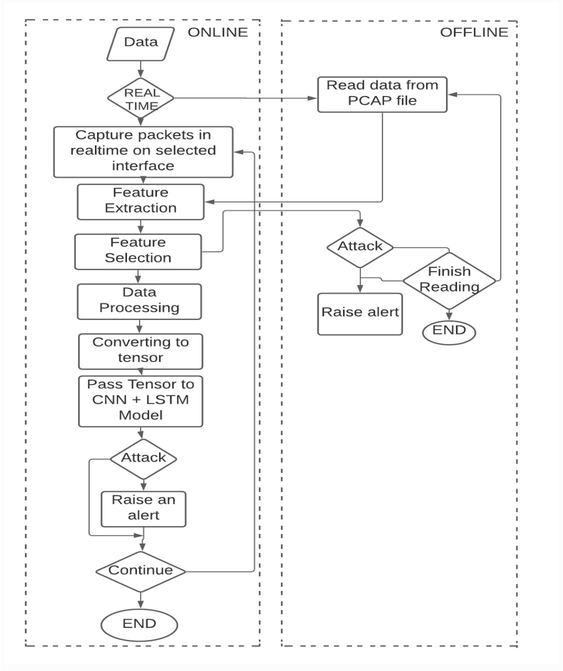

Fig 1: System Design

Above figure represents the block diagram of Intrusion detection system which explains the flow from capturing the packet, extracting the features, preprocessing them and detect the nature of data traffic

A . Dos Attack

A DDoS attack can be broken down into three steps:-

1. Attack preparation
2. Attack Accumulation
3. Attack Saturation.

Attackers first collect all the information from the victim's system such as open ports, IP addresses, and then prepare for attacks. Once the information about the victim's system is extracted the attackers start preparation of dos attacks which builds up in time and enters in the second stage i.e attack accumulation. At this stage, the victim's system experiences some latency as most of the connection ports and network resources are consumed by the botnets.

At the accumulation stage, the system is still able to communicate not only with benign IP addresses but also with the attacker's IP address. It is an absolute necessity that the attack has been detected at the accumulation stage as at the saturation stage the attack completely takes over the victim's system ,rendering it unable to respond to benign IP addresses. This can pose a serious threat in industries where real-time exchange of information between various IoT network devices is crucial.

In order to prevent the system from the attack when it enters the saturation stage, latency for detection of the attack has to be as low as possible. A popular approach is to use flow information over packet information as flow information is the summary of the packets that are exchanged between two IP addresses in a session.

B. Extracting features from data packets in real-time.

We used the Cic Flowmeter [13] tool for extracting the features from bidirectional flows in real-time. The CICFlowMeter generates bidirectional flows, where the first packet regulates the source to destination (Forward) and destination to source (backward) directions. The CICFlowmeter calculates more than 80 statistical network traffic features such as length of forward and backward packets, Number of packets, IAT (time between two packets sent in either direction) etc. and calculates them separately in backward and forward directions. The output of CICFlowMeter is .csv file which is collection of 80 statistical network traffic features for every flow generated along with source ip, destination ip, source port and destination port

UDP packet flow sessions are terminated by flow duration timeout and TCP flow sessions are terminated by connection breakdown when it intercepts FIN packets . Whenever the flow session is terminated, the features are calculated and the feature vector is generated. When flow vectors equal to timesteps are generated they are grouped in one vector of dimensions (timesteps, number of features) and then preprocessed(section 3.C) before being fed to the proposed model

C. Data Preprocessing

The CIC IDS 2017 data set is available in a CSV file. All the features in the data set are in numerical format except for the labels. The nan and infinity values in the data sets are removed and are replaced by the mean of that respective feature.

Data Normalization - Data in the data set varied from very large to very small values thus the data was not uniform and evenly spread. Data normalization is carried out to distribute the data uniformly, ensuring faster training of the model and accurately reaching a local minimum. Data normalization is carried out after filtering of Nan and infinity values from the CICIDS 2017 data set.

r = max_i - min_i + eps…..(1)
data = (data - mean_i) / r….(2)

In (1) max_i is the maximum value of that respective feature and min_i is the minimum value of the feature mean_i is the mean value of the feature in the data

Feature Selection - The features which are selected using random forest classifiers yield better results as compared to other feature selection algorithms thus those features are selected from the data set. In total 21 such features are selected from the CICIDS 2017 dataset.

Converting data in (no_of samples, time steps, no of features) format.-
Since the 1d convolutional layer requires input data in a three-dimensional format the two-dimensional data is thus converted in the number of samples, timesteps, number of features format where the number of samples is equal to (number of records in the dataset) / (number of timesteps).

Label Encoding - The label field in the data set has a string data type and the model processes only integer and float data types hence the labels are converted to integer classes using a one-hot label encoder.

Splitting the data set into training, validation, and testing set:-
Training and testing data are split into 80 % training data, 20% testing data.Validation dataset consists of 25% of the data from training dataset, which gives the estimation of the model's performance while tuning models' hyperparameters.

D. Feature Selection

1. Feature selection using random forest

Random forests are one the most popular machine learning algorithms for feature selection. They offer excellent predictive performance, extreme skipping, and easy interpretation. This is because it is straightforward to determine the importance of each variable on the tree decision.

Random forests consist of 400 to 1200 decision trees and each tree is built over a random extraction of the features. Since, for every tree in a random forest the subset of features are randomly selected ,the trees are de-correlated and thus they are very unlikely to overfit. Each node divides the dataset into two subsets, hence each child node now depends on the features of the respective subset and hosts observations which are different from another subset.

Each tree in a random forest has its own set of properties that determine the importance of a feature. The feature importance can be computed by estimating the amount of impurity that it can reduce. The measure of impurity that was selected for classification is Gini impurity.

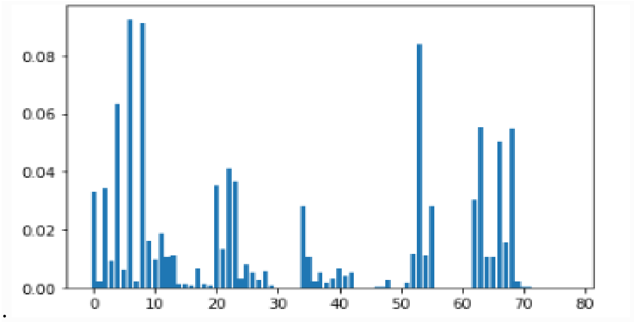

Fig 2: Feature importance visualization

Feature importance visualization graph in which the bar chart(Fig 2)represents features with corresponding importance values where the y axis indicates the value of feature importance and the x-axis indicates the features.
Following are the important features that we extracted from the CICIDS2017 dataset using Feature Importance Random forest algorithm:

Sr No:	Feature Name
1	Destination Port
2	Total Fwd Packets
3	Total Backward Packets
4	Total Length of Fwd Packets
5	Fwd Packet Length Max
6	Fwd Packet Length Mean
7	Fwd Packet Length Std
8	Bwd Packet Length Max
9	Bwd Packet Length Min
10	Bwd Packet Length Std
11	Fwd IAT Total
12	Fwd IAT Mean
13	Fwd IAT Std
14	Fwd IAT Max
15	Fwd Header Length
16	Avg Fwd Segment Size
17	Avg Bwd Segment Size
18	Subflow Fwd Packets
19	Subflow Fwd Bytes
20	Init_Win_bytes_forward
21	act_data_pkt_fwd

Table 1: List of Selected Features

2. Feature selection using correlation and p-value-

The term correlation refers to how closely two variables are linearly dependent on each other. Features that have high correlation are highly linearly dependent and thus they affect the dependent variable similarly. Therefore we can drop one of these two features. The coefficient of selection we selected was 0.8 which indicates high collinearity and thus one of the features can be omitted.

P-value is a probability value that tells us that there's a chance that an observation will be true. It helps us to select or reject a hypothesis.. The removal of different features from a dataset affects the p-value of the dataset differently.we can remove each feature and measure the p-value in each case. Thus p-value helps to decide whether to keep or discard the feature.

The following figure 3 visualizes the correlation heat map where light squares indicate high linearity between two corresponding features. For example, the color of the cell corresponding to feature from row 40 and column 76 is dark which represent that correlation between those feature is low which is below threshold value (0.8) as can be seen from below color scale and similarly the color of the cell corresponding to the feature from row 4 and column 64 which is light and is above our threshold so on of this feature can be omitted.

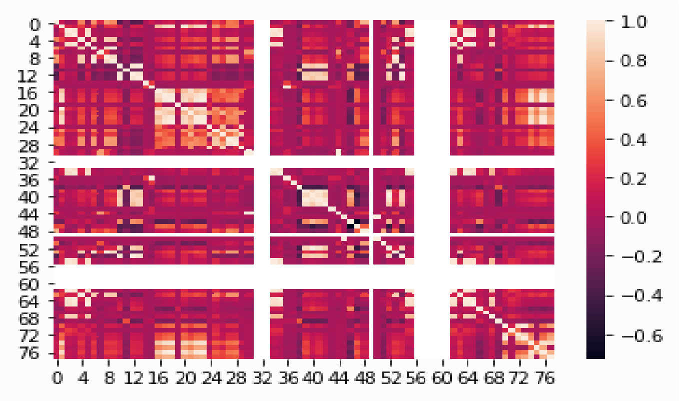

Fig 3: Feature Correlation heat map which represents one to one correlation of features .

E. DL-IDS Architecture

1. Convolutional Neural Network + Long Term Short Memory :

This section is about the traffic classifier we built using a convolutional neural network and recurrent neural network which classifies the nature of the traffic in both spatial and temporal domains.The architecture designed is light weight which is capable of functioning on low resources and low computing power device like raspberry pi. The architecture of the model is shown in the figure. The CNN section is composed of two 1-dimensional convolutional layers which are followed by a 1d max-pooling layer. The CNN section outputs a vector of high dimension to LSTM, hence Max pooling is used to reduce the dimension and extract significant information from the high dimensional data and pack that into low dimensional data, thus reducing the complexities. The output of the max-pooling layer is fed to the Long term short memory layer. The output of LSTM is then flattened and converted to 1d data format and inputted to the dense layer which represents the probability of which class the traffic belongs to.

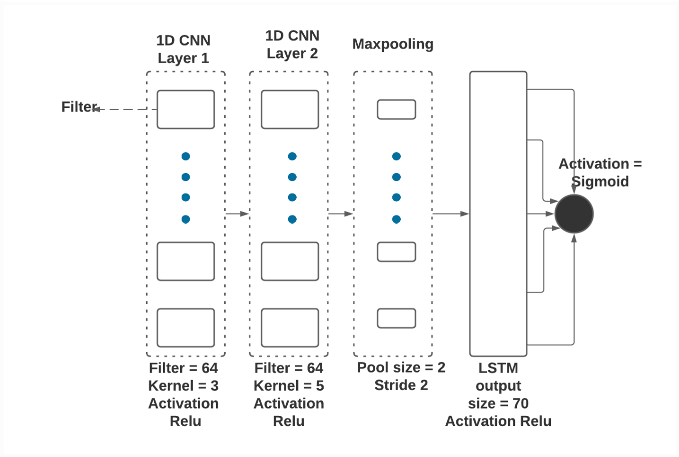
1D CNN Layer 1
1D CNN Layer 2
Maxpooling
Filter
Activation = Sigmoid
Filter = 64 Kernel = 3 Activation Relu
Filter = 64 Kernel = 5 Activation Relu
Pool size = 2 Stride 2
LSTM output size = 70 Activation Relu

Fig 4: CNN + LSTM model architecture

Above figure gives a detailed description of the number of filters, kernels and activation functions used at each layer of the model.

2. Convolutional Neural Network:

The incoming data traffic to the device is the time series data, thus the current features of the traffic flow depend on the previous traffic flows as well. Thus 1 dimensional CNN performs the analysis of the traffic flows in spatial as well as temporal domains. The 1D CNN model is capable of learning the internal representation of the time-series data to achieve comparable performance on the data set. Since 1D CNNs are less computationally intensive they are used for mobile and edge devices.

1D CNN requires input in a three-dimensional format resembling (number of samples, Time steps, Number of features). The individual flow feature vectors are combined together to form a matrix of (4,21) resembling the timesteps which are finally reshaped to a three-dimensional array.

The CNN section comprises two consecutive convolutional layers to effectively learn the traffic features.

- The first layer of CNN has 64 filters and a kernel size equal to 3. The small kernel size establishes the relationship between the close localized features of the input features vector such as IPs and destination ports src ports etc.
- The second layer of the 1d CNN has 64 filters and kernel size equal to 5 which helps to establish the relationship between the data points that are away in a spatial domain such as information in payload and information about flow duration
- The output is then passed to the max-pooling layer thus reducing the dimensions and extracting only significant traffic information from the output vector of the CNN layer.

Input dimensions to the next layer are given by the formula.

$$N_{out}+1 = ((N_{in} + 2p - f)/s) + 1 \quad \ldots.(3) ,$$

where,

N_{in} denotes the number of input features, N_{out} denotes the number of output features , P convolution padding size, S is stride and f is kernel size.

The convolution layer contains an activation function which performs non-linear transformation of the input vector. In formula (4) $f(Z^{l}_{i,j,k})$ is an activation function which takes Z vector and transforms it to output vector A

$$A^{l}_{i,j,k} = f(Z^{l}_{i,j,k}) \quad .....(4)$$

We used ReLU activation for both the 1d convolutional layers.
Backpropagation is used to adjust the weights of the filters at every epoch. α is the learning rate and δ is gradient of vector A with respect to the weights.

$$w^{l} = w^{l} - \alpha \sum \delta^{l} * A^{l-1} \, ...(5)$$

Binary cross entropy algorithm is used as a loss function to optimize training time and accuracy of gradient descent. Adams optimization algorithm is used to finely tune learning rate.

.

Once the convolution operation from two 1d CNN layers and max-pooling layer is finished, the entire traffic flow data is extracted and presented in a feature block. The data is then fed to the RNN.

The model summary of the 1d CNN section is as follows, which represents the input and output dimensions of the feature array to each layer.

Layer(Type)	OutputType	Parameter
conv1d_9(Conv1D)	(None,2,64)	4096
conv1d_10(Conv1D)	(None, 2,64)	4196
Max_pooling1d_3 (MaxPooling1)	(None,1,64)	0
lstm_2 (LSTM)	(None,70)	37800
dropout_2 (Dropout)	(None, 70)	0
dense_2 (Dense)	(None, 1)	71

Table 2: Model Summary of 1D CNN section

3. Long Short Term Memory (LSTM):

A recurrent neural network is commonly used to process time-series data. For instance, it can handle different types of data points, such as images and speech.
We used LSTM neural network architecture to process time series information of network traffic

A commonly used LSTM unit is a cell, which is composed of an output gate, a memory gate, and a forget gate.

- In this project's application, the LSTM takes data of a single connection as a group and judges the characteristics of the data packets of that particular group and relationships to decide the category of nature traffic.
- The LSTM section comprises an LSTM layer and a dense layer at the output.
- In step one the LSTM layer decides what information the cell will discard. This is achieved through a forgetting gate. This gate outputs the value between 0 and 1 where 0 means the information is discarded and 1 means the information is retained. The equation of forget gate is as follows where h_{t-1} and x_t are inputs ,W_t denotes weight and b_f is bias :
  $f_t = \sigma(W_t *[h_{t-1}, x_t] + b_f)$ …(6)
- As the next step, a sigmoid layer decides which information in the cell state needs updating (formula 7). Then, a tanh layer generates a vector as an option for updating (formula 8).A cell's state will be updated once both parts have been combined (formula 9)
  $i_t = \sigma(W_i*[h_{t-1},x_i] + b_i)$ …(7).
  $C^{`\sim}_{t} = tanh(W_c*[h_{t-1}, x_i] + b_i)$ …(8).
  $C_t = f_t*C_{t-1} + i_t*C^{\sim}_{t}$ …(9).

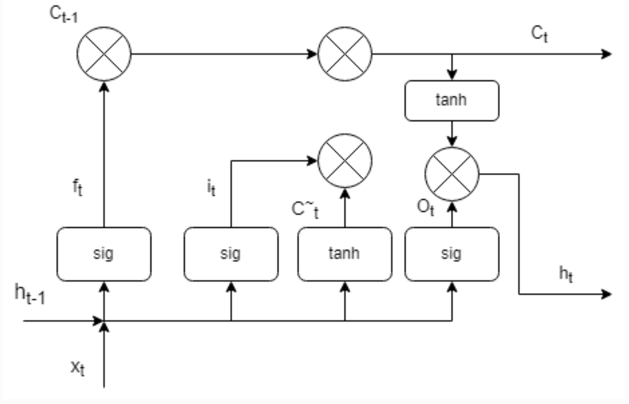

fig 4. Output Gate Operation

- .A sigmoid layer decides which parts of the cell state are exported. Tanh function outputs a value between 1 and -1 and then it is multiplied by the output of the sigmoid gate. [19].
  $O_t = \sigma(W_o*[h_{t-1},x_t] + ,b_o)$ …(10).
  $h_t = O_t*tanh(C_t)$ …(11).

F. INTEGRATION OF MODEL WITH CICFlowMeter

As mentioned in (A) , the CICFlowMeter generates flows, extracts features from the flow information of data traffic, and generates the .csv file of the features. Our goal is to extract features and feed those to the deep learning model without generation of CSV file So cic flowmeter was configured to generate only 21 features that were selected using random forest feature selection algorithm and the resources that were used for generating the rest of the features were eliminated to reduce the latency. The features are preprocessed exactly as mentioned in (C) . After preprocessing, the feature vectors are fed directly to a model where it detects the status of the network traffic. Two different threads are used for flow generation, feature collection, and attack detection so that the performance of the system is not hampered and flow information is not skipped. When an attack is detected the system raises an alarm and continues analyzing the flow information and monitoring the status of traffic until the system is interrupted.

4 RESULT

The validation set is only used to examine the model performance while the training set is used to train the model. If the difference between accuracy and validation accuracy is greater that means for that data the model is overfitting the data points. At this point it is necessary to add regularization to the layers of the model to reduce overfitting and generalize the model.

If the version's prediction is best, the loss is zero; in any other case, the loss is extra. Loss is the penalty for a bad prediction. Loss is a number indicating how bad the model's prediction was on a single instance.

In this paper, we adopted five parameters for validating the performance of the models which are accuracy, precision,f1 score and recall.

A. Results for the model with 21 features:

In the figure below the graph represents the learning curve of training and validation curves where the X axis indicates the epoch of training and the Y axis indicates the accuracy.

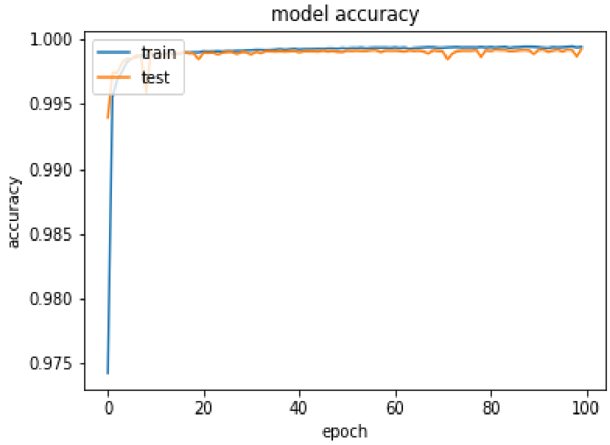

Fig 5: Accuracy Curve for a model trained with features selected using random classifier

In the figure below the graph represents the loss curves of the training and validation data set where X axis indicates the epoch in training and Y axis indicates the loss. As expected, the loss of the model decreases with respect to increase in epochs.

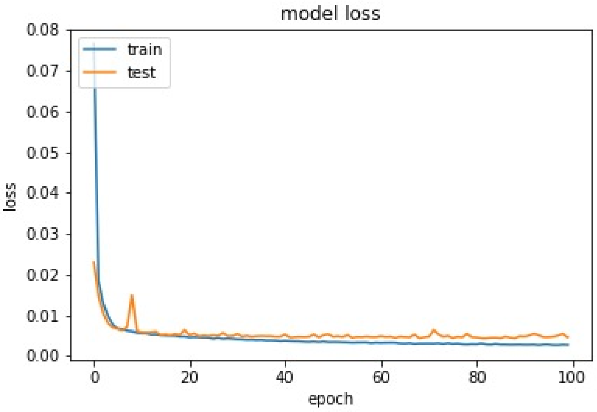

Fig 6: Loss curve for a model trained with features selected using random classifier

The figure below represents a confusion matrix obtained on a testing data set which gives an idea of how well the model is able to classify true positives, true negatives, false positives and false negatives.

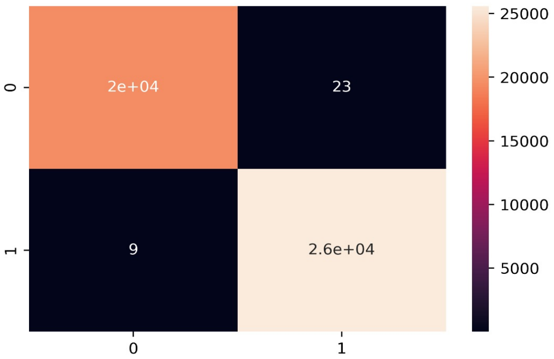

Fig 7: Confusion matrix

- The conclusion from 21 features :

The deviation between the train and validation curves in the accuracy plot is very negligible which indicates that the model has a low bias value. The deviation between the train and validation curves in the loss curve is also low which indicates that the model has a low variance value. As the bias and variance values are low the model is an accurate model.
From the confusion matrix, it is evident that the model is predicting very large true positive and true negative values and very small false positive and false negative values. Thus the model's prediction accuracy and precision is high.

B. Results for a model with 43 features

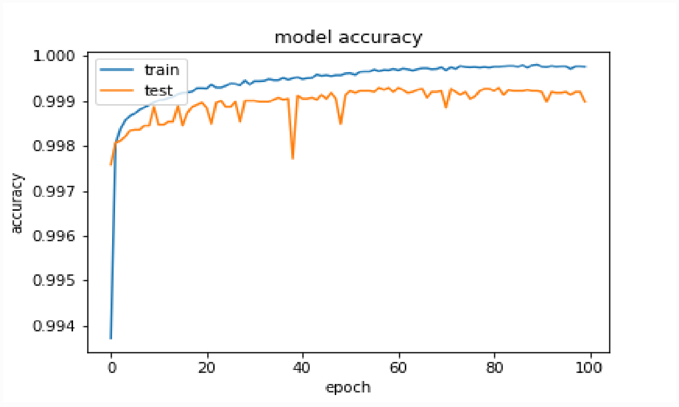

Fig 8: Accuracy curve for a model trained with the features selected using correlation and p-value algorithm

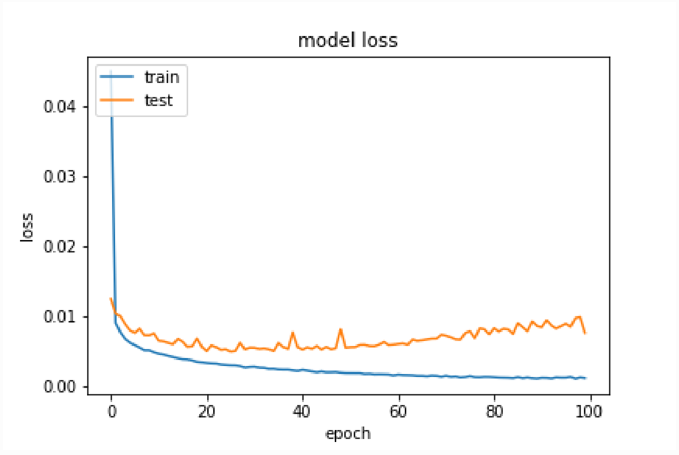

Fig 9: Loss curve for a model trained with features selected using correlation and p-value algorithm

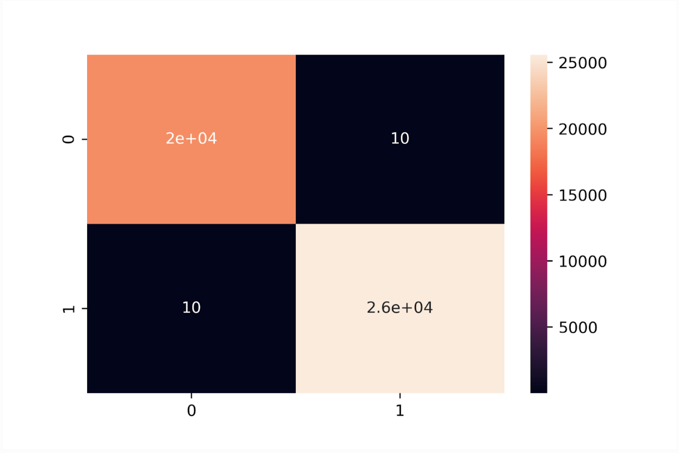

Fig 10: Confusion Matrix

- The conclusion from 43 features :

The deviation between the train and validation curves in the accuracy plot is slightly more but it is still less which indicates that the bias value of the model is still on the lower side. The deviation between the train and validation curves in the loss curve is also high which indicates that the model has a high variance value. As the bias is on the lower side and variance value is on the higher side the model captures noise and slightly overfits the data which is not an ideal case.

From the confusion matrix, it is evident that the model is predicting very large true positive and true negative values and very small false positive and false negative values but as seen from the graphs the model is slightly overfitting the data. Thus the model is not very reliable.

C. Results for the model with 78 features:

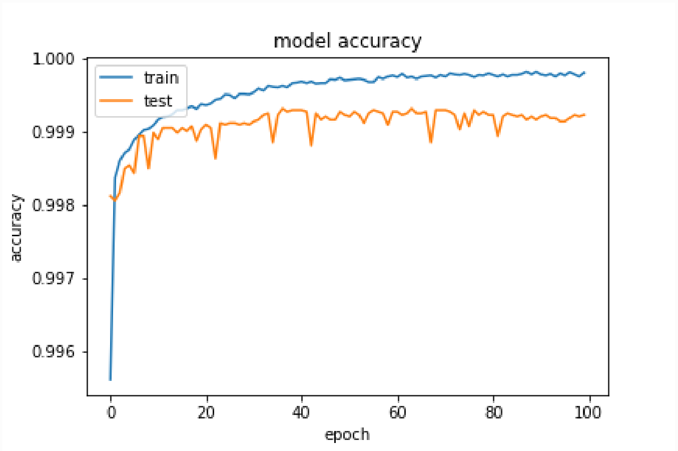

Fig 11: Accuracy curves for the model with all features present in the data set

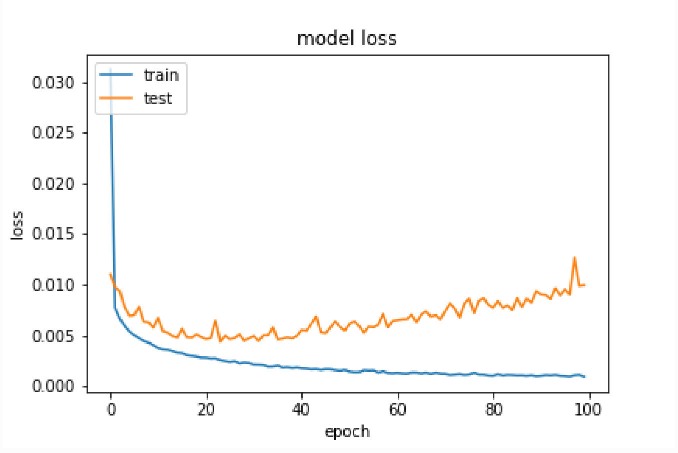

Fig 12: Loss curve for the model with all the features present in the data

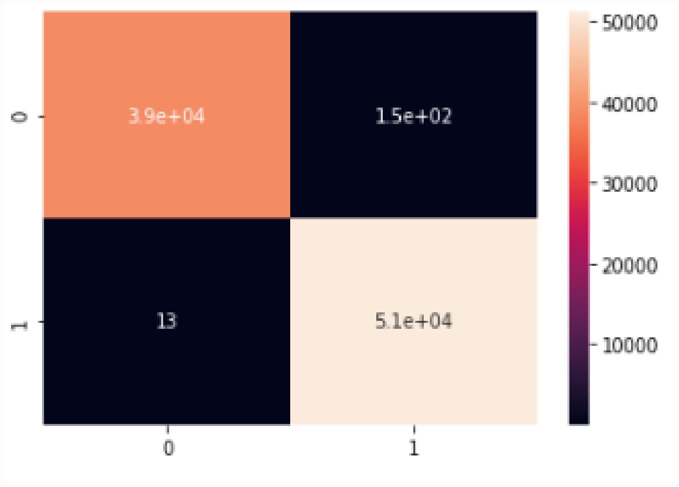

Fig 13: Confusion Matrix

- The conclusion from 78 features :

The deviation between the train and validation curves in the accuracy plot is slightly more which indicates that the bias value is on the higher side which leads to underfitting. The deviation between the train and validation curves in the loss curve is also more which indicates that the model has a higher variance value which indicates overfitting. As the bias and variance values are on the higher side the model is not an accurate model.

From the confusion matrix, it is evident that the model is predicting a very large number of false positives which indicates that the model is not reliable.

Factors	Model 1	Model 2	Model 3
Feature Selection Algorithm	Random Forest	Correlation And P-Value	NA
Number of Significant Features	21	43	78
Accuracy	0.99	0.99	0.97
Recall	1.0	0.96	0.999
F1 Score	0.999	1.0	1.0
Precision	1.0	0.982	0.91
Processing Time	1.187 s	1.962 s	2.139 s

Table 3: Details of Model

5 CONCLUSION

In this paper, we have proposed an intrusion detection system based on deep learning that brings together the spatial domain analysis of convolutional neural networks and temporal domain analysis of long term short memory networks. To analyze network traffic Our model extracts features from network traffic data flow to analyze the network traffic. The features from a single data packet are extracted by CNN whereas temporal information from the data stream is extracted by the LSTM network.

We used the CICIDS2017 dataset to train the model on various types of Ddos attack. To reduce the dimensionality of the features we experimented with a random forest classifier and correlation and p-value algorithm. The features extracted from random forest classifiers were less in number than the features extracted from correlation and p-value algorithms which not only reduced the dimensionality of the features to greater extent but also improved the performance of the model.

We tested the model on CICIDS 2017 data set , the results show that the model gave 99 % accuracy, 0.99% precision and 100% precision respectively for dos attack. Looking at the results we can show that the proposed model performed better than machine learning models. We experimented with our system by deploying it on Raspberry pi 3B+ and launching a dos attack on it from multiple sources. The nature of the DOS traffic was monitored on Wire shark. The IDS was able to detect the nature of network traffic and raise an alert when the dos attack was in the accumulation stage, thus validating the performance of the proposed IDS.

The system also works in offline mode where it analyzes the network traffic which is recorded in pcap files.

FUTURE SCOPE

The traditional machine learning systems have been surpassed by the Deep learning model for the network data classification into normal/attack .

The responsibility for implementing appropriate security solutions does not depend on one team of the IoT ecosystem, but rather on all players involved, from silicon suppliers to manufacturers, developers, lawmakers, and end customers. Reducing the risk associated with safety violations is possible, if safety gets consideration in the planning and design of the product ahead of time, and if there are some basic safety measures in place. Operating and implementing common policies will simplify production and development processes, give the market a reason for more acceptance, and increase the safe standing of IoT products and services. Security will have to be built to enable the IoT to withstand the onslaught of threats posed by technological advances.

With the advancement of quantum computing technology, AI, and understanding systems, and with the continued development and widespread adoption of the IoT system, current security measures and methods will be part of the past. Quantum computing can not only penetrate any type of security known to humanity but can also provide a solution for obtaining a solid security formula. IoT will benefit greatly from this technological advancement, especially in the science of quantum mechanics on microchips. Another study is recommended, as the technology matures and develops, to find out how future security affects the Internet of Things.

REFERENCES

[1] Yavuz, F.Y. (2018). Deep learning in cyber security for internet of things.

[2]Muhuri, Pramita Sree, Prosenjit Chatterjee, Xiaohong Yuan, Kaushik Roy and Albert C. Esterline. “Using a Long Short-Term Memory Recurrent Neural Network (LSTM-RNN) to Classify Network Attacks.” *Inf.* 11 (2020): 243..

[3] Sun, Pengfei & Liu, Pengju & Li, Qi & Liu, Chenxi & Lu, Xiangling & Hao, Ruochen & Chen, Jinpeng. (2020). DL-IDS: Extracting features using CNN-LSTM hybrid network for intrusion detection system. Security and Communication Networks. 2020. 1-11. 10.1155/2020/8890306.

[4] Canedo, J., & Skjellum, A. (2016). Using machine learning to secure IoT systems. *2016 14th Annual Conference on Privacy, Security and Trust (PST)*, 219-222.

[5] Torres, P., Catania, C.A., García, S., & Garino, C.G. (2016). An analysis of Recurrent Neural Networks for Botnet detection behavior. *2016 IEEE Biennial Congress of Argentina (ARGENCON)*, 1-6.

[6] Bostani, H., & Sheikhan, M. (2017). Hybrid of anomaly-based and specification-based IDS for Internet of Things using unsupervised OPF based on MapReduce approach. *Comput. Commun., 98*, 52-71.

[7] P. Chaudhary and B. B. Gupta, "DDoS Detection Framework in Resource Constrained Internet of Things Domain," *2019 IEEE 8th Global Conference on Consumer Electronics (GCCE)*, 2019, pp. 675-678, doi: 10.1109/GCCE46687.2019.9015465.

[8]*Jeffrey Dean and Sanjay Ghemawat. 2008. MapReduce: simplified data processing on large clusters. Commun. ACM 51, 1 (January 2008), 107–113.*

[9] Saied, A., Overill, R. E., & Radzik, T. (2016). Detection of known and unknown DDoS attacks using Artificial Neural Networks. *Neurocomputing*, *172*, 385-393.

[10] Meidan, Y., Bohadana, M., Shabtai, A., Guarnizo, J.D., Ochoa, M., Tippenhauer, N., & Elovici, Y. (2017). ProfilIoT: a machine learning approach for IoT device identification based on network traffic analysis. *Proceedings of the Symposium on Applied Computing*.

[11]Yingxu Lai, Jingwen Zhang, Zenghui Liu, "Industrial Anomaly Detection and Attack Classification Method Based on Convolutional Neural Network", Security and Communication Networks, vol. 2019, Article ID 8124254, 11 pages, 2019.

[12]Kim, J., Kim, J., Thu, H.L., & Kim, H. (2016). Long Short Term Memory Recurrent Neural Network Classifier for Intrusion Detection. *2016 International Conference on Platform Technology and Service (PlatCon)*, 1-5.

[13]Habibi Lashkari, Arash. (2018). CICFlowmeter-V4.0 (formerly known as ISCXFlowMeter) is a network traffic Bi-flow generator and analyser for anomaly detection. https://github.com/ISCX/CICFlowMeter. 10.13140/RG.2.2.13827.20003.