IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 23, 2021, accepted December 14, 2021, date of publication December 21, 2021,
date of current version January 7, 2022.

Digital Object Identifier 10.1109/ACCESS.2021.3137201

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

MUHAMMAD ZEESHAN[1], (Senior Member, IEEE), QAISER RIAZ[1], (Senior Member, IEEE), MUHAMMAD AHMAD BILAL[1], MUHAMMAD K. SHAHZAD[1], HAJIRA JABEEN[2], (Member, IEEE), SYED ALI HAIDER[3], AND AZIZUR RAHIM[1]

[1]Department of Computing, School of Electrical Engineering and Computer Science, National University of Sciences and Technology, Islamabad 44000, Pakistan
[2]CEPLAS—Cluster of Excellence on Plant Sciences, University of Cologne, Albertus-Magnus-Platz, 50923 Köln, Germany
[3]Department of Computer and Information Sciences, State University of New York at Fredonia, Fredonia, NY 14063, USA

Corresponding author: Qaiser Riaz (qaiser.riaz@seecs.edu.pk)

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.

INDEX TERMS Intrusion detection in IoT, deep learning for intrusion detection, DoS detection, DDoS detection.

I. 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 [1]. 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 [2]. 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.

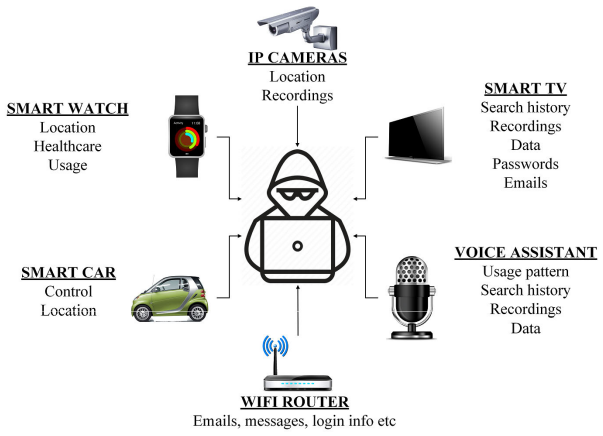


FIGURE 1. Smart devices with the information a hacker can retrieve.

The physical phenomenon, when captured into the digital domain (i.e., IoT), presents a broader range of potential

The associate editor coordinating the review of this manuscript and approving it for publication was Fan Zhang.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

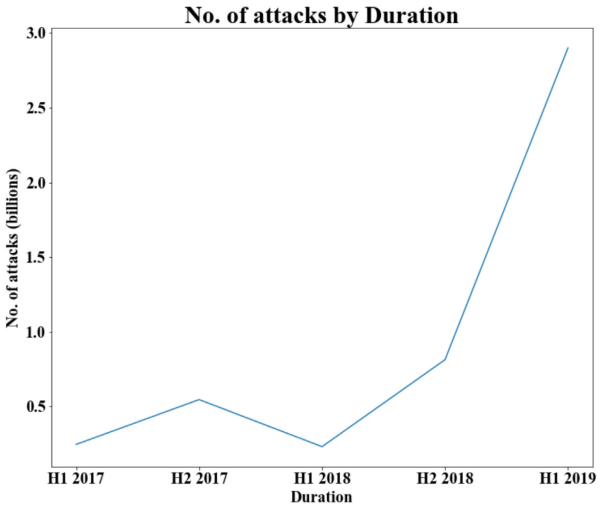


FIGURE 2. No. of cyber-attacks against IoT devices in each 6 months (one half) of the year [4].

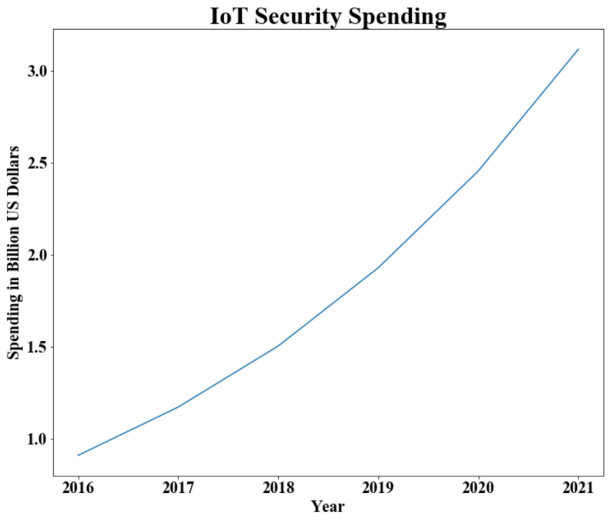


FIGURE 4. IoT security spending worldwide from 2016-2021 [7].

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 [3]. 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).

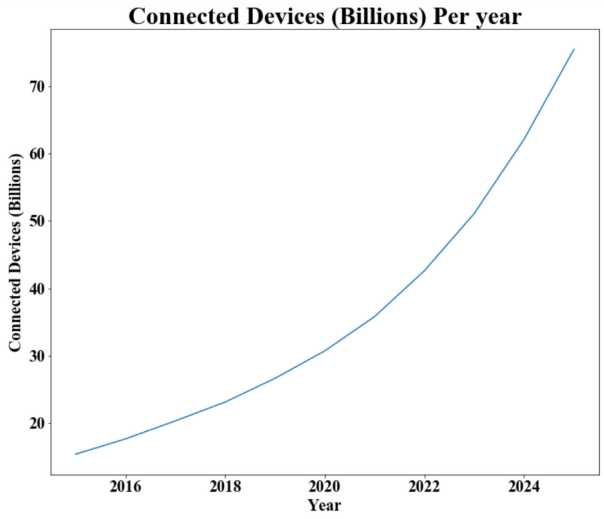


FIGURE 3. No. of IoT devices connected to the internet for each year [6].

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 [5].

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 UNSW-NB15 (2015) [8] and the Bot-IoT (2018) [9] are the most prominent and up-to-date date-sets in the domain. The Bot-IoT data-set is more sophisticated as it covers IoT devices which span over simulated as well as real data. Both data-sets have a common drawback which is the imbalance nature of data, which makes the prediction biased. These data-sets have been explored in detail in a number of previous works [10]–[16]. The researchers have applied both conventional machine learning techniques such as Decision Trees (DT), Naive Bayes (NB), Random Forest (RF), Support Vector Machine (SVM) etc. as well as deep learning algorithms such as CNN, RNN, LSTM etc. to analyze the effects of an intrusion in a large scale IoT network. However, there exist several challenging areas

such as finding common features between the UNSW-NB15 and the Bot-IoT data-sets so that both data-sets can simultaneously be used for training, handling the imbalance nature of the data-sets which results into biased classification, and training a deep learning model over both data-sets to detect DoS and DDoS attacks.

Researchers like Shafiq *et al.*, [14], employed several ML techniques of Decision Trees (DT), Naive Bayes (NB), Random Forest (RF) and Support Vector Machine (SVM) on the Bot-IoT data-set and used Pearson Moment Correlation and Area Under the Curve (AUC) metrics for feature selection and selected top 5 features while achieving promising results. Guizani *et al.*, [15] employed long short-term memory networks (LSTM) on the UNSW-NB15 data-set to analyze the effects of an intrusion in a large scale IoT network. Alkadi *et al.*, [16], used Bidirectional LSTM, where the data flows in both backward and forward directions, on the UNSW-NB15 and the Bot-IoT data-sets separately. In both of the aforementioned works, a relatively smaller subset of both data-sets is used for training which reduces accuracy of prediction due to weak learning and may lead to biased classification. Moreover, none of the existing works compared the two data-sets in order to find same or similar features so that the data from both of the data-sets can used to train supervised ML or deep learning models for meaningful, sophisticated and efficient intrusion detection.

In our previous work [17], we have trained two well-known supervised learning algorithms namely Random Forest and Support Vector Machine using the application and transport layer features from the UNSW-NB15 data-set. The binary and multi-class classification shows that intrusion detection is possible with higher precision using application and transport layer features. The work at hand is an extended work where we have used the Long Short Term Memory (LSTM) based un-supervised deep prediction model to classify IoT network traffic using two benchmark data-sets in the category of NIDS i.e., the UNSW-NB15 and BotIoT. These data-sets cover major attack scenarios based on the most realistic traffic. Since DoS attacks are a common threat to IoT devices, so we merged both data-sets to cover most of the DoS and DDoS scenarios. We have used flow and TCP features for classification because they are most relevant to IoT devices and require fewer features for training and classification, which significantly reduces the processing time. 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.

The remainder of the paper includes Section II covering related work, Section III discussing the proposed PB-DID architecture. Section IV covers the experiments conducted using the solution given in section III and their results. The discussion on results is presented in section IV, and their comparison with other related solutions is covered in section V, which is followed by a conclusion and future works in section VI.

II. RELATED LITERATURE

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) [18].

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 [19]. 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 [18], [20], [21]. According to Nexusguard the DDoS attacks in first quarter of 2020 has increased by 500% as compared to last quarter of 2019 [22].

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 [23]. 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 [23], [24]. It is therefore important that HIDS should be compliant with its devices to track various operating systems. Since investigation trails are used, the process is resource hungry and results in increased load on the host server which is a costly proposition [24], [25].

Furthermore, HIDS can still be compromised once an intruder exploits the host server. This implies, that the credibility of a host dependent detector can be compromised by internal security loopholes of a hosting server [23], [25]. Once an intruder exploits one such flaw, a security loophole might be challenging to identify. Also, through filling the system's audit files, a DoS attack could uncover a host and obstruct the operations of such HIDS [25], [26].

Network-based IDS (NIDS) tracks network traffic to detect mobile threats occurring via a network connection [26]. 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 [27].

There are numerous benefits of using a NIDS [28]. As compared to HIDS. NIDS does not affect network application's efficiency by performing surveillance of the network because from any incoming packet only a few details are required to be extracted. Whereas, in HIDS, operating system's assets are inspected [23]. Moreover, NIDS is compact since it tracks a specific portion of the network and is independent of the target's operating system.

Nonetheless, scalability is one of the key inconveniences in implementing a NIDS. As network gain higher data-rates, hackers may characterize them by employing spy and stealth assaults in low-footprint threats that can leverage loopholes [29], [30]. Network encryption makes it difficult in developing NIDS to gather cipher knowledge, while tunneling in IPsec protocols creates a new danger which allows IPv6 traffic to be encapsulated in the IPv4 and stream the data through non-cooperative devices raising threats of DoS and DDoS which can be alarming [31]. A sophisticated NIDS will cope with encryption throughout the connection by extricating general and empirical details on packets, as in it's length and size which are features related to flow [23], [32], however the payload of packet and extraction of attributes related to packets are still concealed. Through Deep Packet Inspection (DPI) such packets are analyzed for finding patterns, action or synthetic learning hypotheses and methodological approaches of such packets becomes the basis of their designation [32].

A. 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 [33], [34] unique risks against IoT products are discussed. Cervantes *et al.*, [33] 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 [34], Guo *et al.*, proposed to deter threats to the Bluetooth Low Energy (BLE) platform on three separate rates of power drainage.

In comparison to ML techniques and rules-based systems, a new IDS for the IoT setting has been proposed by Anthi *et al.*, [35]. Their goal is to anticipate malicious activities as well as to track their compromised IoT nodes from the DDoS threats within the network. Weka tool (open-source platform) [36] is used for evaluating the performance of classification through NB classifier. Nobakht *et al.*, [37] have also used Weka to investigate the precision of the forensic system for IoT botnets based on Logistic Regression (LR) and SVM.

In recent work, ML based techniques have out performing the classical approach of IDS through signature based detection especially in IoT environment. The reason is that in signature based detection the hackers can thwart the normal behaviour of traffic and also they cannot detect zero day attacks.

B. 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 [39] 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.

Zhou *et al.*, [41] suggested Deep Learning (DL) based intrusion detection to better identify cyber threats. Their framework employs stages of data collection, pre-processing and classification. Proposed model with a learning rate of 0.01 achieves highest accuracy of 96.3% on simulated data and outperforms linear regression, RF and k-nearest neighbours. Feng *et al.*, [42] define an plug and play ad-hoc network that uses captures packets and a DL technique to identify DoS and privacy threats. The algorithm suggests two fundamental classification methods in DL i.e. convolutional neural network (CNN) as well as LSTM to track threats by XSS and SQL. The analysts utilized the KDD CUP 99 data-set [43], that is divided into 30% testing and 70% training. In addition, 57% and 78% accuracy is achieved when XSS attacks with the DNN and CNN were detected, respectively.

Similarly, presented in [44] is a great illustration of DL and predictive learning to recognize intrusions into the network. By using sophisticated data acquisition and classification techniques, the analysis can classify a range of Network Intrusion types. Suggested method includes a discriminator and a generator as two main components. The discriminator is utilized to prevent accumulated data

from existing intrusion observations, whereas the generator produces increased intrusion knowledge. Kim *et al.*, [45] employed the KDD 1999 data-set to create a Deep Neural Network (DNN) for constantly shifting network threats. Two variables are used for the suggested intrusion detection paradigm: four hidden layers and 100 neurons in each layer. For training, activation ReLU function and for optimization, stochastic gradient descent method is utilized. Over 99% classification accuracy is obtained by the suggested method.

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 UNSW-NB15 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.

C. 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 [46] 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.

Over-sampling might be utilized for the production of a larger data-set for a small or minor class. The most uncomplicated oversampling technique is random over-sampling [47]. To increase the minor class size, data from the data-set is selected in a random pattern, which is then duplicated to make the minor class size bigger. Another method frequently used for over-sampling is the synthetic minority over-sampling technique (SMOTE) [48]. Synthetic results using SMOTE are generated using an algorithm of the k-nearest neighbors' samples among the minor class and one of its neighbors [49]. The feature vector for each feature's difference of feature's value and its neighboring feature's value is taken.

D. 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 [11], 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.

Churcher *et al.*, [12] compared various ML techniques such as KNN, SVM, DT, NB, RF, ANN and LR on the Bot-IoT data-set. They extracted only those features which either had no missing values or which were relevant. To create balanced data-set authors used 1.5 Million packets. Best classification accuracy of 99% was achieved by applying KNN. Improved Conditional Variational AutoEncoder (ICAVE) technique [13] was used to create balance in the data-set. Two data-sets were used for this purpose which include NSL-KDD (two variants) and the UNSW-NB15. They also applied DNN algorithm on the balanced data-sets and achieved accuracy score of 85.97% on NSL-KDD (KDDTest+), 75.43% on NSL-KDD (KDDTest-21) and 89.08% on the UNSW-NB15 data-set. Shafiq *et al.*, [14], employed several ML techniques of DT, NB, RF and SVM on the Bot-IoT data-set. They used Pearson Moment Correlation for features correlation with the class label and AUC metric to measure each feature's importance. By employing these techniques for feature selection, they used top 5 features only and achieved promising results of 99.99%, 97.5%, 99.98% and 97.8% by using DT, NB, RF and SVM respectively. Guizani *et al.*, [15] and Alkadi *et al.*, [16], deployed LSTM technique on the UNSW-NB15 and the Bot-IoT data-set. In [15], the authors achieved 70% accuracy by using LSTM on the UNSW-NB15 data-set. They took IDS to the next level by using it in countering the effects of an intrusion. In a large IoT network they created surveillance zones and locked the zone which is compromised. Only the compromised zone is then deployed with counters to recover from an intrusion. In this manner they reduced the resources utilization. In [16], BiLSTM is used on the UNSW-NB15 and the Bot-IoT data-sets separately. In BiLSTM, data flows in both backward and forward directions. They achieved 99.41% and 98.91% accuracy on the UNSW-NB15 and the Bot-IoT data-sets respectively.

It is a common practice to use feature importance or information gain to select the most contributing features, but there is a possibility that some other features might take the place of previous features if some more data is added. To address this problem, we need a few features that play the role of deciding features even when more data is added. Various studies have created a model based on a single data-set and used the same model to test/validate it on some other data-set, but there are almost none who compared features of two or more data-sets and combined them to create an entirely

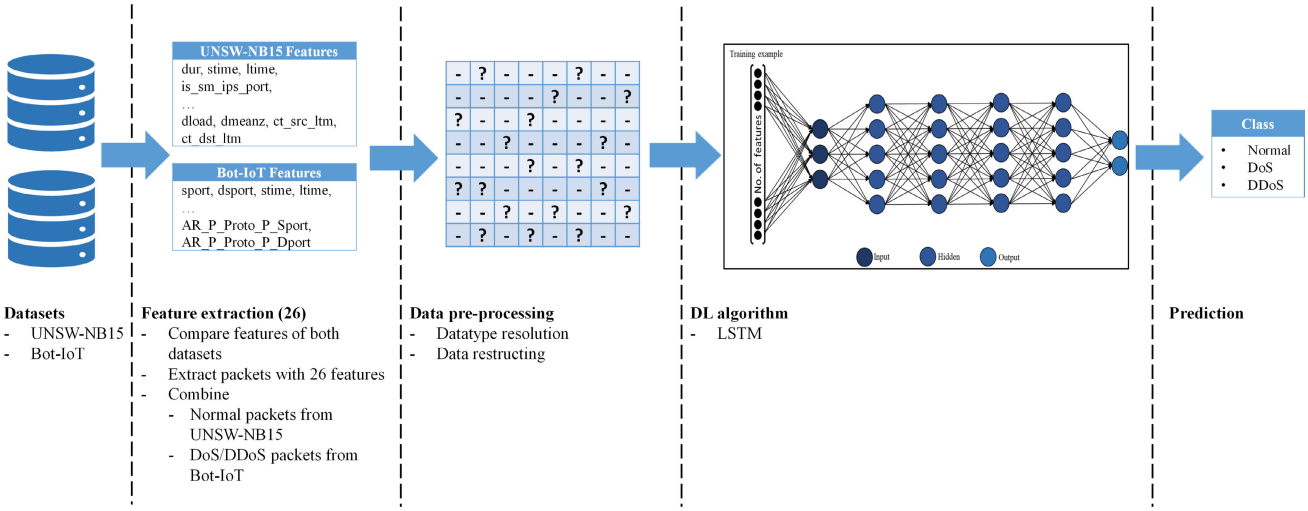


FIGURE 5. Proposed structure of PB-DID for attack classification using DL.

new data-set. In most studies, a small portion of the data-set is used for training and testing, which results in weak learning for some classes, whereas by using a full data-set, the model can learn better and will not miss out on any detail.

III. PROPOSED METHODOLOGY

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 data-sets, features selection, data pre-processing and selection and model training using un-supervised LSTM deep learning model.

A. 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.

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 non-anomalous 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 [8].

2) BOT-IoT DATA-SET

This data-set is the latest in the field. Koroniotis *et al.*, [9] 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

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**	

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 [9].

B. FEATURES COMPARISON

In the UNSW-NB15, there are 49 features including 48^{th} as a multi-class label and 49^{th} as a binary label. In the Bot-IoT 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.

C. FEATURES SELECTION

In the proposed PB-DID architecture, clusters of features in both data-sets are created according to flow, Domain

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		10	sport	
2	ltime		11	dport	
3	saddr		12	pkts	
4	daddr		13	bytes	
5	dur	Flow	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	TCP
			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	

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.

D. DATA PRE-PROCESSING

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

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., $N_IN_Conn_PSrcIP$, $N_IN_Conn_PDstIP$. These features will be meaningless if the IP addresses are entirely removed. Similarly, we have converted the 'proto' feature into an integer type.

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 [9] 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 data-set where ARP protocol is used.

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 data-set 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 *non-anomalous (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 data-set 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

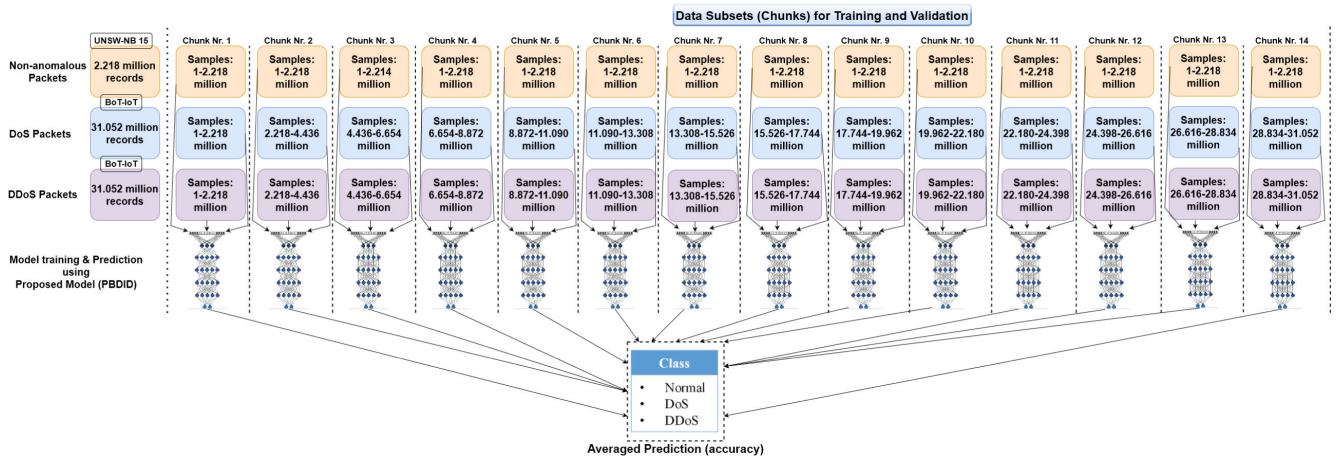


FIGURE 6. The process of merging the data from UNSW-NB15 and BoT-IoT data-sets is presented here. To mitigate over-fitting, the same number of samples for all three classes are kept in each data chunk. The proposed PB-DID model is trained separately on each data chunk and an averaged prediction (in %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

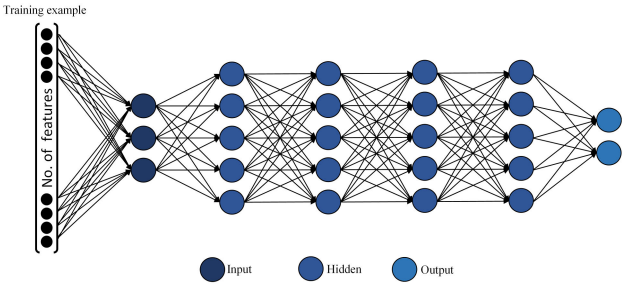


FIGURE 7. Basic structure of a DL algorithm.

in Figure 6, in each data chunk, the *non-anomalous* samples are repeated, whereas, the *DDoS* and *DoS* samples are always unique. The proposed PB-DID model is trained and validated with each data chunk separately and an averaged prediction (in % accuracy) overall 14 chunks is computed. The proposed data merging strategy utilizes 100% of the *non-anomalous* samples (2.218 million from UNSW-NB15), 80.65% of the *DDoS* samples (31.052 out of 38.5 million *DDoS* samples from BoT-IoT), 94.1% of the *DoS* samples (31.052 out of 33 million *DoS* samples from BoT-IoT), and a total of 64.322 million samples in all 14 data chunks are used for training and validation. The proposed strategy not only resolves the data imbalance issue, hence minimizing over-fitting, but also allows utilization of maximum number of samples from both UNSW-NB15 and BoT-IoT data-sets for training and validation of the PB-DID model.

We created batches of 128 packets of one category and give them one label. To fulfill this requirement, we kept the closest multiple of 128 which came out to be 2,218,240. In the final configuration, PB-DID has 17330 packets of batch size 128 of all three categories.

E. 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.

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.

F. PERFORMANCE METRICS

We used two metrics for measuring the performance of the proposed PB-DID model namely confusion matrices

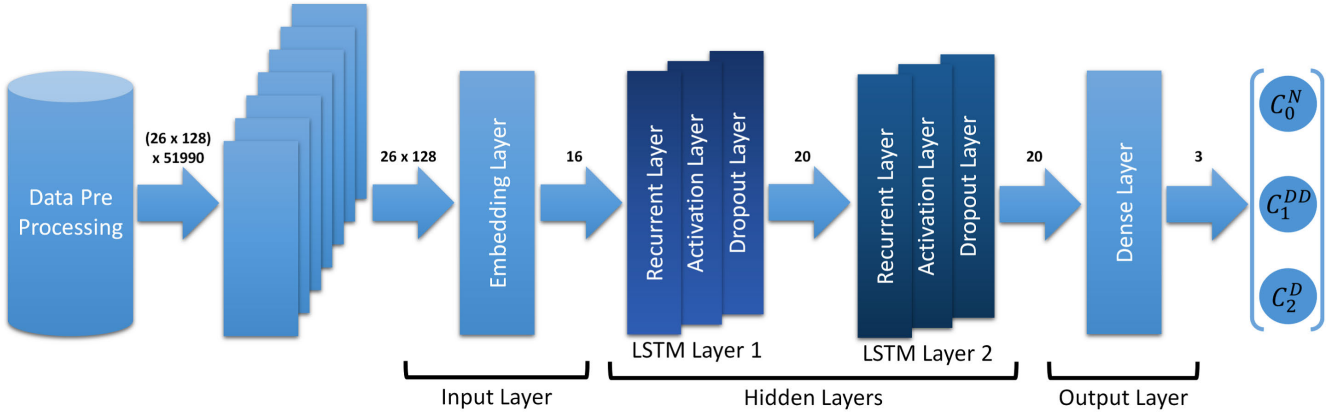


FIGURE 8. Structure of LSTM model showing input, hidden and output layers. In output layer C^i_j, i = 0, 1 or 2 is the label and j = non-anomalous (N) when i = 0, DDOS (DD) when i = 1 and DOS (D) when i = 2 is the class.

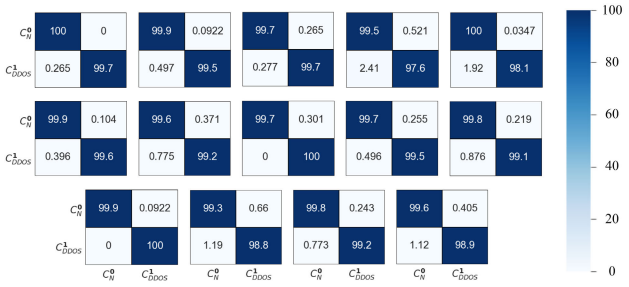


FIGURE 9. Confusion matrices of all 14 data chunks, C^i_j, i = 0 or 1 is the label and j = non-anomalous (N) when i = 0 and DDOS (DD) when i = 1 is the class.

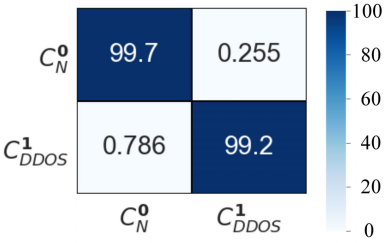


FIGURE 10. Confusion matrix calculated by averaging all 14 data chunk confusion matrices involving non-anomalous and DDoS packets.

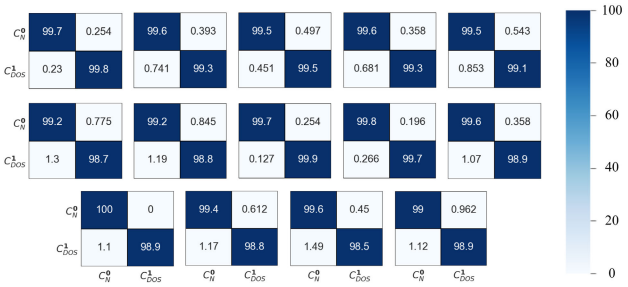


FIGURE 11. Confusion matrices of all 14 data chunks, C^i_j, i = 0 or 1 is the label and j = non-anomalous (N) when i = 0 and DOS (D) when i = 1 is the class.

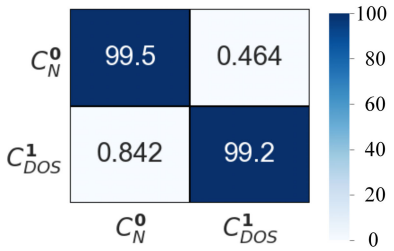


FIGURE 12. Confusion matrix calculated by averaging all 14 data chunk confusion matrices involving non-anomalous and DoS packets.

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.

IV. 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 [51] using tensorflow as back-end and the language used is Python3.

A. CLASSIFICATION RESULTS

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

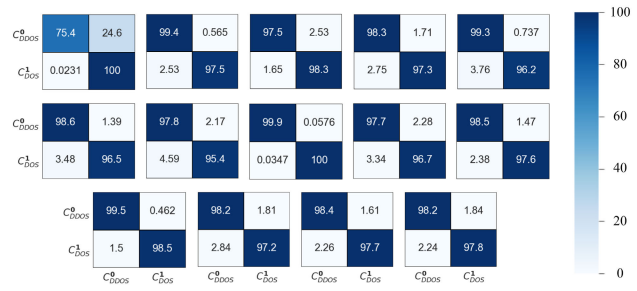


FIGURE 13. Confusion matrices of all 14 data chunks, C^i_j, i = 0 or 1 is the label and j = DDOS (DD) when i = 0 and DOS (D) when i = 1 is the class.

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**

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

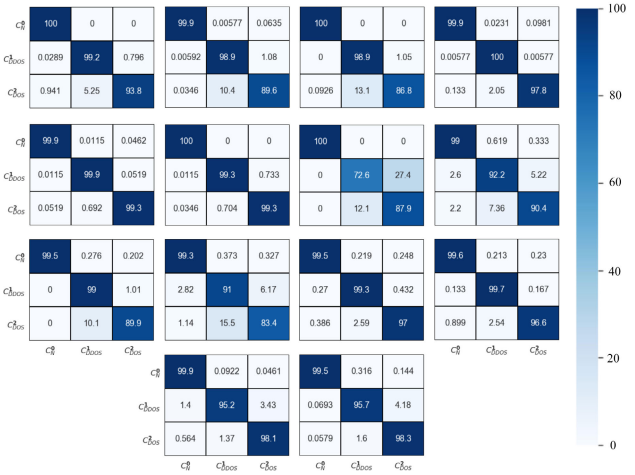


FIGURE 15. Confusion matrices of all 14 data chunks, C^i_j, i = 0, 1 or 2 is the label and j = non-anomalous (N) when i = 0, DDOS when i = 1 and DOS when i = 2 is the class.

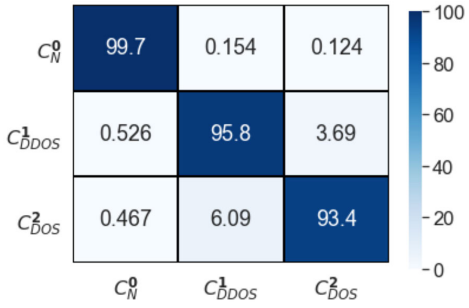


FIGURE 16. Confusion matrix calculated by averaging all 14 data chunk confusion matrices.

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.

B. 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.

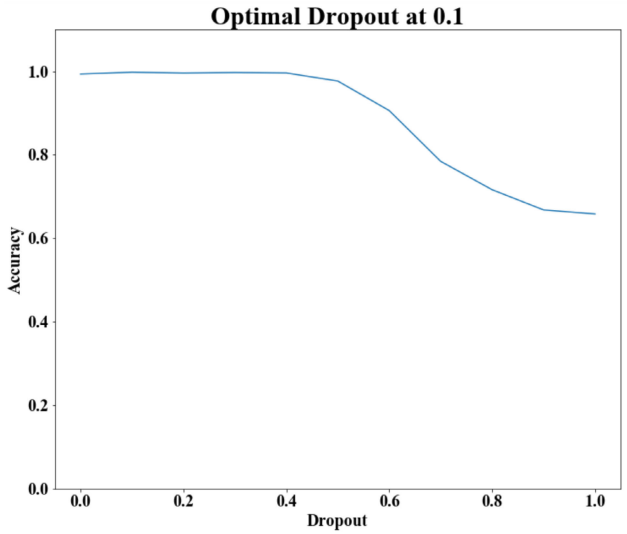


FIGURE 17. Optimal dropout graph over values of [0,1] where best accuracy of 99.79% is achieved at ***0.1***.

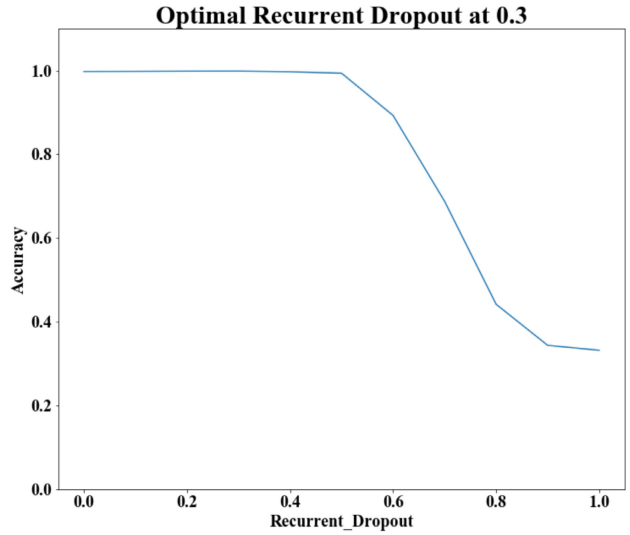


FIGURE 18. Optimal recurrent dropout graph over [0,1] and best accuracy 99.885% at ***0.3*** is achieved.

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***	-	-	-	-

V. DISCUSSION AND COMPARISON

In binary classification, our proposed LSTM based un-supervised deep learning model (PB-DID) achieves classification accuracies of above 99% for non-anomalous vs. anomalous packets (non-anomalous vs. DDOS and non-anomalous vs. DOS). For binary classification between DDOS and DOS i.e. correct classification of attack type,

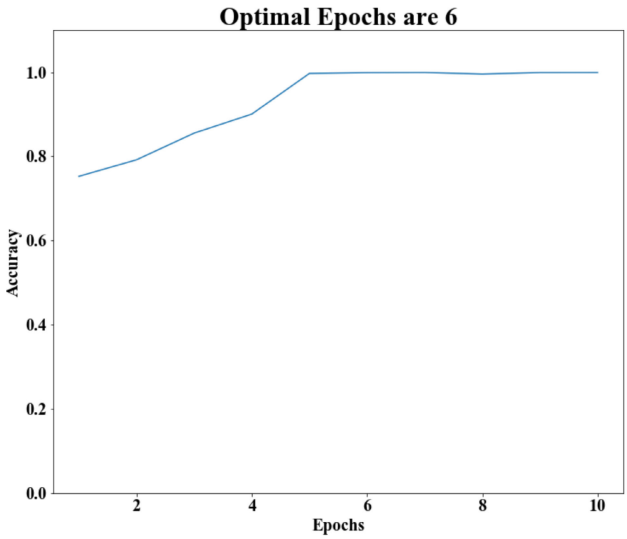


FIGURE 19. Optimal epochs = ***6*** after which the graph becomes smooth in accuracy.

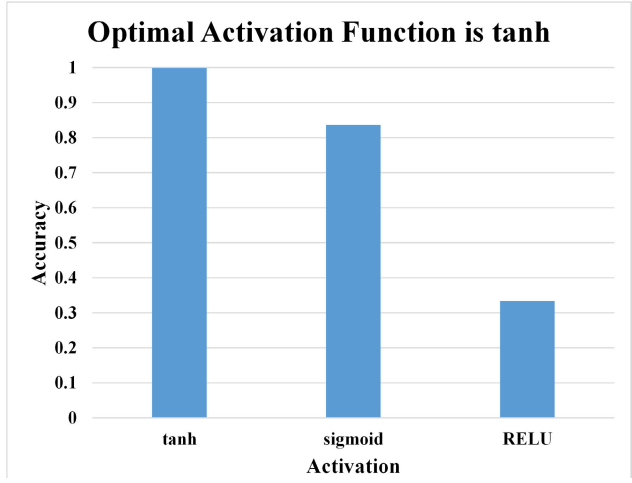


FIGURE 20. Optimal activation function where ***tanh*** gives best accuracy of 99.885%.

the classification accuracies remain above 96%. In case of ternary classification (non-anomalous vs. DDOS vs. DOS packets), the overall classification accuracies remain above 96% where 99.7% of the packets are correctly classified as non-anomalous. Most of the confusions are seen between DOS and DDOS packets where 6% of the DOS packets are misclassified as DDOS packets. Major reason for this is the minute difference between DDoS and DoS attack, which comes from their origin. If we observe the packet parameters in the data-set, the difference between both categories is the source IP address. This is because of the inherited natures of DOS and DDOS attacks where in DoS a single machine is sending huge traffic to the destination and in case of DDOS, multiple sources with different IP addresses send massive traffic to a single destination.

As mentioned in the Introduction Section, this work is an extension of our previous work [17] where we used

TABLE 7. Comparison of PB-DID technique with other related techniques. Last column shows the no. of classes classified.

Technique	Algorithm	data-set(s)	data-set comparison	No. of features	Amount of data-set used	Accuracy(%)	No. of classes classified
Koroniotis et al., [9]	SVM RNN LSTM	Bot-IoT	No	10 best and full	100%	88.3 / 99.9 99.7 / 97.9 99.7 / 98	11
Shafiq et al., [14]	DT NB RF SVM	Bot-IoT	No	5 best	NA	99.99 97.5 99.98 97.8	8
Guizani et al., [15]	LSTM	UNSW-NB15	No	Full	10%	70	Binary
Alkadi et al., [16]	BiLSTM	UNSW-NB15 Bot-IoT	No	Full	14000 pkts	99.41 98.91	16
Khraisat et al., [52]	Hybrid IDS	Bot-IoT	No	13	18,473 pkts	99.97	5
Ibitoye et al., [53]	FNN SNN	Bot-IoT	No	10 best	20%	95 91	5
Larriva et al., [10]	MLP	NSL-KDD, UGR16 and UNSW-NB15 and	No	Full	NSL-KDD (NA), 17% UGR16 and 10% UNSW-NB15	99.7 99.3 99.2	NA
Popoola et al., [11]	DRNN	Bot-IoT	No	37	5%	100	11
Churcher et al., [12]	KNN SVM DT NB RF ANN LR	Bot-IoT	No	19	2%	99 79 96 94 95 97 74	11
Yang et al., [13]	DNN	NSL-KDD (KDDTest+) NSL-KDD (KDDTest-21) UNSW-NB15	No	NA	47736 pkts of NSL-KDD (KDDTest+), 37042 pkts of NSL-KDD (KDDTest-21), and 10% UNSW-NB15	85.97 75.43 89.08	5 10
PB-DID (Proposed method)	**LSTM**	**Bot-IoT and UNSW-NB15**	**Yes**	**26**	**96% Bot-IoT and 87% UNSW-NB15**	**96.3**	**3**

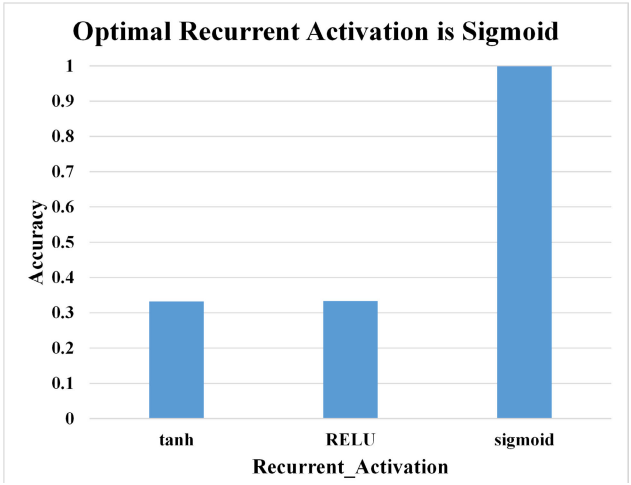


FIGURE 21. Optimal recurrent activation function which shows ***sigmoid*** gives best accuracy of 99.885%.

approximately 60% of the UNSW-NB15 data-set for training and validation using RF and SVM. In this work, we have used all (2.2 Million) non-anomalous packets from the UNSW-NB15 data-set and all DoS and DDoS packets from the Bot-IoT data-set. In order to make both data-sets compatible for experimentation, we needed to calculate few features of the UNSW-NB15 which were missing in the Bot-IoT and vice versa. Through our experiments on these data-sets, we conclude that these data-sets can be combined, compared, experimented and analyzed together.

A. COMPARISON WITH EXISTING APPROACHES

We compared our results with other studies which have used the Bot-IoT and/or the UNSW-NB15 data-sets. All of the existing studies have used only one of the two data-sets i.e. either the Bot-IoT or the UNSW-NB15 for training and validation. In this work, we have compared two data-sets, created a new custom data-set and proposed an LSTM based un-supervised PB-DID architecture. Larriva *et al.*, [10] used various pre-processing techniques and applied them on NSL-KDD, UGR16, and the UNSW-NB15 data-sets separately. The data-sets were used to train MLP and achieved accuracies of 99.7%, 99.3%, and 99.2% for NSL-KDD, UGR16, and the UNSW-NB15 respectively. The authors did not compare/merged the data-sets and the number of classes used for classification is also unclear. Popoola *et al.*, [11] applied DRNN on a normalized and balanced data-set (by using SMOTE) and achieved a 100% accuracy score against 11 classes. However, they only used 5% of the Bot-IoT data-set. Authors in [12] used various ML algorithms on approximately 2% of the data-set, with 19 features and achieved the best accuracy score of 99% by using KNN. However, since a very small portion of the data was used for training so the reported results can be considered statistically insignificant. Yang *et al.*, [13] have balanced the data-set by using the ICAVE technique and applied DNN on three data-sets separately but no comparison/merger of data-sets was involved. Koroniotis *et al.*, [9] used full and ten best features of the Bot-IoT data-set and applied three ML and DL algorithms i.e., SVM, recurrent neural network (RNN), and LSTM. They achieved the best accuracy of 99.9% by using SVM with full features. Shafiq *et al.*, [14] also used the Bot-IoT data-set for training and validation. They used the AUC metric to select the top 5 features and applied ML techniques and achieved 99.99% accuracy with DT. Guizani *et al.*, [15] used an advanced form of RNN, LSTM on the UNSW-NB15 data-set, and achieved 70% accuracy. Similarly, authors in [16] used the UNSW-NB15 and the

Bot-IoT data-sets independently and applied the BiLSTM technique. They classified 16 classes by comparing the classes of data-sets and identifying sub-categories in both data-sets.

Khraisat *et al.*, [52] used the technique of hybrid IDS on the Bot-IoT data-set and kept only the top thirteen features that have information gain threshold above 0.2. They achieved an accuracy of 99.97%. Ibitoye *et al.*, [53] used feed forward ANN (FNN) and self normalized neural network (SNN) on the Bot-IoT data-set with top ten features and achieved an accuracy of 95% with FNN and 91% with SNN. The summary of the comparison between the existing approaches and the proposed approach is given in Table 7.

The proposed PB-DID architecture uses all of the data from the Bot-IoT and the UNSW-NB15 data-sets (by merging them to create a single customized data-set) and trains the LSTM based un-supervised deep model using 26 features. Data imbalance issue was also resolved during the data-sets merging. The average classification accuracy remains above 96% for binary and multi-class classifications. The results show that the proposed solution can be applied in real IoT scenarios and it can detect intrusions of different types using flow and TCP features with higher accuracy.

VI. 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 non-anomalous, 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.

REFERENCES

[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-on-iot-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-iot-devices-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-rampant-reportclaims/#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-connected-devices-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 UNSW-NB15 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.

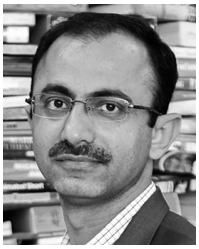
[19] S. Pontarelli, G. Bianchi, and S. Teofili, "Traffic-aware design of a high-speed FPGA network intrusion detection system," *IEEE Trans. Comput.*, vol. 62, no. 11, pp. 2322–2334, Nov. 2013.

[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.
[26] I. Corona, G. Giacinto, and F. Roli, "Adversarial attacks against intrusion detection systems: Taxonomy, solutions and open issues," *Inf. Sci.*, vol. 239, pp. 201–225, Aug. 2013.
[27] A. Milenkoski, M. Vieira, S. Kounev, A. Avritzer, and B. D. Payne, "Evaluating computer intrusion detection systems: A survey of common practices," *ACM Comput. Surv.*, vol. 48, no. 1, pp. 1–41, Sep. 2015.
[28] J. Peng, K.-K. R. Choo, and H. Ashman, "User profiling in intrusion detection: A review," *J. Netw. Comput. Appl.*, vol. 72, pp. 14–27, Sep. 2016.
[29] G. Creech, "Developing a high-accuracy cross platform host-based intrusion detection system capable of reliably detecting zero-day attacks," Ph.D. dissertation, Univ. New South Wales, Canberra, NSW, Australia, 2014.
[30] G. Creech and J. Hu, "A semantic approach to host-based intrusion detection systems using contiguousand discontiguous system call patterns," *IEEE Trans. Comput.*, vol. 63, no. 4, pp. 807–819, Apr. 2014.
[31] M. Kumar, M. Hanumanthappa, and T. S. Kumar, "Encrypted traffic and IPsec challenges for intrusion detection system," in *Proc. Int. Conf. Adv. Comput.* New Delhi, India: Springer, 2013, pp. 721–727.
[32] R. Bar-Yanai, M. Langberg, D. Peleg, and L. Roditty, "Realtime classification for encrypted traffic," in *Proc. Int. Symp. Exp. Algorithms*. Berlin, Germany: Springer, 2010, pp. 373–385.
[33] C. Cervantes, D. Poplade, M. Nogueira, and A. Santos, "Detection of sinkhole attacks for supporting secure routing on 6LoWPAN for Internet of Things," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manage. (IM)*, May 2015, pp. 606–611.
[34] Z. Guo, I. G. Harris, Y. Jiang, and L.-F. Tsaur, "An efficient approach to prevent battery exhaustion attack on BLE-based mesh networks," in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, Jan. 2017, pp. 1–5.
[35] E. Anthi, L. Williams, and P. Burnap, "Pulse: An adaptive intrusion detection for the Internet of Things," 2018.
[36] *Weka 3: Machine Learning Software in Java*. Accessed: Jan. 2, 2021. [Online]. Available: https://www.cs.waikato.ac.nz/ml/weka/
[37] M. Nobakht, V. Sivaraman, and R. Boreli, "A host-based intrusion detection and mitigation framework for smart home IoT using OpenFlow," in *Proc. 11th Int. Conf. Availability, Rel. Secur. (ARES)*, Aug. 2016, pp. 147–156.
[38] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho, "Deep learning approach for network intrusion detection in software defined networking," in *Proc. Int. Conf. Wireless Netw. Mobile Commun. (WINCOM)*, Oct. 2016, pp. 258–263.
[39] University of New Brunswick. (2009). *NSL-KDD Dataset*. [Online]. Available: https://www.unb.ca/cic/datasets/nsl.html
[40] S. Potluri and C. Diedrich, "Accelerated deep neural networks for enhanced intrusion detection system," in *Proc. IEEE 21st Int. Conf. Emerg. Technol. Factory Automat. (ETFA)*, Sep. 2016, pp. 1–8.
[41] L. Zhou, X. Ouyang, H. Ying, L. Han, Y. Cheng, and T. Zhang, "Cyber-attack classification in smart grid via deep neural network," in *Proc. 2nd Int. Conf. Comput. Sci. Appl. Eng. (CSAE)*, 2018, pp. 1–5.
[42] F. Feng, X. Liu, B. Yong, R. Zhou, and Q. Zhou, "Anomaly detection in ad-hoc networks based on deep learning model: A plug and play device," *Ad Hoc Netw.*, vol. 84, pp. 82–89, Mar. 2019.
[43] Irvine University of California. (Oct. 1999). *KDD Cup 1999 Data*. [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
[44] H. Zhang, X. Yu, P. Ren, C. Luo, and G. Min, "Deep adversarial learning in intrusion detection: A data augmentation enhanced framework," 2019, *arXiv:1901.07949*.
[45] J. Kim, N. Shin, S. Y. Jo, and S. H. Kim, "Method of intrusion detection using deep neural network," in *Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp)*, Feb. 2017, pp. 313–316.
[46] N. Japkowicz, "The class imbalance problem: Significance and strategies," in *Proc. Int. Conf. Artif. Intell.*, 2000, pp. 1–7.
[47] S. J. Dattagupta, "A performance comparison of oversampling methods for data generation in imbalanced learning tasks," Ph.D. dissertation, NOVA Inf. Manage. School, Lisboa, Portugal, 2018.
[48] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, no. 28, pp. 321–357, Jun. 2006.
[49] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *Amer. Statist.*, vol. 46, no. 3, pp. 175–185, 1992. [Online]. Available: http://www.jstor.org/stable/2685209
[50] M. Hargrave. (Apr. 2019). *Deep Learning*. [Online]. Available: https://www.investopedia.com/terms/d/deep-learning.asp
[51] E. Bisong, *Google Colaboratory*. Berkeley, CA, USA: Apress, 2019, pp. 59–64, doi: 10.1007/978-1-4842-4470-8_7.
[52] A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman, and A. Alazab, "A novel ensemble of hybrid intrusion detection system for detecting Internet of Things attacks," *Electronics*, vol. 8, no. 11, p. 1210, Oct. 2019.
[53] O. Ibitoye, O. Shafiq, and A. Matrawy, "Analyzing adversarial attacks against deep learning for intrusion detection in IoT networks," 2019, *arXiv:1905.05137*.

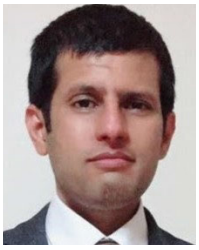


MUHAMMAD ZEESHAN (Senior Member, IEEE) received the Ph.D. degree in information technology from the National University of Science and Technology (NUST), Islamabad, Pakistan, in 2016. Since 2016, he has been working as an Assistant Professor with the Department of Computing, School of Electrical Engineering and Computer Science, NUST. His research interests include wireless networks, networks data analysis, network security, and the Internet of Things.

QAISER RIAZ (Senior Member, IEEE) received the M.S. degree in autonomous systems from the Bonn-Rhein-Sieg University of Applied Sciences, Sankt Augustin, Germany, in 2011, and the Ph.D. (Dr.Rer.Nat.) degree in computer science from the University of Bonn, Germany, in 2016.

Since 2016, he has been an Assistant Professor with the Department of Computing, School of Electrical Engineering and Computer Science, National University of Sciences and Technology, Islamabad, Pakistan. His research interests include motion capturing, human motion analysis and synthesis using low-cost sensors, character animation, and machine and deep learning.

MUHAMMAD AHMAD BILAL was born in Pakistan. He received the B.S. degree (Hons.) in software engineering from the Military College of Signals, NUST, Islamabad, Pakistan, in 2016. He is currently pursuing the M.S. degree in computer science with the School of Electrical Engineering and Computer Science (SEECS), NUST. His current research interests include machine learning, the IoT Security, the IoT anomaly and intrusion traffic classification, and network traffic classification.



MUHAMMAD K. SHAHZAD received the B.E. degree in information technology from The University of Lahore, in 2004, the M.S. degree in information technology from the National University of Science and Technology (NUST), Islamabad, Pakistan, in 2007, and the Ph.D. degree in computer engineering from Sungkyunkwan University (SKKU), Suwon, South Korea. He joined the MONET Laboratory, SKKU, as a Postdoctoral Researcher, after graduation, in 2016. Later, he worked as an Assistant Professor with the Electrical Energy Engineering Department, Keimyung University, Daegu, South Korea, from 2017 to 2018. He is currently working as an Assistant Professor with the Department of Computing, School of Electrical Engineering and Computer Science, NUST. His responsibilities include a PG CS Co-ordinator, a PG Data Science Co-ordinator, a PG CS Interviews, and also worked as a QS Ranking Rubrics lead. His research interests include artificial intelligence, data science, and graph theory.

HAJIRA JABEEN (Member, IEEE) received the Ph.D. degree in computer Science from the National University of Computing and Emerging Sciences, Islamabad, Pakistan. She is currently working as a Data Science Expert with the CEPLAS—Cluster of Excellence on Plant Sciences, University of Cologne, Germany. Her research interests include big data artificial intelligence, evolutionary computation, semantic web, data mining, and machine learning.

SYED ALI HAIDER received the bachelor's and master's degrees in computer science from the SZABIST, Karachi, Pakistan, the second master's degree in digital multimedia and communication systems from the University of Strathclyde, Glasgow, U.K., in 2006, and the Ph.D. degree in optical science and engineering from the University of North Carolina at Charlotte, USA, in 2012. He is currently working as an Assistant Professor with the Department of Computer and Information Science, State University of New York at Fredonia, USA. His research interests include software-defined networks, student progress monitoring, and broad spectrum applications of machine learning.

AZIZUR RAHIM received the M.S. degree in electrical engineering from the COMSATS, Islamabad, Pakistan, and the Ph.D. degree in computer application technologies from The Alpha Laboratory, School of Software, Dalian University of Technology, Dalian, China. He is currently working as an Assistant Professor with the School of Electrical Engineering and Computer Science (SEECS), National University of Science and Technology (NUST), Islamabad. His research interests include mobile and social computing, ad-hoc networks, VANETs, mobile social networks, and vehicular social networks.