Journal of Harbin Engineering University Vol 44 No. 8
ISSN: 1006-7043 August 2023

A Machine Learning Based Method for Improvement of DDoS Attack Defense in IoT Networks

Debi Prasad Mishra[1], Nibedita Adhikari[2], Laxminarayan Dash[3]

[1,2] Department of Computer Science and Application, Utkal University, Bhubaneswar, Odisha
[3]Department of Computer Science and Engineering, National Institute of Science and Technology, Berhampur, Odisha
[1]dpmishra.07@gmail.com
[2]nibedita.cs@utkaluniversity.ac.in
[3]laxminarayandash20@gmail.com

Abstract

The "Internet of Things" (IoT) has been a factor of major impact in facilitating connectivity among a wide range of devices and machines. It offers enhanced user experiences, better management, control, and analysis while facilitating informed decision-making. However, the low processing and storage capacity makes them vulnerable to a variety of network threats. "The Distributed Denial-of-Service (DDoS)" has been perceived as a significant issue in the realm of IoT services, causing disruptions and compromising data flow within and outside of IoT networks. Although there has been several research in this field, accuracy, and efficiency of classification emerge as definitive factors to design a sustainable defense mechanism. Most of the studies are concerned only with attacks targeted to only a specific type of network protocol. The previous research work has been carried out with limited available datasets. The current paper aims to achieve sustainable accuracy in application scenarios using hybrid network traffic which has been missed in all of the earlier research works. In addition to the above, this research uses a custom data set and also facilitates faster detection and blocking without incurring external networking overheads at the edge points of the network.

Keywords: Internet of Things, Denial of Service attack, network threats, multi-protocol attacks, processing overheads, accuracy.

1. Introduction

The Internet of Things (IoT) has emerged as one of the widely accepted solutions for implementing smart devices due to its capability to provide support for low-power lossy networks [1]. It depicts a network architecture where a multitude of devices, such as sensors, are interconnected through either a private or public community. The functionalities of IoT devices can be remotely managed irrespective of their geographical locations. Various information and data belonging to such devices are shared over a communication channel using mutually agreed-upon protocols laid down by the standardizing bodies.

The Denial of Service (DoS) is an attack that prevents user requests for genuine network services. This scenario arises when the network path gets flooded by a large number of illegitimate requests. In other words, such types of attacks can be referred to as an attack on the availability of resources [2].

A DDoS attack is a scenario where cyber-criminals flood the network with unnecessary traffic or false traffic which leads to a situation where the network resources become inaccessible at a certain instance of time. The attackers take advantage of a huge number of botnets or zombies for the attack purpose. The IoT botnet attack is designed in a master-slave structure in which the attackers gain access to a small number of master devices and these master devices control a large number of slave devices. The slave devices are nothing but compromised devices employing malicious codes. These attacks tend to take down the whole IoT network for a certain period which may account for

the loss of both data and financial revenue for organizations dependent on such facilities. DDoS attacks block all the further incoming traffic which allows other users to get access to the IoT devices.
The rise of DDoS attacks in IoT networks is attributed to their limited memory capacity and low operational bandwidth, making them vulnerable to such attacks. The IoT devices are connected to personal computers, hand-held devices, or even cloud services which further is an added concern for data leakage or service breakdown [14]. In the event of a botnet attack, botnets consume all available memory and the available bandwidth. This scenario further facilitates the attackers to corrupt a network in a comparatively less amount of time. In addition to the vulnerabilities mentioned earlier, there are even more inherent factors like weak password configuration, open access admin management, or weak security configuration that further escalate the security concern and make it easy for attackers to initiate different attacks.
There are even certain services like Shodan [13], and Censys [12] which can provide crucial information about promiscuous IoT devices to attackers [15]. In this paper, a mechanism for enhancing performance is presented, aiming to effectively mitigate DDoS attacks in the IoT environment. This mechanism takes into consideration the utilization of multiple types of communication protocols by IoT devices. This work also tries to eliminate the issues of misclassification of attack packets that can arise as explained in [1] while at the same time improving the efficiency of classification in a network that uses a variety of network protocols. To achieve the results, the packets are captured in a network comprising different types of network traffic and they are further subjected to processes like data pre-processing and feature selection before utilizing a suitable classifier.
The organization of this work has been done in different sections as per the following descriptions. In Section 2 some background idea about DDoS attacks and botnet attacks has been provided. In Section 3, a study of different research done toward the mitigation of DDoS attacks has been presented. Section 5 represents the proposed methodology along with different stages of intermediate work. The subsequent section presents discussion of the results of the proposed work along with evaluation of performance metrics. In the concluding remarks, the paper describes generalized remarks regarding the functionality of the proposed approach along with a brief discussion on future dimensions.

2. Background

The description of a DDoS attack has been provided in the preceding section. The DDoS involves a large number of computing nodes being centrally exploited by a central handler. This process is mostly automated which leverages the modern age computing facilities. These facilities enable attackers to launch attacks in larger volumes. A typical scenario has been shown in the Figure 1 below.

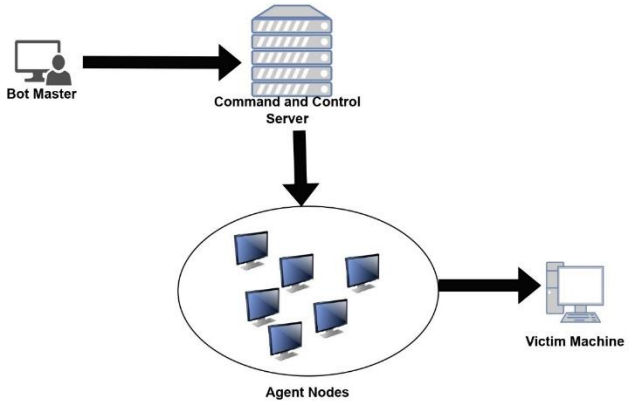

Figure 1: The Scenario of a DDoS Attack

The "Internet of Things" (IoT) system consists of a network of devices that may be made up of embedded hardware, software, and various protocols to provide connectivity and message exchange services using the Internet. The application scenarios of IoT devices may range from low-level requirements to mission-critical applications which would require a high amount of availability. A typical DDoS attack on a network is launched with the help of botnets or network zombies that can produce enormous amounts of traffic. Botnets are capable of creating a network of computers controlled by a single master node which takes control of the connected nodes with

the help of malware or some malicious codes [1]. The Figure 2 depicts various stages of IoT botnets which a botnet passes through during the course of attack formation.

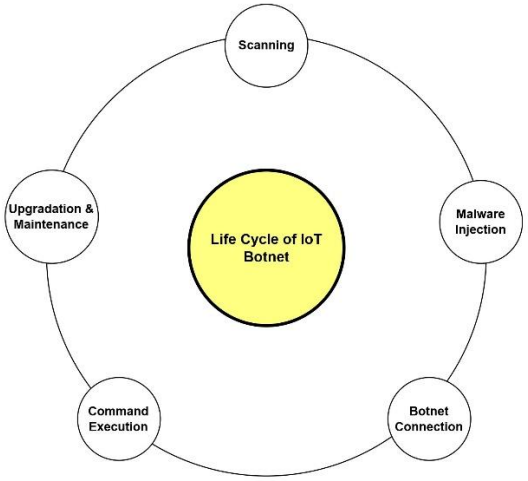

Figure. 2 Life cycle of IoT Botnets

Based on the types of attacks they employ, the existing methods of DDoS attacks can be broadly categorized into three groups: volume-based attacks, protocol attacks, and application layer attacks [11].

While Volumetric DDoS attacks primarily aim to induce congestion, they can also serve as a cover for more sophisticated and targeted DDoS operations. In these situations, potential attackers seek to disrupt operations and divert attention by employing various tactics. They may monitor the situation closely and rapidly modify their attacks to evade static defensive measures. Referred to as "Trojan Horse" DDoS attacks, these malicious activities aim to disable a firewall or intrusion prevention system. By doing so, attackers gain unauthorized access to a network, enabling them to deploy malware and ultimately exfiltrate valuable data.

Unlike volume-based attacks, protocol attacks focus on exploiting server resources instead of consuming bandwidth. These attacks specifically target "intermediate communication equipment," referring to components such as firewalls and load balancers that act as intermediaries between the server and the website. Hackers overwhelm websites and servers by sending fraudulent protocol requests, depleting the available resources. The effectiveness of these attacks is measured in terms of the number of packets transmitted per second.

Application layer threats typically require fewer resources compared to protocol and volume-based attacks. Network service degradation can occur as a result of these DDoS attacks, wherein a flood of seemingly legitimate requests, mimicking user traffic patterns, overwhelms the system. Application layer attacks, on the other hand, can remain undetected since they specifically target specific application packets. The severity of these attacks is measured in requests per second.

The IoT devices possess vulnerabilities that render them susceptible to various types of network attacks, including DDoS attacks. Some of these include factors like hardware limitations, poor network policy, improper data security management, and sometimes bugs in software programs may also lead to networks being compromised.

3. Related Research

Several research has been carried out and several solutions have been proposed for effective protection from DDoS attacks in operational IoT networks. The techniques for such attack detection can be classified into two different categories, the graph-based methodology and the flow-based methodology [5]. In the first kind of methodology, the communication patterns of nodes of a network are analyzed to monitor their behavioral patterns and anomalies in their behavior whereas the second methodology involves analysis of network traffic using machine learning algorithms by making a correlation of traffic patterns [10]. The work in [28] uses particle swarm and grey wolf optimization methods for data collection. Similarly, the work of [29] researches machine learning-based intrusion detection in home automation systems. This emphasizes that the default configurations of the devices and even software bugs can sometimes be attributed to incidences of intrusion in IoT devices.

The work published in [4] uses conventional classifiers against NSL-KDD, IoTDevNet, DS2OS,

IoTID20, and IoT Botnet datasets to evaluate the performance of deep learning-based networks. The limitations of this work lie in high-traffic scenarios where there are possibilities of miscalculation.

A framework for DDoS attack mitigation has been proposed by Adat and Gupta in [3] that can be helpful in a network of devices having lesser storage and processing capabilities. A graph-based was proposed in the work of Nguen et.al [9] which utilizes both a convolutional neural network (CNN) and a deep learning model for the detection of attack patterns. Several studies ([10], [11], [22], [26]) have been conducted to investigate the creation of a hybrid defense mechanism that combines both the flow and graph-based approaches. The works shown in [3], [9-11], [22], [26] fail to protect synthetic traffic generators.

Lima et.al [7] presented an intelligent DDoS detection method that can effectively classify network traffic and notify the network admin about attacks using a cloud platform. The mechanism of attack detection comprises attack signatures or attack databases and a machine learning algorithm. It is obvious that at one point, the security mechanism may fall short of protection in case of heavy traffic or missing attack signatures in the attack database. Sriram et.al [8] presented a deep-learning-based attack detection method using flow-based metrics for protection from botnet attacks. The machine learning model in [8] uses SVM which accounts for a longer learning period and hence, it can be assumed that the projected protection mechanism may not be sufficient. It may take more time to identify a potential attack.

Parra et al. [15] presented a distributed framework that utilizes deep learning methodology over cloud infrastructure. This framework comprises two different machine learning models: a distributed neural network model for protection from phishing attacks and a second model comprising recurrent neural networks (RNN) and LSTM to withstand botnet attacks. Pektacs and Acarman [16] used network traffic to carry out botnet identification using deep learning and classify network traffic into two categories namely, genuine and malicious. The DDoS attack defense methodology developed by Ahmed et al. [17] utilized a deep learning mechanism in its design. The works in [15-17] have raised concerns over security against synthetic network traffic.

Maeda et al. [18] used software-defined networks (SDN) and deep learning mechanisms for the effective classification of attack patterns. They used data collected from the network for the training of neural networks. Although this work provides a substantially good amount of protection, it fails to isolate the infected node(s) from the network. Meidan et al. [19] employed auto-encoders as a means of detecting botnets in IoT networks. The authors injected IoT devices with some well-known DDoS attacks and then trained autoencoders using both genuine and malicious traffic separately. But this work also fails to provide results in cases of scaled up data traffic. Nasif et. al [6] proposed a monitor agent-based methodology to monitor and protect networks from DDoS attacks. The solutions in [23], [24] and [25] have tried to develop a lightweight defense mechanism in addition to various techniques of machine learning but they are limited to only specific types of attacks or network protocols.

The Table 2 presents a comprehensive overview of the research conducted so far in the same context. Upon reviewing previous research, it becomes evident that the majority of the conducted studies have relied on machine learning algorithms that employ a network traffic through flow-based approach to detect DDoS attacks. It can also be summarized that a majority of such techniques fail to perform properly on data sets due to the diversity in attack patterns [20]. Even most of the countermeasures developed so far are expected to handle only one type of communication protocol used by IoT devices at any point of time [21-22]. However, several other communication protocols are used by IoT operational networks and the guarantee of security in such networks is yet to be addressed by research activities carried out so far.

Table 1: A review of existing detection techniques for botnet attacks

Ref.	Datasets	Approach	Type of Attack	Detection Stage
Nguyen [9]	IoT-PoT	Graph-based	IoT botnet	Post Attack
Wang [10]	Self-collected	Hybrid	Botnet	Post Attack
Yassin [21]	Self-collected	Graph-based	Mirai Attack	Post Attack
Almutairi [22]	Self-collected	Flow-based	HTTPS, P2P, IRC, DNS	Post Attack
Blaise [23]	CTY-13	Flow-based	Botnet	Post Attack
Soe [24]	N-BaIoT	Flow-based	Mirai, BASHLITE	Both pre and post-attack
Sriram [8]	N-BaIoT	Flow-based	Mirai, BASHLITE	Both pre and post-attack
Nugraha [25]	CTU-13	Flow-based	Botnet	Post Attack
Parra [15]	N-BaIoT	Flow-based	Mirai, BASHLITE	Both pre and post-attack
Pektacs [16]	CTU-13	Hybrid	Botnet	Post Attack
Ahmed [17]	CTU-13	Flow-based	Botnet	Post Attack
Maeda [18]	CTU-13	Flow-based	Botnet	Post Attack
Meidan [19]	N-BaIoT	Flow-based	IoT botnets, Mirai, BASHLITE	Post Attack
Bovenzi [26]	Bot-IoT	Flow-based	IoT botnet	Both pre and post-attack
Mirsky [27]	Self-collected	Flow-based	IoT botnet	Both pre and post-attack

4. Proposed Methodology

The current attack detection techniques are capable of identifying a botnet attack only after a device gets infected with malicious scripts designed by an attacker. In addition to this, it has also been observed in previous sections that the performance of different techniques appears to be constrained by datasets used for training machine learning models. This kind of inadequacy of attack mitigation methods is primarily caused due to two reasons. The first reason is due to the diversity of botnet attacks. The second can be attributed to the lack of adequate methods to effectively utilize attack databases to extract features that can provide effective insights toward the mitigation of such attacks.

However, it is essential to provide a protective mechanism to hide IoT devices from such types of attacks during the initial scanning stages to prevent IoT devices from getting compromised.

In this work, a method named as **D**eep-**L**earning Based **M**ulti-Protocol **D**efence **(DLMD)** has been proposed to prevent botnet attacks in their early stages (such as scanning attacks). Further, it can identify DDoS attacks even before the attack initiation. An attacker might utilize bot-infected IoT devices to carry out a variety of malicious tasks. Such activities may range from flooding of spam mail to DDoS attacks, etc. In this work, however, the focus is to identify DDoS attacks launched by bot-infected IoT devices. This suggested method makes use of a dataset that has first been trained to detect

scanning activity and then has been trained to recognize DDoS attacks initiated either inside or from outside of the network by an attacker or compromised IoT devices.

The next phases describe the methods of data set generation and fitting the same into operation for the experimental purpose as per the stages mentioned below.

4.1 Data collection

In the related research section, the need for a suitable data set for research into the mitigation of DDoS attacks caused by botnets in IoT networks has been highlighted. Hence, in the primitive state of research, the emphasis was given to the design of a custom dataset in this research work.

In the first stage, a few computers and mobile phones were connected to a network and packets were captured in the network using *Wireshark*. Two things were considered while capturing packets in the network. Primarily, malicious traffic for attack patterns like TCP Syn attack, ICMP attack, and UDP flood attack was considered. In addition to the attacks, TCP, UDP, and ICMP traffic were captured for the experimental purpose. The Figure 3 below explains the steps involved in the experimental setup.

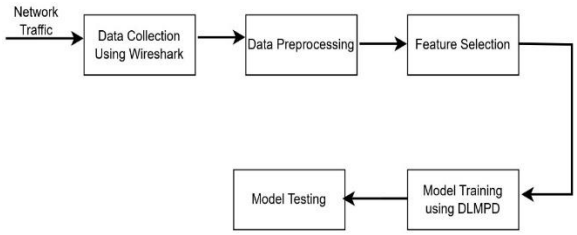

Figure 3: Stages in Experimental Setup

4.2 Data preprocessing

The data pre-processing includes the counting of null values in each feature, categorical features, numerical features, and continuous features. The output from the packet capturing process of *Wireshark* is received in a *.pcap* format which is then subjected to preprocessing. In this stage, data containing null or incomplete data were rejected. The attack traffic values were labeled as 1 for legitimate traffic and 0 for attack traffic. The simulation was carried out for four hours and 834760 instances of data were collected and recorded.

4.3 Feature selection

After preprocessing the data and eliminating structural errors, unwanted data, noisy values, and null values from the raw dataset, we utilized DNN (Deep Neural Network) to extract a set of 23 features from the dataset. These features are outlined in the Table 2 displayed below. The neural networks are capable of receiving user inputs, performing required complex calculations on the input data, and providing output in elaborate formats that can be used as a reference for various purposes. Then the dataset is made to fit the model for finding the accuracy of the model over 100 epochs which is used to separate the training dataset into distinct phases and is useful for periodic evaluation. It can be easily comprehended that in the case of an instance where a DDoS attack is happening, the number of transmitted packets will be less, and received packets will be considerably higher. Hence, these two factors will be crucial in labeling the attack type.

4.4 Model training for detection of DDoS attack

Upon completing the feature selection process, the dataset was divided into two separate parts for training and testing purposes. Following the holdout method, 70 percent of the dataset, obtained after feature selection, was allocated for training, and the remaining 30 percent was designated for testing.

The entire dataset was separated into four different parts for the current experiment. The first three parts consisted of TCP, UDP, and ICMP individually and the fourth part consisted of mixed traffic containing all the three protocols. Each of the portions has 1,50,000 instances. This hybrid approach has been followed to investigate the efficiency of the proposed methodology against a variety of attacks targeting devices that may run using a single type of protocol or a mix of different protocols.

Table 2: Features of the Dataset

Sl No	Features	Meaning
1	Pkt_count	Number of packets
2	byte_count	Count of bytes
3	duration_sec	Packet transmission
4	duration_nsec	Packet transmission (in nanoseconds)
5	Src_IP	Source IP
6	Dest_IP	Destination IP
7	Port_Number	Port number
8	tx_bytes	Number of bytes transferred
9	rx_bytes	Number of bytes received
10	dt_field	Shows the date and time
11	Byte_Per_Flow	Byte count during a flow
12	Pkt_Per_Flow	Packet count during a flow
13	Pkt_rate	Number of packets transmitted per second
14	number_of_Pkt_ins_messages	Messages that are generated by the switch
15	Flow_entries_of_switch	Entries in the flow table
16	tx_kbps	Packet transmission (in kbps)
17	rx_kbps	Packet reception (in kbps)
18	Port_Bandwidth	tx_kbps +rx_kbps
19	Dt	Destination
20	Label	‘0’ for genuine traffic and ‘1’ for attack traffic
21	Tot_kbps	Packet transmission rate (kbps)
22	Flows	Count of packets from a source to a destination
23	Protocol	Protocol used

4.5 Experimental setup

For experimental purposes, the first step is to capture network traffic and create a dataset for the experiment which has already been discussed in section 4.4. The packet capturing was done in a virtual environment using two instances of both Windows and Linux. The deep learning algorithm (DLMD)is carried out on Google Collab. The *.pcap* file generated from the packet capturing process is then converted to *.csv* and the *.csv* file is fed as input to the machine learning algorithm. In the next step, accuracies against nine different classifiers i.e., Deep neural network, XGBoost, RBF-SVM, Decision Tree, SVM, SGD, LR, Naive Bayes, and Quadratic are calculated. The Figure 4 below shows the step-by-step method of the DLMD algorithm.

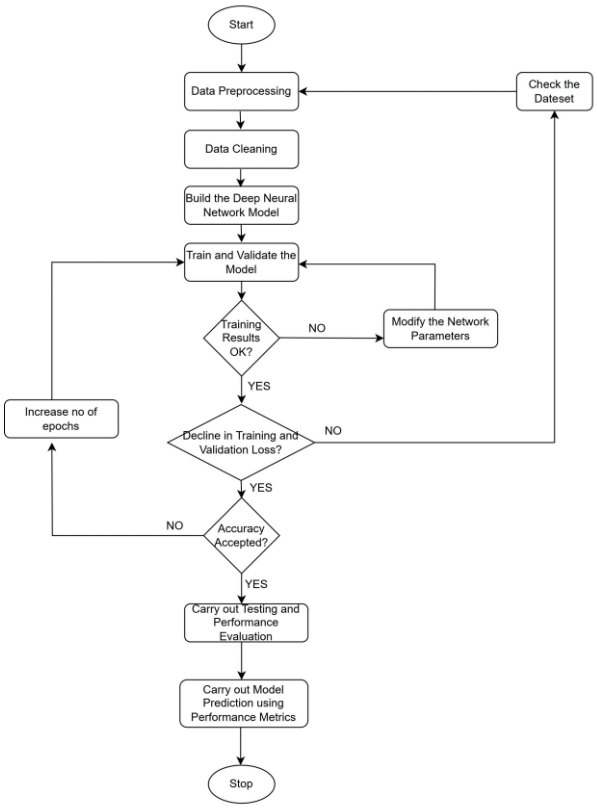

Figure 4: Flow Diagram of DLMD Method

5. Results and Discussion

The comparison of the accuracy of the model in this work against TCP, UDP, ICMP, and mixed traffic using different classifiers has been mentioned in Table 3.

Table 3: Accuracy Values with Different Classifiers

Sl No	Classifier	UDP	TCP	ICMP	Mixed
1	DNN	95.851	96.85	96.46	98.645
2	XGBoost	94.894	94.43	95.26	98.356
3	RBF_SVM	91.112	90.95	91.21	97.431
4	Decision Tree	93.64	93.62	94.78	96.594
5	KNN	92.408	91.96	91.96	96.481
6	SGD	75.114	76.52	78.24	83.946
7	Logistic Regression	71.95	70.54	72.65	83.901
8	Naive Bayes	64.19	65.52	69.06	71.49
9	Quadratic	43.145	43.25	53.54	50.003

DNN-Deep Neural network, XGBoost-Extreme Gradient Boosting, RBF_SVM-Radial Basis Function Support Vector Machine, KNN-K- Nearest Neighbor, SGD-Stochastic Gradient Descent

It can be inferred from the observations in the Table 3 that the proposed model provides consistent accuracy rates when the network is flooded with a single type of traffic. For example, the model provides accuracy values of ~96% for DNN in the presence of TCP, ICMP, and UDP traffic in the network. The same classifier yields an accuracy value of ~ 98% in the presence of mixed data traffic. A similar pattern can also be observed for other classifiers as in the Table 3 above. The observations in the preceding table suggest that the proposed model can be useful in application scenarios comprising devices being operated using multiple types of network traffic.

The validation of the proposed model is critical to justify the accuracy score of the proposed model. Hence, the learning method in DLMD algorithm was subjected to an evaluation of training loss and accuracy scores over a fixed number of epochs. The graphical representation of this analysis is presented in the Figure 5 below.

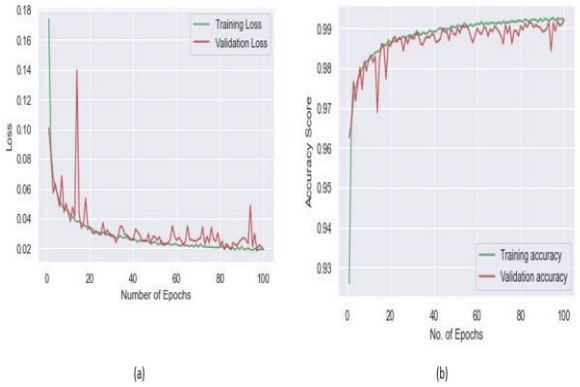

Figure 5: (a)Training loss Vs number of epochs (b)Accuracy Vs number of epochs

It can be affirmed from the Figure 5 that both loss and accuracy when compared with the numbers of epochs seem to almost converge. The learning curve in the Figure 5(a) demonstrates a strong alignment, with the training loss plot gradually decreasing to 0.02% and the validation loss plot following a similar trend, reaching 0.02% as well. Figure 5(b) displays a plot demonstrating the progressive growth of training accuracy, reaching an impressive 99.14% as the number of epochs increases. Additionally, the validation accuracy plot exhibits a similar trend, reaching a stable point of 99.14% with only a slight difference from the training accuracy. These observations indicate that the model fits well and performs effectively. To further guarantee the suitability of the proposed DLMD model, metrics like accuracy, precision, recall, and F1 scores were calculated. The accuracy value gives a rough idea about the total percentage of correct classifications yielded by the proposed model. A substantially good value of precision ensures a minimal amount of misclassification or false positives to be exact. Similarly, recall values are indicators of the number of false negatives. In addition to accuracy, precision, and recall F1 score is useful to judge a model's effectiveness as it computes the correct classifications across the entire dataset. The DLMD model also provides values of accuracy, precision, recall, and F1 scores of 98.64%, 97.63%, 98.35%, and 98.26% respectively. The values of such performance indicators are shown in the Figure 6.

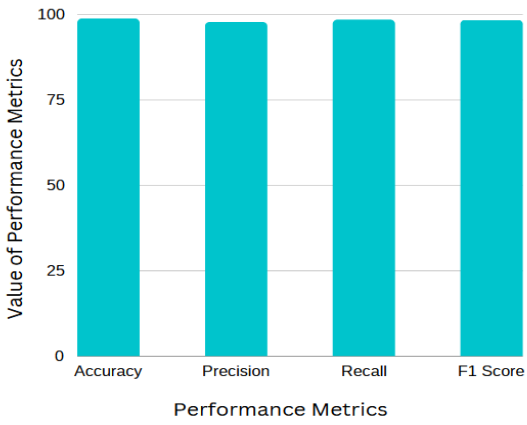

Figure 6: Performance Indicators of Machine Learning Model

Table 4: Comparison of DMLD with previous research work

Ref	Datasets	Approach	Detection Stage	Results Obtained in Earlier Work (As per Reference)	Results obtained using DLMD Method
[10]	Self-collected	H	Post	Only C&C* unit level Accuracy-91.66%	Both C&C Unit and End User node protection Accuracy-98.6%
[21]	Self-collected	G	Post	Only registry detection method available	Network ports, IP addresses and underlying protocols are profiles for detection
[22]	Self-collected	F	Post	Only TCP and UDP Accuracy-97.6%	TCP, UDP and ICMP Accuracy-98.6%
[24]	N-BaIoT	F	Hybrid	Protection only at the C&C unit level	End-user node protection
[25]	CTU-13	F	Post	Only server-side detection available	SDN environment for detection
[15]	N-BaIoT	F	Hybrid	URL based approach for attack detection	Protocol based approach for attack detection
[16]	CTU-13	H	Post	Accuracy-97.6%	Accuracy-98.6%
[17]	CTU-13	H	Post	A mixed traffic datasets with 2000 instances	Mixed traffic dataset of 1,50,000 instances
[18]	CTU-13	H	Post	SDN and REST API used only for detection	End-user SDN nodes have been used for both detection and blocking
[19]	N-BaIoT	H	Post	Only signature-based botnet detection.	Protocol based attack detection and blocking

***Approaches**: **F**- Flow Based, **G**-Graph Based, **H**-Hybrid comprising of both flow and graph-based methods*

***Detection Stage**: **Pre**-Pre attack, **Post**-Post attack, **Hybrid**-Combining both Pre and Post attack detection methods*

** Command & Control Unit*

During the course of this research, there has been a thorough study on previously carried out research work in the section 3 and the analytical finding has been presented in the Table 1. The Table 4 above compares distinctive factors of the proposed model with some of the work discussed earlier. For instance, only a very few of them have carried out their research on custom-made datasets whereas others have relied on datasets from various sources. This can be emphasized that the pattern of attacks may vary over time and in those circumstances may not be practically very useful if datasets are not updated over time. Similarly, the work in [22] uses mixed network traffic, but the accuracy is 97.6% whereas the DMLD yields an accuracy of 98.6%. The work in [10] and [24] is only intended to provide protection at the C&C unit level whereas the proposed DMLD method provides protection even at both the C&C unit and end-user level. In another instance, the work in [21] provides registry detection method which may enforce additional programming overheads. Hence, the algorithm in this work uses IP, network protocols and different service ports for profiling and monitoring network attacks. The dataset for training the model in DMLD contains around 1,50,000 instances comprising of mixed mode traffic whereas, the work in [17] takes only 2,000 data instances in to account. Hence, the chances of misclassification in the case of DMLD would definitely be at a very low level while providing a higher value of accuracy. The comparison in the above table also reveals that no other research has yielded an accuracy greater than the achieved accuracy of 98.6%. In the work of [18], SDN and

REST API has been used for attack detection. The DLMD uses SDN for both detection and blocking of malicious nodes. A majority of work carried out can only detect attack in post-attack stage and only a handful of work are intended to detect attack at both pre-attack and post-attack stages. Hence, in a nutshell it can be emphasized that the proposed DLMD algorithm can provide protection against attacks in an IoT network of varieties of application protocols at a considerably higher accuracy of 98.6% which is faster without the need of additional end devices. The protection can be used for both end-devices and C&C units with facilities for both detection and blocking of malicious nodes without major concerns for misclassification of attacks.

6. Conclusion

The objective of this research paper is to diminish the effect of DDoS attacks through the utilization of a machine learning approach, which can leverage diverse network communication methods. This work aims to detect and block network hosts desiring to create DDoS attack scenarios by classifying them based on network traffic. Furthermore, there are several pertinent concerns regarding the previously utilized datasets, which have already been deliberated. To tackle these concerns, a fundamental aspect of this endeavor involves capturing network traffic and crafting a fresh dataset. Following a process of data preprocessing and feature selection, the model underwent training and testing using the DLMD method described in Section 4. It has been observed from previous research findings that a majority of the DDoS attack mitigation techniques are intended for devices using a single type of communication protocol. Hence, the machine learning model in this methodology was trained considering data instances of TCP, ICMP, and UDP separately three different times and mixed protocol data instances. It is noteworthy that typical IoT applications utilize multiple types of communication protocols. Along the same line, the model used here was tested according to appropriate testing parameters. The proposed methodology even achieved an accuracy score of 98.6 which is unique in this kind of research.

This work is currently focusing only on a few selected communication protocols and in the future, this model can be extended to several other communication protocols that may suffer from attacks. This research work has been carried out on a dataset that has been taken from a simulated environment and this can be further improved upon its usage in a network with live traffic and live genuine and malicious nodes.

References

[1] Zhang, C., & Green, R. (2015, April). Communication security in Internet of thing: preventive measure and avoid DDoS attack over IoT network. In Proceedings of the 18th symposium on communications & networking (pp. 8-15).

[2] Salim, M. M., Rathore, S., & Park, J. H. (2020). Distributed denial of service attacks and its defenses in IoT: a survey. The Journal of Supercomputing, 76, 5320-5363.

[3] Adat, V., & Gupta, B. B. (2017, April). A DDoS attack mitigation framework for the internet of things. In 2017 international conference on communication and signal processing (ICCSP) (pp. 2036-2041). IEEE.

[4] Islam, N., Farhin, F., Sultana, I., Kaiser, M. S., Rahman, M. S., Mahmud, M., ... & Cho, G. H. (2021). Towards Machine Learning Based Intrusion Detection in IoT Networks. Computers, Materials & Continua, 69(2).

[5] Hussain, F., Abbas, S. G., Husnain, M., Fayyaz, U. U., Shahzad, F., & Shah, G. A. (2020, November). IoT DoS and DDoS attack detection using ResNet. In 2020 IEEE 23rd International Multitopic Conference (INMIC) (pp. 1-6). IEEE.

[6] Nsaif, M. R., Abbood, M. F., & Mahdi, A. F. (2020). Detection and prevention algorithm of DDoS attack over the IoT networks. TEM Journal, 9(3), 899.

[7] Lima Filho, F. S. D., Silveira, F. A., de Medeiros Brito Junior, A., Vargas-Solar, G., & Silveira, L. F. (2019). Smart detection: an online approach for DoS/DDoS attack detection

using machine learning. Security and Communication Networks, 2019, 1-15.

[8] Sriram, S., Vinayakumar, R., Alazab, M., & Soman, K. P. (2020, July). Network flow-based IoT botnet attack detection using deep learning. In IEEE INFOCOM 2020-IEEE conference on computer communications workshops (INFOCOM WKSHPS) (pp. 189-194). IEEE.

[9] Nguyen, H. T., Ngo, Q. D., & Le, V. H. (2020). A novel graph-based approach for IoT botnet detection. International Journal of Information Security, 19(5), 567-577.

[10] Wang, W., Shang, Y., He, Y., Li, Y., & Liu, J. (2020). BotMark: Automated botnet detection with hybrid analysis of flow-based and graph-based traffic behaviors. Information Sciences, 511, 284-296.

[11] Shurman, M. M., Khrais, R. M., & Yateem, A. A. (2019, December). IoT denial-of-service attack detection and prevention using hybrid IDS. In 2019 International Arab Conference on Information Technology (ACIT) (pp. 252-254). IEEE.

[12] Censys. (2023, May 30). Exposure Management and Threat Hunting Solutions | Censys. https://censys.io/

[13] Shodan. (n.d.). Shodan. https://www.shodan.io/

[14] Vlajic, N., & Zhou, D. (2018). IoT is a land of opportunity for DDoS hackers. Computer, 51(7), 26-34.

[15] Parra, G. D. L. T., Rad, P., Choo, K. K. R., & Beebe, N. (2020). Detecting Internet of Things attacks using distributed deep learning. Journal of Network and Computer Applications, 163, 102662.

[16] Pektaş, A., & Acarman, T. (2018). Botnet detection based on network flow summary and deep learning. International Journal of Network Management, 28(6), e2039.

[17] Ahmed, A. A., Jabbar, W. A., Sadiq, A. S., & Patel, H. (2020). Deep learning-based classification model for botnet attack detection. Journal of Ambient Intelligence and Humanized Computing, 1-10.

[18] Maeda, S., Kanai, A., Tanimoto, S., Hatashima, T., & Ohkubo, K. (2019, January). A botnet detection method on SDN using deep learning. In 2019 IEEE International Conference on Consumer Electronics (ICCE) (pp. 1-6). IEEE.

[19] Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Shabtai, A., Breitenbacher, D., & Elovici, Y. (2018). N-bait—network-based detection of iot botnet attacks using deep autoencoders. IEEE Pervasive Computing, 17(3), 12-22.

[20] Hussain, F., Abbas, S. G., Fayyaz, U. U., Shah, G. A., Toqeer, A., & Ali, A. (2020, November). Towards a universal features set for IoT botnet attacks detection. In 2020 IEEE 23rd International Multitopic Conference (INMIC) (pp. 1-6). IEEE.

[21] Yassin, W., Abdullah, R., Abdollah, M. F., Mas' ud, Z., & Bakhari, F. A. (2019, December). An IoT botnet prediction model using frequency based dependency graph: Proof-of-concept. In Proceedings of the 2019 7th International Conference on Information Technology: IoT and Smart City (pp. 344-352).

[22] Almutairi, S., Mahfoudh, S., Almutairi, S., & Alowibdi, J. S. (2020). Hybrid botnet detection based on host and network analysis. Journal of Computer Networks and Communications, 2020, 1-16.

[23] Blaise, A., Bouet, M., Conan, V., & Secci, S. (2020). Botnet fingerprinting: A frequency distributions scheme for lightweight bot detection. IEEE Transactions on Network and Service Management, 17(3), 1701-1714.

[24] Soe, Y. N., Feng, Y., Santosa, P. I., Hartanto, R., & Sakurai, K. (2020). Machine learning-based IoT-botnet attack detection with sequential architecture. Sensors, 20(16), 4372.

[25] Nugraha, B., Nambiar, A., & Bauschert, T. (2020, October). Performance evaluation of botnet detection using deep learning techniques. In 2020 11th International Conference on Network of the Future (NoF) (pp. 141-149). IEEE.

[26] Bovenzi, G., Aceto, G., Ciuonzo, D., Persico, V., & Pescapé, A. (2020, December). A hierarchical hybrid intrusion detection approach in IoT scenarios. In GLOBECOM

2020-2020 IEEE Global Communications Conference (pp. 1-7). IEEE.

[27] Mirsky, Y., Doitshman, T., Elovici, Y., & Shabtai, A. (2018). Kitsune: an ensemble of autoencoders for online network intrusion detection. arXiv preprint arXiv:1802.09089.

[28] Keserwani, P. K., Govil, M. C., Pilli, E. S., & Govil, P. (2021). A smart anomaly-based intrusion detection system for the Internet of Things (IoT) network using GWO–PSO–RF model. Journal of Reliable Intelligent Environments, 7, 3-21.

[29] Illy, P., Kaddoum, G., Kaur, K., & Garg, S. (2022). ML-based IDPS enhancement with complementary features for home IoT networks. IEEE Transactions on Network and Service Management, 19(2), 772-783.