Protocol-Based Deep Intrusion Detection for DoS and DDoS Attacks Using UNSW-NB15 and Bot-IoT Data-Sets

PRAMODH K

University of Visvesvaraya College of Engineering

ABSTRACT

Since its inception, the Internet of Things (IoT) has witnessed mushroom growth as a breakthrough technology. In a nutshell, IoT is the integration of devices and data such that processes are automated and centralized to a certain extent. IoT is revolutionizing the way business is done and is transforming society as a whole. As this technology advances further, the need to exploit detection and weakness awareness increases to prevent unauthorized access to critical resources and business functions, thereby rendering the system unavailable. Denial of Service (DoS) and Distributed DoS attacks are all too common. In this paper, we propose a Protocol Based Deep Intrusion Detection (PB-DID) architecture, in which we created a data-set of packets from IoT traffic by comparing features from the UNSWNB15 and Bot-IoT data-sets based on flow and Transmission Control Protocol (TCP). We classify non-anomalous, DoS, and DDoS traffic uniquely by taking care of the problems like imbalanced and over-fitting. We have achieved a classification accuracy of 96.3% by using deep learning (DL) technique.

CHAPTER - 1 INTRODUCTION

Home automation systems provide several ingress points – like smart meters, wireless lamps/bulbs, surveillance equipment, and smart thermostats, to name a few. Such connected devices with attached sensors provide hackers with a tremendous opportunity to exploit the system. While IoT has made the management of various daily tasks simple, it is essential to guarantee that criminals do not enter our homes using a loophole. As systems evolve and become security smart, hackers become smarter. IoT devices gather a huge amount of data during their lifespan. With the rapid growth of 5G implementation, data communication between devices and networks is projected to increase many folds. It is pertinent to note that if the data captured/generated by these devices is not secured, it remains available for stealing for financial gain or worst, it may put the lives of people at risk across the globe. The physical phenomenon, when captured into the digital domain (i.e., IoT), presents a broader range of potential

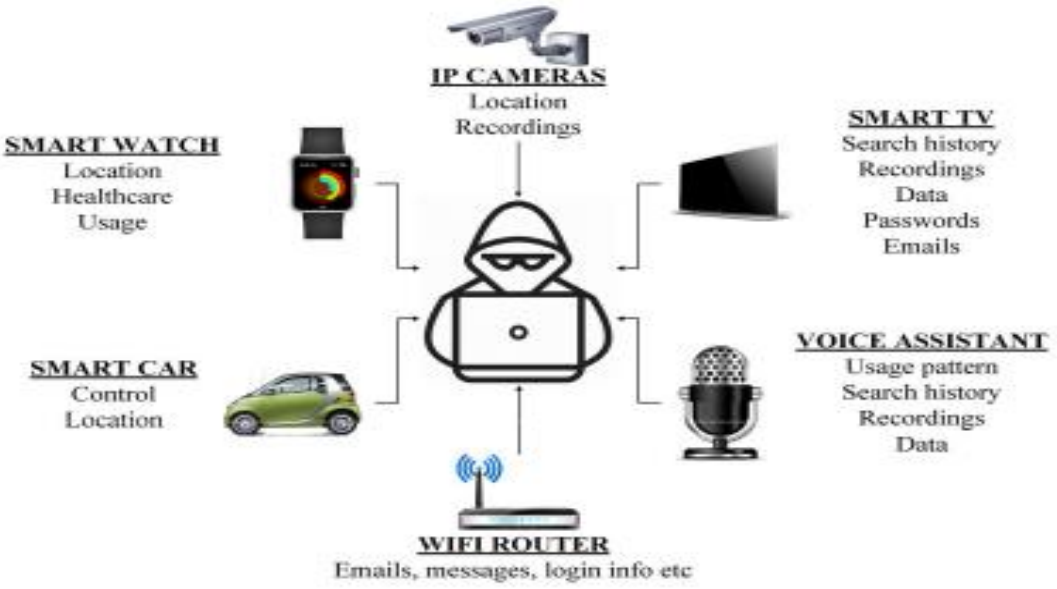

FIGURE 1. Smart devices with the information a hacker can retrieve.

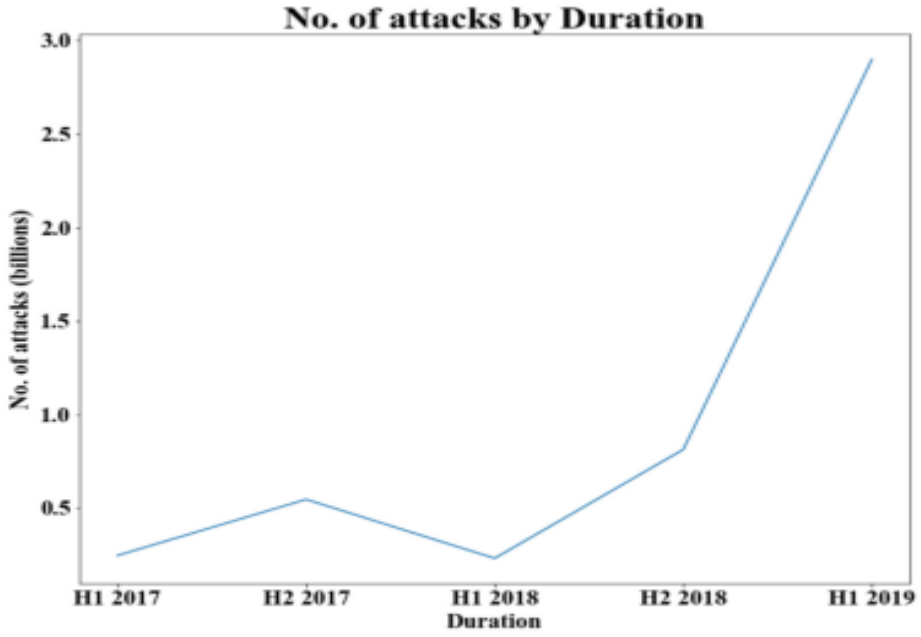

FIGURE 2. No. of cyber-attacks against IoT devices in each 6 months (one half) of the year [4].

loopholes that can be exploited. Ever growing need to properly equip IoT systems with adequate security is evidenced by the fact that about 50% of the world's leading exploits targeted IoT devices during 2018, and most of these were related to the exploitation of IP cameras. Cyber criminals might snuff out private communications, participate in disruptive on-site operations, or obtain a vantage point to trigger DDoS or ransomware attacks. Protective devices such as cameras are not immune to such attacks either. In the first half of 2019, cyber-attacks increased by more than three times the second half of 2018 and 2.98 billion cyber-attacks were recorded for IoT

devices (Fig. 2). This dramatic increase in attacks is due to an increase in the adoption of IoT devices. Approximately 30 billion IoT devices are connected to the internet in 2020, which will increase to 75 billion by the year 2025 (Fig. 3).

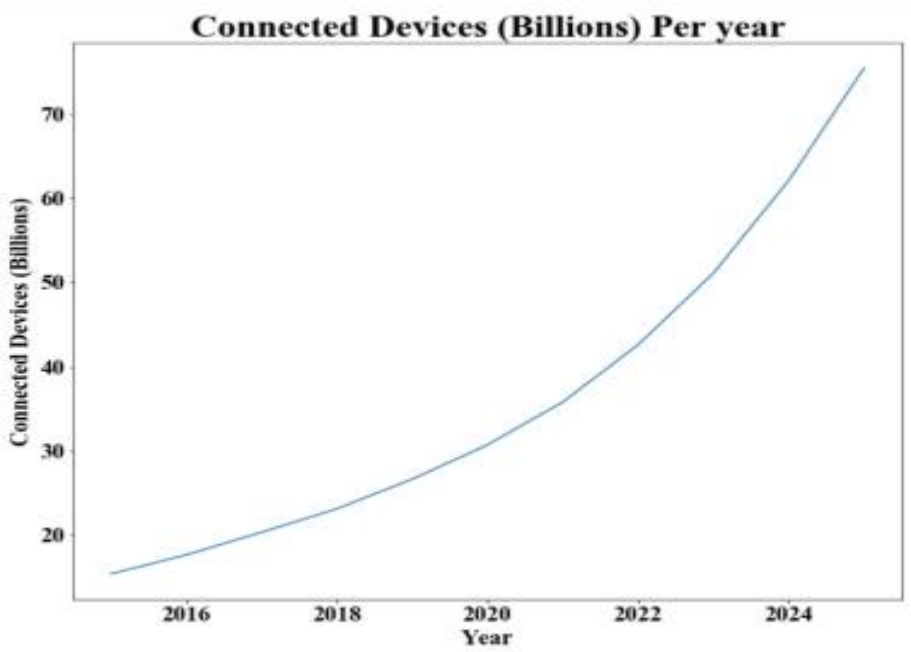

FIGURE 3. **No. of IoT devices connected to the internet for each year [6].**

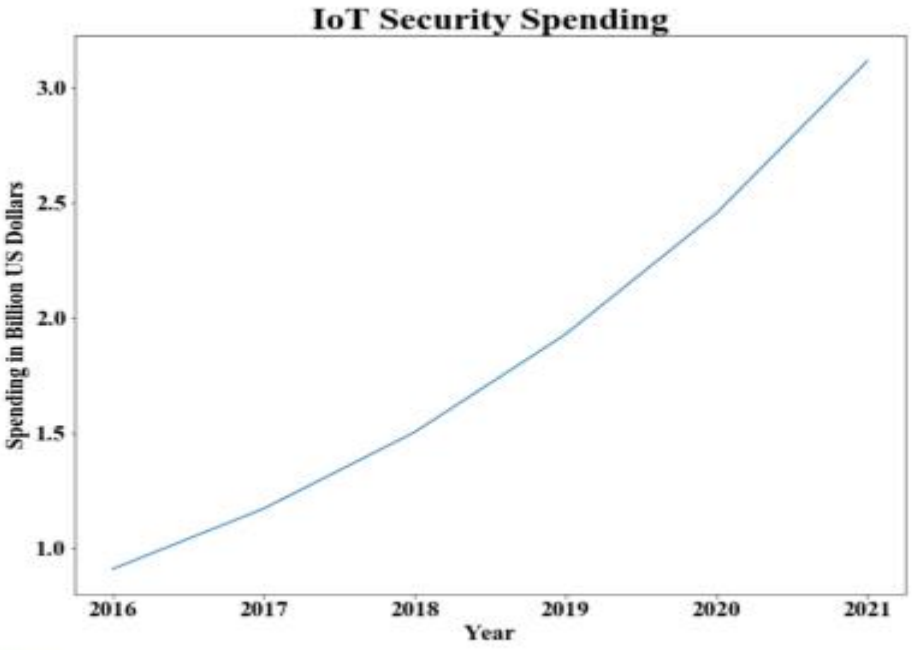

FIGURE 4. **IoT security spending worldwide from 2016-2021 [7].**

IoT security spending has reached 2.5 Billion US Dollars in 2020, almost 25% more than in 2019 (Fig. 4). According to Palo Alto Networks 2020 IoT threat report, 98% of IoT devices traffic is in plain and goes in the network unencrypted. Moreover, 51% of IoT threats involve medical care devices that disrupt healthcare quality

and put patient's data at risk. The healthcare VLAN network consists of both IoT and IT devices. This allows attackers to spread malware from computers to vulnerable IoT devices.

Above in view, a lot of research is focused on securing IoT devices against intrusion. Because great economic burden prevails in which huge revenue is spent on security of IoT devices. Many researchers have employed techniques for threat detection using supervised machine learning (ML) but few have employed solutions using deep learning. In these solutions, a major predicament comes when trying to acquire data related to IoT devices. Whichever data we acquire today will be insufficient tomorrow, as new, sophisticated and more complex attacks are created and implanted. The best solution to this problem is to acquire the data which is realistic and latest one in order to cover maximum portion of attacks happening today.

The major contributions of the work at hand are as follows:

• Comparison of the UNSW-NB15 and the Bot-IoT data sets to identify common features between them.

• Union of both data-sets using common features which fall in flow and TCP categories.

• Removal of issues pertaining to the imbalance nature of data-sets and biased classification.

• Employing LSTM based un-supervised deep learning model for the detection of DoS and DDoS attacks. The model is trained on all of the data available in the UNSW-NB15 and BotIoT data-sets to cover the maximum possible types of packets.

• The employed deep model has been cross validated using different performance metrics.

CHAPTER – 2 RELATED WORK

In the cybersecurity domain, towards cyber antagonists, an Intrusion Detection System (IDS) serves as a clear defense line, which is very critical for a system. Due to mobile device penetration and the popularity of apps that quickly accomplish various user tasks, we have become dependent on this parallel universe defined by electronic devices. Users need to be made aware of the implication of negligence towards secure practices to protect network infrastructure against intrusion threats. A device is considered protected if it successfully achieves data protection, confidentiality, Integrity and availability (CIA).
Attacks are designed to undermine the security systems in place. An intruder, by definition, is unwanted and must be kept outside the network in order to maintain a reliable authentication and authorization process. Denial of Service DoS attack, for instance, floods computing resources with information, which destroys the concept of availability, while malware disrupts the implementation of a program that infringes the concept of integrity. An IDS is a surveillance and review tool for operations in a computerized system or infrastructure to identify perceived threats by their observations of offences related to the principles of CIA in computer security policy.
To track network dependent system an IDS can be employed in host mode or network mode. A host dependent IDS (HIDS) tracks host affairs by gathering data of computer system activities. In this kind of device, a monitor must be mounted to track hosts and record their actions on the operating system for an investigation trail. It is therefore important that HIDS should be compliant with its devices to track various operating systems.
Network-based IDS (NIDS) tracks network traffic to detect mobile threats occurring via a network connection. This seems to be an effective security approach since, when entering a host infrastructure and documenting itself on operating systems audits, it offers a strong layer of protection towards a malicious activity. While a HIDS might identify breaches within hosts, this happens typically after accessing a host's system assets, including its data and services. To follow ''prevention is better than cure,'' the best safety approach is to avoid proven and zero-day assaults over networks before reaching the hosts. Since there can be no assurance that the device assets are not compromised, even if a HIDS senses a threat, the architecture of a smart NIDS allows means of preventing these attacks, but it is indeed quite challenging to implement this consistently. Hence, a HIDS and NIDS compound was constructed to create a hybrid IDS to track network activity and control host operations.

2.1 IDS TECHNIQUES BASED ON SUPERVISED ML IN IoT SCENARIO

In recent times, IDSs have drawn attention of several analysts and developers with increasingly deployed IoT devices in the IoT world. In certain works unique risks against IoT products are discussed. Cervantes et al., gave a solution for routing systems to detect sinkhole threats. In static and portable conditions, the classification

accuracy reached 92% and 72% respectively. In their study, Guo et al., proposed to deter threats to the Bluetooth Low Energy (BLE) platform on three separate rates of power drainage. and privacy can expose people's lives as well as their health to malicious individuals through hack attacks.

2.2 IDS TECHNIQUES BASED ON DEEP LEARNING MODELS IN IoT SCENARIO

Tang et al., [38] suggested a method for intrusion detection in Software Defined Networks (SDN) using Deep Neural Network. The proposed DL technique for IDS can assess all switches in OpenFlow and is deployed in the SDN controller. Binary classification (non-anomalous and malicious) is done using NSL-KDD data-set in which they only kept six features which can easily be obtained in SDN out of fourty one features of the data-set. Authors show that 0.001 as learning rate was most successful with maximum receiver operating characteristic curve when compared with other learners. The DL method was adopted by Potluri et al., [40] as the classification technique for the network information. The data-set they used for evaluation is NSL-KDD that comprises of 39 forms of attacks divided in four threat classes. Their analysis indicates that the binary classification rate is high.

Ahmad et. al., [17] applied supervised ML techniques on the UNSW-NB15 data-set by identifying and using only the application and transport layer features from the UNSWNB15 data-set. The authors applied RF, SVM and ANN on full features, flow & MQTT features, TCP features and top features from flow and TCP features for doing binary and multiclass classification. By applying RF they achieved best accuracy of above 98% in binary and above 97% in multiclass classification.

2.3 IMBALANCED DATA-SETS

There are two main methods to solve the issues of the imbalanced nature of data-sets, i.e., oversampling and undersampling. In under-sampling, a subset equivalent to other class(es) from the class with the majority of samples is gathered. For the under-sampling of the majority class, Japkowicz provides two simple ways. The first one is to consider a random sample that produces only a majority class subset by randomly selecting primary class samples from the data-set. Japkowicz refers to the other form as a focused sample. This implies that a subset is generated by removing outliers of the dominant class.

2.4 STATE OF THE ART USING THE UNSW-NB15 AND THE BOT-IoT DATA-SETS

Larriva et al., [10] proposed pre-processing techniques on the UNSW-NB15, UGR16 and NSL-KDD data-sets. They applied different pre-processing techniques such as z-score, min–max, and no pre-processing on different data-sets of predefined categories. For classification, MLP was used and any accuracy of 99.7%, 99.3%, and 99.2% was reported for NSL-KDD, UGR16, and the UNSW-NB15 data-sets respectively. However, no comparison between the data-sets was presented by the authors. Synthetic Minority Oversampling Technique (SMOTE) is proposed on the Bot-IoT data-set in, which caters for data imbalance in order to avoid over/under fitting issues. The authors created a balanced data set by using SMOTE, which generates synthetic examples using techniques such as rotation and skew in order to achieve class balance. They applied normalization on feature set and using Deep RNN (DRNN) achieved an accuracy of 100%. However, the normalization was applied on derived data points and hence undermined the realistic nature of the data-set. Such normalization of features compromises the underlying variance phenomenon present in the data-sets.

CHAPTER – 3 PROPOSED METHODOLOGY

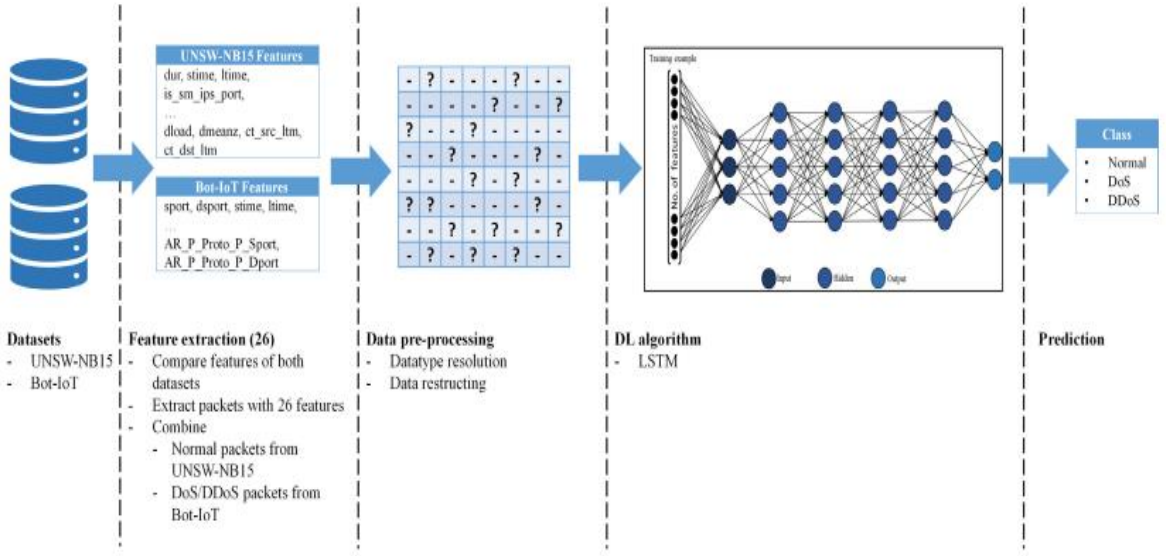

FIGURE 5. Proposed structure of PB-DID for attack classification using DL.

In this section, we discuss the proposed Protocol Based Deep Intrusion Detection (PB-DID) architecture as shown in Fig. 5. The process involves features comparison to find similar features in the UNSW-NB15 and the Bot-IoT datasets, features selection, data pre-processing and selection and model training using un-supervised LSTM deep learning model.

3.1 DATA-SETS

We used two well-known raw network packets data-sets namely the UNSW-NB15 and the Bot-IoT for training and validation. Unlike many other studies which used a smaller part of these data-sets, we have used complete data for model training. Brief description of both data-sets is given below.

3.1.1 UNSW-NB15

The data-set was published by Moustafa et al., [8] in 2015. This data-set is simulated with over 2.5 million network packets. This data-set consists of nine types of attacks (Exploits, Recconaisance, DoS, Generic, Shellcode, Fuzzers, Backdoors, Worms and Analysis)) with non-anomalous packets as well. More than 87% packets are of nonanomalous type which makes the data-set highly imbalanced. Packets distribution is given in Table 1. More details about this data-set can be found in.

3.1.2 BOT-IoT DATA-SET

This data-set is the latest in the field. Koroniotis et al., published the data-set in 2018. It consists of more than 72 million records with the mixture of simulated and real time scenarios. It has four categories of attacks but major portion of data-set has DoS and DDoS type of packets. This data-set is imbalanced just like the UNSW-NB15 data-set. Records distribution of the Bot-IoT data-set is given in Table 2. More details about the data-set can be found in.

TABLE 1. Class distribution of UNSW-NB15 data-set.

Class	No. of records	% of total data
Non-anomalous (normal)	2,218,761	87.35
Exploits	44,525	1.75
Reconnaissance	13,987	0.55
DoS	16,353	0.64
Generic	215,481	8.48
Shellcode	1,511	0.06
Fuzzers	24,246	0.95
Analysis	2,677	0.11
Backdoor	2,329	0.1
Worms	174	0.01
Total	**2,540,044**	

TABLE 2. Class distribution in the Bot-IoT data-set.

Class	No. of records	% of total data
Non-anomalous (normal)	9,543	0.013
Information gathering	1,821,639	2.480
DDoS	38,532,480	52.500
DoS	33,005,194	45.000
Information theft	1,587	0.002
Total	**73,370,443**	

3.2 FEATURES COMPARISON

In the UNSW-NB15, there are 49 features including 48th as a multi-class label and 49th as a binary label. In the BotIoT data-set, there are 46 features and the last three are label features. In proposed PB-DID, the features in both data-set are compared and we found that 29 features in the Bot-IoT are similar or can be evaluated in the UNSW-NB15 data-set as well. The list of features is given in Table 3.

TABLE 3. Common features in the Bot-IoT [9] and the UNSW-NB15 [8] data-sets. Features from 18-29 are used for training and validation in the work at hand.

Ser No.	Feature	Description
1	Stime	Start time of connection
2	Proto	Protocol being used in the flow
3	proto number	'Proto' feature representation in numerical form
4	saddr	IP of source
5	sport	Port number of source
6	daddr	IP address of destination
7	dport	Port number of destination
8	pkts	No. of packets involved in current flow
9	bytes	No. of bytes involved in current flow
10	state	State of current transaction
11	state number	'state' feature representation in numerical form
12	ltime	Finish time of connection
13	dur	Total record duration
14	spkts	No. of packets involved in current flow from source-to-destination
15	dpkts	No. of packets involved in current flow from destination-to-source
16	sbytes	No. of bytes involved in current flow from source-to-destination
17	dbytes	No. of bytes involved in current flow from destination-to-source
18	TnBPSrcIP	Total 'bytes' of a 'saddr' in 100 connections
19	TnBPDstIP	Total 'bytes' of a 'daddr' in 100 connections
20	TnP_PSrcIP	Total 'pkts' of a 'saddr' in 100 connections
21	TnP_PDstIP	Total 'pkts' of a 'daddr' in 100 connections
22	TnP_PerProto	Total 'pkts' of a 'proto' in 100 connections
23	TnP_Per_Dport	Total 'pkts' of a 'dport' in 100 connections
24	AR_P_Proto_P_SrcIP	Avg 'pkts' / 'dur' per 'saddr' in 100 connections
25	AR_P_Proto_P_DstIP	Avg 'pkts' / 'dur' per 'daddr' in 100 connections
26	N_IN_Conn_P_SrcIP	Connections with same 'saddr' in 100 connections
27	N_IN_Conn_P_DstIP	Connections with same 'daddr' in 100 connections
28	AR_P_Proto_P_Sport	Avg 'pkts' / 'dur' per 'sport' in 100 connections
29	AR_P_Proto_P_Dport	Avg 'pkts' / 'dur' per 'sport' in 100 connections

TABLE 4. Features categorization according to clusters taken from the Bot-IoT [9] and the UNSW-NB15 [8].

Ser No.	Feature	Cluster	Ser No.	Feature	Cluster
1	Stime	Flow	10	sport	TCP
2	ltime		11	dport	
3	saddr		12	pkts	
4	daddr		13	bytes	
5	dur		14	proto number	
6	AR_P_Proto_P_SrcIP		15	spkts	
7	AR_P_Proto_P_DstIP		16	dpkts	
8	N_IN_Conn_P_SrcIP		17	sbytes	
9	N_IN_Conn_P_DstIP		18	dbytes	
			19	TnBPSrcIP	
			20	TnBPDstIP	
			21	TnP_PSrcIP	
			22	TnP_PDstIP	
			23	TnP_PerProto	
			24	TnP_Per_Dport	
			25	AR_P_Proto_P_Sport	
			26	AR_P_Proto_P_Dport	

3.3 FEATURES SELECTION

In the proposed PB-DID architecture, clusters of features in both data-sets are created according to flow, Domain Name System (DNS)/File Transfer Protocol (FTP)/Hypertext Transfer Protocol (HTTP), Message Queuing Telemetry Transport (MQTT) and TCP. The major portion of features falls into two clusters i.e. flow and TCP. The clusters are given in Table 4. Here the clusters are created by analyzing each feature's description reported by the authors. We kept a minimum number of features while covering the application and transport layer. Major contributions from the application layer are the flow features whereas from the transport layer are of TCP protocol. Therefore, both of these clusters are chosen to create optimized scenarios by keeping maximum information of a packet. This approach significantly reduces the computational time required during the learning phase.

3.4 DATA PRE-PROCESSING

In this section we explain different data pre-processing steps.

3.4.1 DATA TYPE RESOLUTION Some of the features used in PB-DID such as 'saddr', 'daddr', and 'proto' (see Table 4) are of the categorical type that needs to be converted into algorithm executable form. The 'saddr' and 'daddr' are source and destination IP addresses respectively, whereas 'proto' is the protocol type being used in the flow. We have assigned numerical values to all source and destination IP addresses. In the UNSW-NB15

data-set, a total of 49 IP addresses are used, whereas in the Bot-IoT data-set, a total of 301 IP addresses are used. In the process of merging both data-sets, we replaced the IP addresses with 350 randomly generated unique integer numbers. This anonymization of the IP addresses greatly helps to rule out over-fitting. Furthermore, it also helps in keeping the IP addresses in the training and validation data-sets as there exist a few features which are evaluated based upon the IP addresses e.g., NINConnPS rcIP, NINConnPDstIP. These features will be meaningless if the IP addresses are entirely removed. Similarly, we have converted the 'proto' feature into an integer type.

3.4.2 MISSING PORT NUMBERS

In the Bot-IoT full data-set, the packets using ARP protocol have missing source and destination port numbers, which is understandable. Koroniotis et al., in mentioned that they have given -1 as port numbers where ARP protocol is used in the 5% extracted the Bot-IoT data-set. We used the same value in PB-DID and assigned it to port numbers in full dataset where ARP protocol is used.

3.4.3 RESOLVING THE DATA IMBALANCE ISSUE

Imbalance data is a well-known problem in machine learning which occurs when the distribution of different classes is biased. In an imbalanced data-set, the distribution of different classes can be slightly imbalanced or severely imbalanced. Any learning model trained over a severely imbalanced dataset will result in poor predictive performance against minor classes. The UNSW-NB15 and BoT-IoT data-sets are good examples of imbalance data as 87.35% data in UNSW-NB15 is non-anomalous (Table. 1), whereas, only 0.013% data in BoT-IoT is non-anomalous data (Table. 2). Moreover, in the BoT-IoT data-set, around 52.5% of the data is of type DDoS and around 45% of the data is of type DoS. Hence, none of the two data-sets can be solely used to train and predict nonanomalous (normal), DDoS, or DoS packets. For this reason, a merger of both data-sets is essential to achieve meaningful predictions.

The process of merging the data from UNSW-NB15 and BoT-IoT data-sets is presented in Figure 6. There are around 2.218 million non-anomalous packets in UNSW-NB15 dataset whereas, in BoT-IoT data-set, 38.5 million packets are of type DDOS and 33 million packets are of type DoS. In the proposed PB-DID, we consider 2.218 non-anomalous packets as a complete data unit and create 14 equal data chunks for DDoS and DOS packets. Each of the 14 chunks contain 2.218 million unique packets of type DDos and DoS i.e., for each chunk, there are 2.218 million non-anomalous packets, 2.218 million DDoS packets, and 2.218 million DoS packets. Hence, each of the 14 chunks contains a total of 6.654 million packets (see Figure 6). This strategy assures equal distribution of the data for the three classes in all chunks and mitigates the problem of over-fitting. As shown in Figure 6, in each data chunk, the non-anomalous samples are repeated, whereas, the DDoS and DoS samples are always unique.

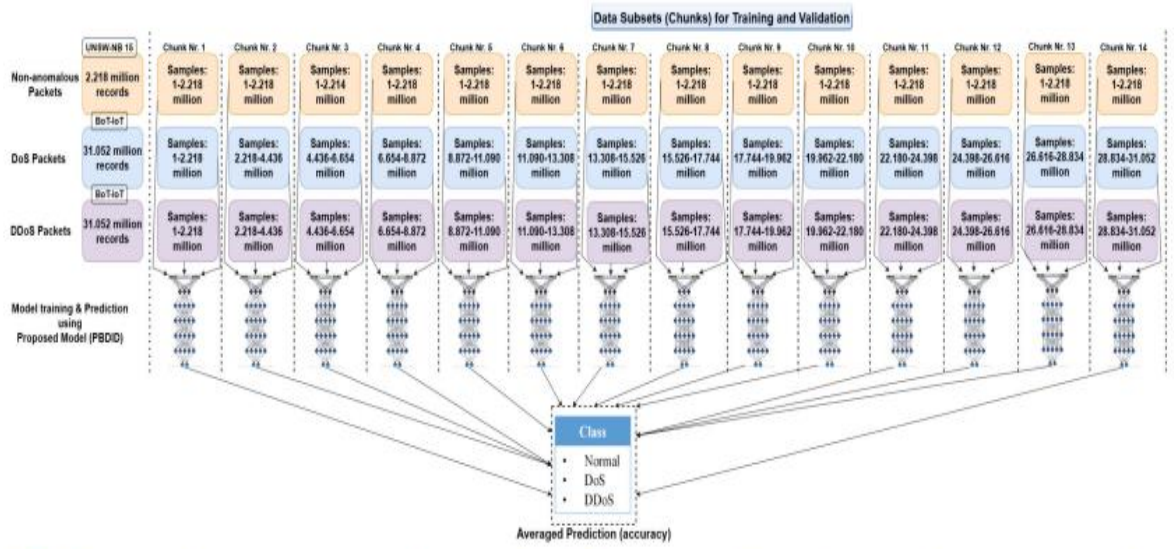

FIGURE 6. The process of merging the data from UNSW-NB15 and BoT-IoT data-sets is presented here. To mitigate over-fitting, the same number of samples for all three classes are kept in each data chunk. The proposed PB-DID model is trained separately on each data chunk and an averaged prediction (in % accuracy) overall 14 data chunks is computed.

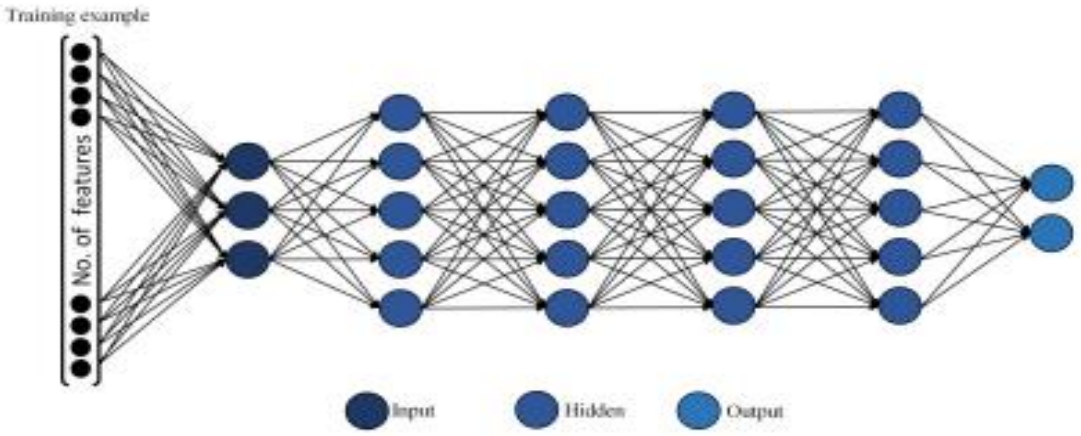

FIGURE 7. Basic structure of a DL algorithm.

3.5 DEEP INTRUSION DETECTION

Deep Learning is a subclass of ML which mainly uses hierarchical stages in Artificial Neural Networks (ANN). Just like human brain, ANN's are built as a web linking neuron nodes. Although standard algorithms linearly build insights of data, the hierarchy of DL systems allows computers to interpret the data in a nonlinear way [50]. The basic structure of a DL algorithm is shown in Fig. 7.

3.5.1 MODEL SELECTION

The deep model used in PB-DID architecture is an LSTM model with an input layer, two hidden layers and an output layer. The input layer is embedding layer which takes the input of batches (26 x 128) created in section 3.4.3 and gives an output of 16 dimensions which is given as input to first of two hidden LSTM layers. The embedding layer creates a vector of each training example. It works similarly as the one-hot encoding function in Keras works. One hot encoding function is used for one feature at a time, but all the features are used simultaneously in the embedding layer. Each entry of a vector is initialized with random weights, and the embedding layer automatically learns the weights with each iteration. Both LSTM layers have 20 nodes, which give an output of 20 dimensions. In LSTM layers, we use activation, recurrent activation functions and dropout, recurrent dropout functions. We have used two types of output layers, one for the binary classification and the other for multi-class classification. In binary classification, the last layer of the model is a dense layer with two neurons. There are three types of outputs, one for non-anomalous and DDoS, second for non-anomalous and DoS, and third for DDoS and DoS packets making it a binary classification. Here it is pertinent to mention that in this classification, we give the input of only those packets to the model for which we are doing the classification. In multi-class, the last layer of the model is a dense layer with three neurons, which outputs each class's probabilities, as shown in Fig. 8.

3.6 PERFORMANCE METRICS

We used two metrics for measuring the performance of the proposed PB-DID model namely confusion matrices and accuracy score. In a confusion matrix, there are four possible options, true positive (TP), true negative (TN), false positive (FP) and false negative (FN). The first part in every option shows whether the prediction is true or false and second part shows that prediction is positive or negative. The accuracy score shows the accuracy score of the predictions by the underlying deep model.

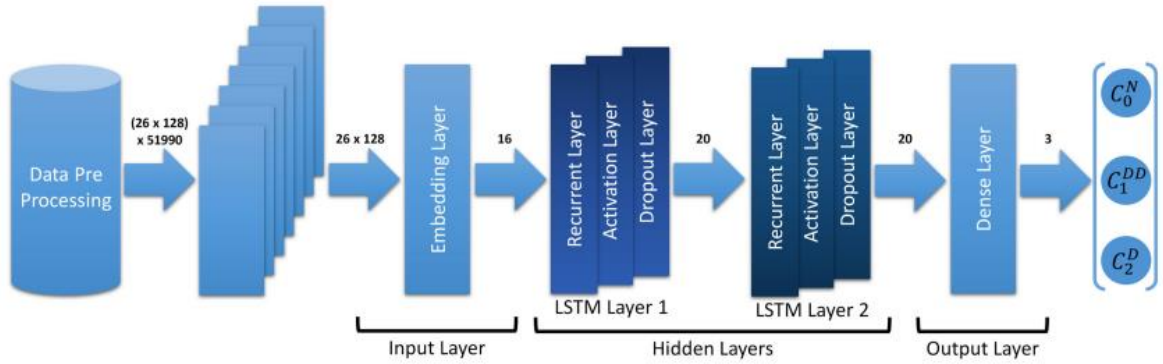

FIGURE 8. Structure of LSTM model showing input, hidden and output layers. In output layer C^i_j, i = 0, 1 or 2 is the label and j = non-anomalous (N) when i = 0, DDOS (DD) when i = 1 and DOS (D) when i = 2 is the class.

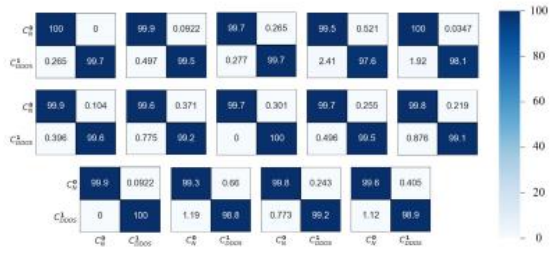

FIGURE 9. Confusion matrices of all 14 data chunks, C^i_j, i = 0 or 1 is the label and j = non-anomalous (N) when i = 0 and DDOS (DD) when i = 1 is the class.

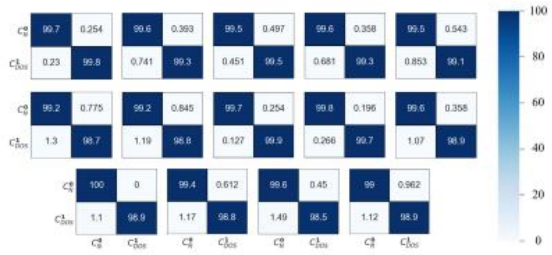

FIGURE 11. Confusion matrices of all 14 data chunks, C^i_j, i = 0 or 1 is the label and j = non-anomalous (N) when i = 0 and DOS (D) when i = 1 is the class.

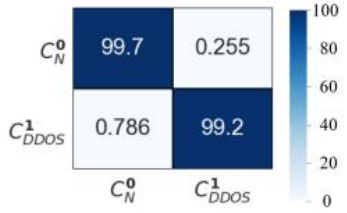

FIGURE 10. Confusion matrix calculated by averaging all 14 data chunk confusion matrices involving non-anomalous and DDoS packets.

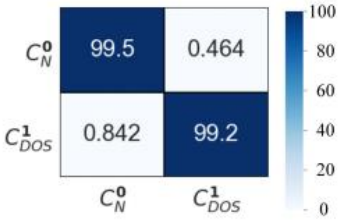

FIGURE 12. Confusion matrix calculated by averaging all 14 data chunk confusion matrices involving non-anomalous and DoS packets.

CHAPTER – 4 EXPERIMENTS AND RESULTS

In this section we discuss the results of experiments performed using the proposed PB-DID architecture. All experiments are performed on google colab using tensorflow as back-end and the language used is Python3.

4.1 CLASSIFICATION RESULTS

4.1.1 BINARY CLASSIFICATION

We performed binary classification by taking two classes at a time from a total of three classes which gives us three different configurations of experiments. Fig. 9 shows the heat map of fourteen confusion matrices for each chunk of DDoS as discussed in section 3.4.3 and Fig. 10 shows the overall confusion matrix after averaging the results of all chunks of DDoS packets. The classification accuracies for non anomalous vs. DDOS (anomolus) packets remain above 99%. Similarly, Fig. 11 shows the heat map of fourteen confusion matrices for each chunk of DoS and Fig. 12 shows the overall

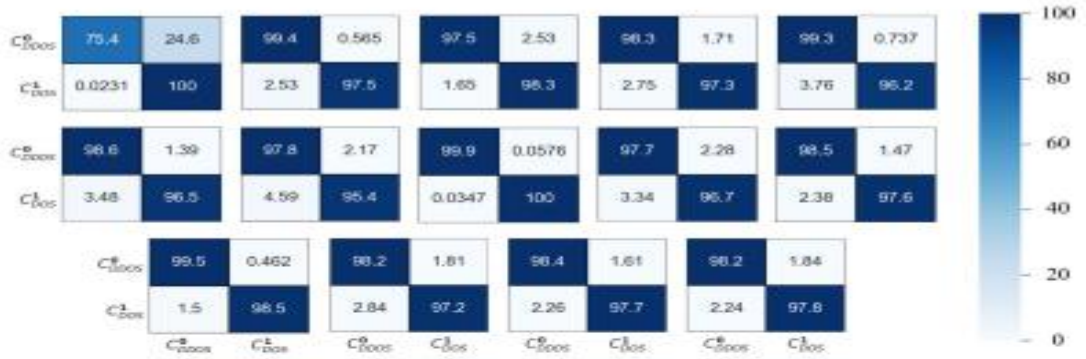

FIGURE 13. Confusion matrices of all 14 data chunks, C^i_j, i = 0 or 1 is the label and j = DDOS (DD) when i = 0 and DOS (D) when i = 1 is the class.

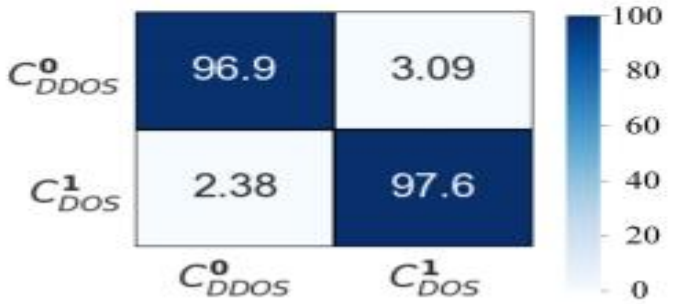

FIGURE 14. Confusion matrix calculated by averaging all 14 data chunk confusion matrices involving DDOS and DoS packets.

confusion matrix after averaging the results of all chunks of DoS packets. Again, the classification accuracies for non anomalous vs. DOS (anamolus) packets remain above 99%. Likewise, Fig. 13 shows the heat map of fourteen confusion matrices for each chunk of DDoS and DoS and Fig. 14 shows the overall confusion matrix after averaging the results of all chunks of DoS packets. The classification accuracies for DDOS (anomalous) vs. DOS (anamolus) packets remain above 97%. The accuracy scores of all three configurations of all fourteen chunks and the overall accuracy achieved is given in Table 5.

TABLE 5. Accuracy score of all chunks of DDOS / DoS packets and overall accuracy achieved.

Chunk	Accuracy (%)			
	non-anomalous/DDoS	non-anomalous/DoS	DDoS/DoS	Multi-class
1	99.86728217	99.7576457	87.6745528	97.66301212
2	99.70569565	99.43313281	98.45354876	96.15356783
3	99.7289035	99.52614851	97.90727252	95.257184
4	98.52864809	99.48066936	97.77125086	99.22677438
5	99.01904212	99.30178881	97.75533756	99.71148298
6	99.75034835	98.96400046	97.56618029	99.50567417
7	99.42695068	98.98242368	96.618805	86.85740734
8	99.84997115	99.80957877	99.95383728	93.8900158
9	99.62449451	99.76905312	97.18455527	96.13593025
10	99.45191254	99.28414733	98.07592304	91.22459121
11	99.95383728	99.45181766	99.01904212	98.618256
12	99.07546516	99.10972367	97.6707985	98.605389
13	99.49165271	99.030359	98.06358382	97.70032827
14	99.23527026	98.95905621	97.96342237	97.87800643
Overall	**99.48**	**99.35**	**97.26**	**96.32**

4.1.2 MULTI-CLASS CLASSIFICATION

Fig. 15 shows heat map of fourteen confusion matrices for each chunk of non-anomalous, DoS and DDoS cases as discussed in section 3.4.3 and Fig. 16 shows overall confusion matrix after averaging the results from all the chunks. The average classification accuracy remains above 96% where 99.7% are correctly classified as non-

anomalous. It is observable in the confusion matrix that around 6% of the DOS packets are misclassified as DDOS packets whereas around 4% of the DDOS packets are misclassified as DOS packets. The accuracy score of all fourteen chunks and overall accuracy achieved is given in the last column of Table 5.

FIGURE 15. Confusion matrices of all 14 data chunks, C^i_j, i = 0, 1 or 2 is the label and j = non-anomalous (N) when i = 0, DDOS when i = 1 and DOS when i = 2 is the class.

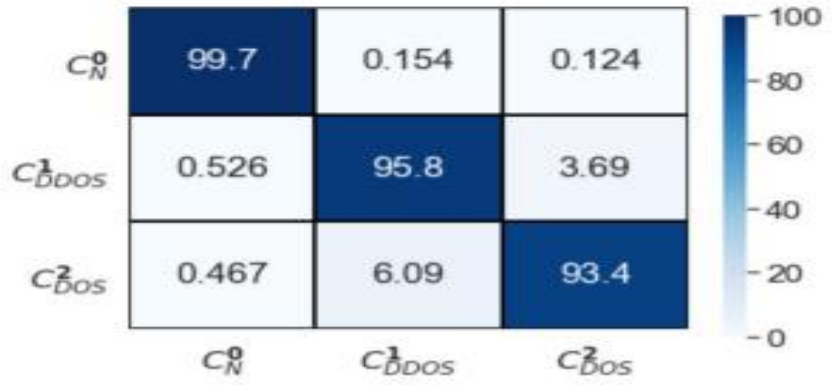

FIGURE 16. Confusion matrix calculated by averaging all 14 data chunk confusion matrices.

4.2 PARAMETERS TUNING

We performed parameter tuning by taking a bracket of values and performing the experiments. The optimal values are chosen, where we achieved the best accuracy. Five parameters are involved in parameter tuning: dropout, recurrent dropout, activation function, recurrent activation function, and number of epochs. Dropout and recurrent dropout are tested over a range of 0 to 1 (Figs. 17, 18,); activation and recurrent activation are tested over RELU, sigmoid, and tanh functions (Fig. 20, 21). The epochs are tested over a range 1 to 10 (Fig. 19). The model summary of our methodology with optimal parameters is given in Table 6.

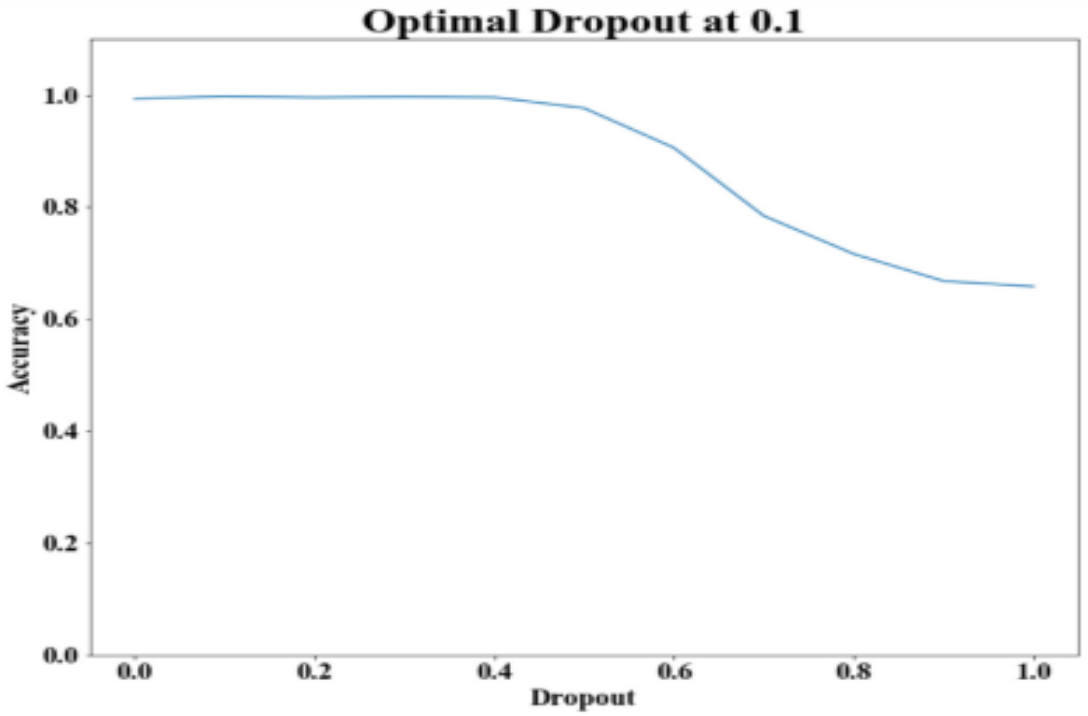

FIGURE 17. Optimal dropout graph over values of [0,1] where best accuracy of 99.79% is achieved at ***0.1***.

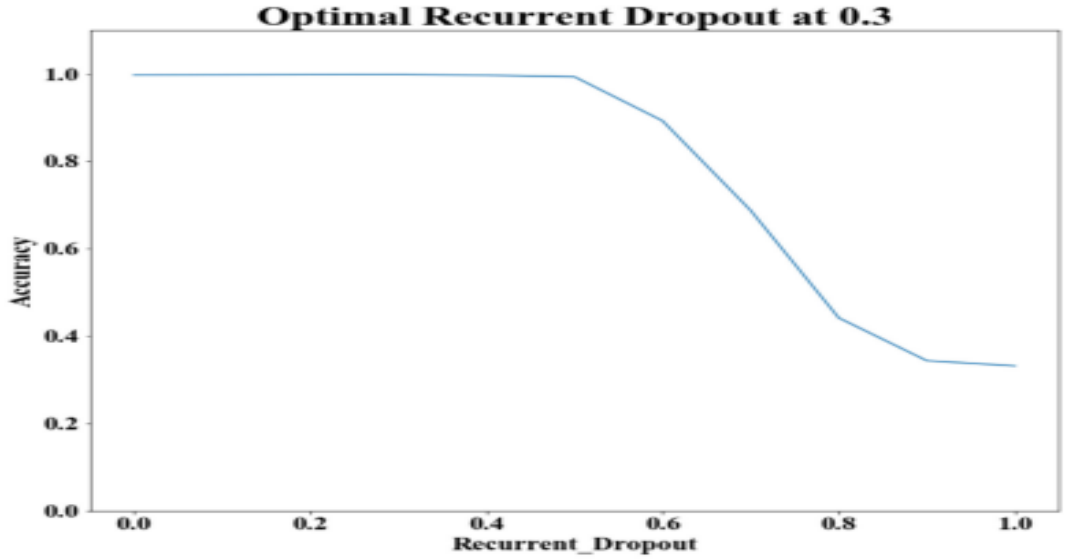

FIGURE 18. Optimal recurrent dropout graph over [0,1] and best accuracy 99.885% at ***0.3*** is achieved.

TABLE 6. LSTM model summary. * in output layer and total shows the parameters involved in binary/multi-class classification.

Layer	Output shape	Parameters	Dropout		Activation	
			Dropout	Recurrent dropout	Activation	Recurrent activation
Embedding	16	53248	-	-	-	-
LSTM	20	2960	0.1	0.3	tanh	sigmoid
LSTM	20	3280	0.1	0.3	tanh	sigmoid
Dense	2/3*	42/63*	-	-	Softmax	-
Total	-	59530/59551*	-	-	-	-

CHAPTER– 5 CONCLUSION

In this paper, PB-DID is proposed in which we have compared the features of the two latest benchmark data sets, the UNSW-NB15 and the Bot-IoT. Both data-sets are created by researchers of the University of New South Wales. In PB-DID, the standard features of flow and TCP category among both data-sets are analyzed and combined with those features. The problems of public data-sets like imbalance in nature and over-fitting are solved by selecting an equal number of packets from each category. We classified nonanomalous, DoS, and DDoS traffic by employing the DL technique and achieved an accuracy of 96.3% by covering almost both data-sets in full. This work is unique in the way that we have reduced (almost half) the number of features given for the identification of malicious traffic and covered two latest bench-marked data-sets. We aim to improve the feature comparison and selection technique by using other renowned and benchmark data-sets in the future. We will add more attack types to cover the vast majority of threats to IoT devices today in classification.

BIBLIOGRAPHY

[1] O. Amir. (Nov. 2019). Cyber Threats to IoT in 2020. [Online]. Available: https://www.techradar.com/news/cyber-threats-to-iot-in-2020

[2] LIFARS. (Mar. 2020). Impact of 5G Network on IoT Security. [Online]. Available: https://lifars.com/2020/03/impact-of-5g-network-oniot-security

[3] D. Ciso. (Mar. 2019). 50% of Top 12 Global Exploits Targeted IoT Devices: Fortinet Threat Landscape Report. [Online]. Available: https://www.dynamicciso.com/50-of-top-12-global-exploits-targeted-iotdevices-fortinet-threat-landscape-report

[4] Z. Doffman. (Sep. 2019). Cyberattacks on IoT Devices Surge 300% in 2019, 'Measured in Billions', Report Claims. [Online]. Available: https://www.forbes.com/sites/zakdoffman/2019/09/14/ dangerouscyberattacks-on-iot-devices-up-300-in-2019-now-rampantreportclaims/#2abdaf3f5892 [5] Unit 42. (Mar. 2020). 2020 Unit 42 IoT Threat Report. [Online]. Available: https://unit42.paloaltonetworks.com/iot-threat-report-2020/

[6] SR Department. (Nov. 2016). Internet of Things (IoT) Connected Devices Installed Base Worldwide From 2015 to 2025. [Online]. Available: https://www.statista.com/statistics/471264/iot-number-of-connecteddevices-worldwide/

[7] T. Alsop. (Jun. 2020). Internet of Things Security Spending Worldwide From 2016 to 2021. [Online]. Available: https://www.statista. com/statistics/543089/iot-security-spending-worldwide/ [8] N. Moustafa and J. Slay, ''UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set),'' in Proc. IEEE Military Commun. Inf. Syst. Conf. (MilCIS), Nov. 2015, pp. 1–6.

[9] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, ''Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset,'' Future Gener. Comput. Syst., vol. 100, pp. 779–796, Nov. 2019.

[10] X. Larriva-Novo, V. A. Villagrá, M. Vega-Barbas, D. Rivera, and M. S. Rodrigo, ''An IoT-focused intrusion detection system approach based on preprocessing characterization for cybersecurity datasets,'' Sensors, vol. 21, no. 2, p. 656, Jan. 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/2/656

[11] S. I. Popoola, B. Adebisi, R. Ande, M. Hammoudeh, K. Anoh, and A. A. Atayero, ''SMOTE-DRNN: A deep learning algorithm for botnet detection in the Internet-of-Things networks,'' Sensors, vol. 21, no. 9, p. 2985, Apr. 2021. [Online]. Available: https://www.mdpi.com/1424- 8220/21/9/2985

[12] A. Churcher, R. Ullah, J. Ahmad, S. ur Rehman, F. Masood, M. Gogate, F. Alqahtani, B. Nour, and W. J. Buchanan, ‘‘An experimental analysis of attack classification using machine learning in IoT networks,’’ Sensors, vol. 21, no. 2, p. 446, Jan. 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/2/446

[13] Y. Yang, K. Zheng, C. Wu, and Y. Yang, ‘‘Improving the classification effectiveness of intrusion detection by using improved conditional variational AutoEncoder and deep neural network,’’ Sensors, vol. 19, no. 11, p. 2528, Jun. 2019. [Online]. Available: https://www.mdpi.com/1424- 8220/19/11/2528

[14] M. Shafiq, Z. Tian, A. K. Bashir, X. Du, and M. Guizani, ‘‘CorrAUC: A malicious bot-IoT traffic detection method in IoT network using machine-learning techniques,’’ IEEE Internet Things J., vol. 8, no. 5, pp. 3242–3254, Mar. 2021.

[15] N. Guizani and A. Ghafoor, ‘‘A network function virtualization system for detecting malware in large IoT based networks,’’ IEEE J. Sel. Areas Commun., vol. 38, no. 6, pp. 1218–1228, Jun. 2020.

[16] O. Alkadi, N. Moustafa, B. Turnbull, and K.-K.-R. Choo, ‘‘A deep blockchain framework-enabled collaborative intrusion detection for protecting IoT and cloud networks,’’ IEEE Internet Things J., vol. 8, no. 12, pp. 9463–9472, Jun. 2021.

[17] M. Ahmad, Q. Riaz, M. Zeeshan, H. Tahir, S. A. Haider, and M. S. Khan, ‘‘Intrusion detection in Internet of Things using supervised machine learning based on application and transport layer features using UNSWNB15 data-set,’’ EURASIP J. Wireless Commun. Netw., vol. 2021, no. 1, pp. 1–23, Dec. 2021.

[18] A. Shameli-Sendi, M. Cheriet, and A. Hamou-Lhadj, ‘‘Taxonomy of intrusion risk assessment and response system,’’ Comput. Secur., vol. 45, pp. 1–16, Sep. 2014.

[19] S. Pontarelli, G. Bianchi, and S. Teofili, ‘‘Traffic-aware design of a highspeed FPGA network intrusion detection system,’’ IEEE Trans. Comput., vol. 62, no. 11, pp. 2322–2334, Nov. 2013.

[20] Z. Inayat, A. Gani, N. B. Anuar, M. K. Khan, and S. Anwar, ‘‘Intrusion response systems: Foundations, design, and challenges,’’ J. Netw. Comput. Appl., vol. 62, pp. 53–74, Feb. 2016.

[21] S. Anwar, J. M. Zain, M. F. Zolkipli, Z. Inayat, S. Khan, B. Anthony, and V. Chang, ‘‘From intrusion detection to an intrusion response system: Fundamentals, requirements, and future directions,’’ Algorithms, vol. 10, no. 2, p. 39, Mar. 2017.

[22] (2020). DDoS Threat Report 2020 Q1. [Online]. Available: https://blog.nexusguard.com/threat-report/ddos-threat-report-2020-q1

[23] D. Moon, S. B. Pan, and I. Kim, ‘‘Host-based intrusion detection system for secure human-centric computing,’’ J. Supercomput., vol. 72, no. 7, pp. 2520–2536, Jul. 2016.

[24] H. A. Kholidy and C. F. Baiardi, ‘‘A framework for intrusion detection in cloud systems,’’ in Proc. 9th Int. Conf. Inf. Technol.-New Generat., 2012, p. 978.

[25] K. E. Price, ‘‘Host-based misuse detection and conventional operating systems audit data collection,’’ M.S. thesis, Purdue Univ., West Lafayette, IN, USA, 1997.