DIDDOS: An approach for detection and identification of Distributed Denial of Service (DDoS) cyberattacks using Gated Recurrent Units (GRU)

Saif ur Rehman[a], Mubashir Khaliq[a], Syed Ibrahim Imtiaz[b], Amir Rasool[c], Muahmmad Shafiq[d,*], Abdul Rehman Javed[e], Zunera Jalil[e], Ali Kashif Bashir[f]

[a]*Department of Computer Science, PAF Complex, E-9, Air University, Islamabad, Pakistan*
[b]*National Center for Cyber Security, PAF Complex, E-9, Air University, Islamabad, Pakistan*
[c]*Institute of Avionics and Aeronautics, Air University, Islamabad, Pakistan*
[d]*Department of Cyberspace Institute of Advanced Technology, GuangZhou University, GuangZhou, 510006, China*
[e]*Department of Cyber Security, PAF Complex, E-9, Air University, Islamabad, Pakistan*
[f]*School of Computing, Mathematics, and Digital Technology, Manchester Metropolitan University, United Kingdom*

Abstract

Distributed Denial of Service (DDoS) attacks can put the communication networks in instability by throwing malicious traffic and requests in bulk over the network. Computer networks form a complex chain of nodes resulting in a formation of vigorous structure. Thus, in this scenario, it becomes a challenging task to provide an efficient and secure environment for the user. Numerous approaches have been adopted in the past to detect and prevent DDoS attacks but lack in providing efficient and reliable attack detection. As a result, there is still notable room for improvement in providing security against DDoS attacks. To overcome the problem of DDoS attacks detection, in this paper, a novel high-efficient approach is proposed named *DIDDOS* to protect against real-world new type DDoS attacks using Gated Recurrent Unit (GRU) a type of Recurrent Neural Network (RNN). For effective performance results different classification algorithms are applied Gated Recurrent Units (GRU), Recurrent Neural Networks (RNN), Naive Bayes (NB), and Sequential Minimal Optimization (SMO) are utilized to detect and identify DDoS attacks. For the performance evaluation metrics like accuracy, recall, f1-score, precision are used to evaluate the efficiency of the machine and deep learning classifiers. Experimental results yield the highest accuracy of 99.69% for DDoS classification in case of reflection attacks and 99.94% for DDoS classification in case of exploitation attacks using GRU.

Keywords: Cyberattack, Cybersecurity, DDoS, IDS, Deep Learning, GRU, Malware, Network, RNN, Traffic

---

*Corresponding author
Email addresses: `181065@students.au.edu.pk` (Saif ur Rehman), `160565@students.au.edu.pk` (Mubashir

1. Introduction

Internet security is one of the paramount challenges and primary concern of Information Technology (IT) specifically for the Internet of Things (IoT), Mobile devices, and Medical data [1, 2, 3, 4]. As the demand for IT services is increasing, similarly potential cyberattacks are increasing rapidly [5, 6, 7, 8, 9]. Among the many existing cyberattacks (i.e., DDOS, phishing, zero-days, rootkits, drive-by, password, SQL injection, ransomware), the Distributed Denial of Service (DDoS) attack can be utilized to breach the intranet and Internet resources of a particular organization or online business [10, 11, 12, 13]. Usually, in this attack, legitimate users are deprived of using web-based services provided by a large number of compromised machines that are highly vulnerable. DDoS attacks attempt to make a machine or network resource unavailable to its intended users. DDoS attacks are sent by two or more persons, or bots [14, 15], while DoS attacks are sent by one person or system. A bot is a compromised device created when a computer is penetrated by software from a malware code [16]. In this paper, the main focus is to keep an eye on DDoS attacks. These can be implemented in network, transport, and application layers using different protocols, such as TCP, UDP, ICMP, and HTTP. Furthermore, a DDoS attack [17, 18, 19] can be a large-scale coordinated attack on the provision of services of a victim system or network resources, launched indirectly through a large number of compromised computer agents on the internet [20, 13, 21]. Before applying an attack the attacker takes a large number of computer machines under his control over the internet and these computers are vulnerable machines. The attacker exploits these computer weaknesses by inserting malicious code or some other hacking technique so that they become operational under his command.

DDoS attacks are constantly evolving as the nature of the technology used and the motivations of the attackers are changing. Even today, perpetrators are being caught and charged with DDoS attacks launched via botnets that cause tens of thousands of dollars of damage to the victims. Last year's massive attack on Estonian Government web sites bought this attack method squarely into the public eye [22]. DDoS attacks on the Internet can be launched using two techniques. In the first technique, the attacker sends some malicious packets to the victim to confuse a protocol or an application running on it (i.e., vulnerability attack [23]). The Second technique essentially includes the network/transport-level/application-level flooding attacks [23], in which an attacker to do one or both of the following: (i) interrupt a legitimate user's connec-

Khaliq), syedibrahimimtiaz@gmail.com (Syed Ibrahim Imtiaz), aamir.rasool.au@gmail.com (Amir Rasool), srsshafiq@gmail.com (Muahmmad Shafiq), abdulrehman.cs@au.edu.pk (Abdul Rehman Javed), zunera.jalil@mail.au.edu.pk (Zunera Jalil), dr.alikashif.b@ieee.org (Ali Kashif Bashir)

tivity by exhausting bandwidth, network resources, or router processing capacity or (ii) disrupt services of a legitimate user's by exhausting the server resources such as CPU, memory, disk/database bandwidth and I/O bandwidth.

State-of-the-art studies [24, 25, 26] lack in providing accurate detection of real-world new types of Distributed Denial of Service (DDoS) cyberattacks and identify the type of DDOS attack. (i.e., NTP, UDP). The *DIDDOS* improves the detection and identification of real-world new types of Distributed Denial of Service (DDoS) using customized GRU as well as addresses the limitation of limited attack samples (imbalanced data) in the dataset by improving the representation of minority class.

The main contributions to this paper are:

- Propose an approach named *DIDDOS* to detect a real-world Distributed Denial of Service (DDoS) cyberattacks and identify the type of DDOS attack.
- Evaluate the effectiveness of the *DIDDOS* using conventional machine learning classifiers (i.e., Naïve Bayes (NB), Sequential Minimal Optimization (SMO)) and deep learning approaches by using Gated Recurrent Units (GRU) and Recurrent Neural Networks (RNN).
- Present a comparative analysis with state-of-the-art studies and conventional approaches (i.e., Recurrent Neural Networks (RNN), Naive Bayes (NB), Sequential Minimal Optimization (SMO)).
- Experimental results conclude that GRU provides efficient detection and identification rate than RNN, other conventional algorithms, and state-of-the-art studies.

The rest of the paper is organized as follows. Section 2 briefly covers the related work and recent advancements on DDOS attack detection and identification. Section 3 provides extensive discussion on the selected dataset. Section 4 presents the proposed approach *DIDDOS* for DDoS attack detection and identification. The experimental setup and results are articulated in Section 5. Section 6 presents the comparative analysis with state-of-the-art studies and conventional machine learning algorithms and overall discussion. Section 7 concludes the paper and leads towards future work.

2. Related Work

The number of DDoS attacks is increasing every year and from statistics [27] of Cisco Visual Networking Index (VNI) in 2017, it is confirmed that DDoS attacks are anticipated to double to 14.5 million by 2022.

This shows that DDoS attacks are increasing at a very unpleasant rate. However, this is a very challenging task to update the detection techniques up to the current DDoS attacks. The authors in [28] proposed the dataset ”DDoS Attack 2007” containing the traffic traces for one whole hour stored in the pcap format and details of attack traffic to the victim, as well as responses to the attack from the victim. In 2004, the authors Mirkovic and Reiher et. al. [23] introduced classifications of different DDoS attacks and conceivable guard components. The attacks were classified as automation, vulnerability, source address validity, attack rate dynamics, characterization, the persistence of agents, victim, and impact on the victim. In automation-based techniques, the machine is checked for vulnerability. In this research, the activity feed is checked to access the DDoS resistance mechanism. The authors in [29] performed a study that proposes a classification dependent on the degree of automation, architecture, impact, vulnerability, attack rate dynamics, scanning strategy, propagation strategy, and packet content. They also categorize the data into prevention and detection groups and claim that this classification is the best to detect where was the attack originated. The study also proposed a framework that can detect any DDoS attack using the K-means algorithm. However, no experiments are being conducted to validate the proposed classification.

In 2016, the study [30] concentrated on DDoS Taxonomy in the cloud computing paradigm. The authors propose the classification for the different potential DDoS attacks as a degree of automation, vulnerability, attack rate dynamics, and attack impact. Some resembling work was researched by [23] but it was unique because of DDoS attack classification features which include real-time response, throughput, request, response time, and zero-day attack detection ability. The research by Masdari and Jalali [31] concentrated on the analysis of DDoS attacks in cloud computing. In their study, they showed that different DDoS attacks by showing how these attacks violated the vulnerabilities. Lastly, the study also characterized the DDoS attacks dependent on some modules like virtual machines, cloud scheduler, hyper-visor, web service, cloud clients, IaaS, and SaaS-based attacks [32]. The most effective cloud computing attacks have been recognized as bandwidth attacks, connectivity attacks, resource exhaustion, limitation exploitation, process disruption, data corruption, and physical disruption. The primary features of the researches are discussed in Table 1.

Modi et al.[34] proposed a NIDS that integrates the Naive Bayes classifier and Snort. In their study, they showed that Snort signature-based detection system filters the captured packets. The captured packets will be divided into two sets: intrusion packets and non-intrusion packets. The intrusion packets will be logged and denied by the system. Meanwhile, the non-intrusion packets will be pre-processed and fed

Table 1: Primary Features of the Related Works [24] in Terms of Providing Multilevel Protection

Authors	**OSI-Layer**	**Network-Based Environment**	**Known Attacks/ Potential threats**	**Defense Mechanism**
[23]	×	×	✓	✓
[29]	×	×	✓	✓
[30]	×	Cloud Computing	✓	✓
[31]	Application Network Transport	Cloud Computing	✓	✓
[33]	Application	×	✓	×

into the anomaly detection module. The anomaly detection module employs the Naive Bayes classifier to further classify the non-intrusion packets into normal and intrusion packets. Once the packets are classified as intrusions, they will be logged and denied. Only when the packets are labeled as normal can they be allowed to go to the system. Similarly authors in [35] proposed a deep learning model for anomaly detection in connected vehicles. Qin et al. [36] designed a similar framework as [37] did. Jing et al. [38] have proposed a Support Vector Machine(SVM) with a new scaling technique in 2019. The necessary steps are: (1) divide the dataset into the training set and testing set; (2) Pre-processing the data (both training set and testing set) with scaling technique; (3) Train the SVM model with the training set; (4) Test the model with the testing set; (5) Record the classification result. Authors in [39, 13, 40] used various feature engineering and machine learning for the intrusion detection.

In the area of intrusion detection, several researchers endeavour hard to develop effective model for the intrusion detection in RNN [41, 42, 43, 44, 45, 46]. Yin et al.[41] use RNN with forwarding propagation and weights updates (backpropagation). Qureshi et al.[42] rebalanced the KDD'99 dataset before training and testing. The proportion of abnormal data in the training set is rebalanced to 46.5%. The authors have referred to the work of Bajaj et al. [47] about feature reduction and dropped some features in the preprocessing to improve the detection rate. Althubiti et al. [43] use Long-Short-Term-Memory RNN and ADAM optimizer. Meng et al. [44] took a further step and integrate kernel PCA and LSTM. Kernel PCA is a type of dimension reduction technique and this technique is different from PCA because it generalizes PCA from linear to nonlinear dimension reduction. The overall Detection Rate tested on KDD'99 is 99.46%, while the False Alarm Rate is 4.86%. Le et al. [45] compared several gradient descent optimizers with LSTM. Gradient Descent is a classic optimizer used in deep learning. However, there are many variations of Gradient Descent optimizers. The scope of all the above attacks is limited because there are new attacks that can be carried out using TCP/UDP based protocols at the application layer.

This work aims to overcome the limitations in such a way that a dataset that has been released in 2019 is utilized. The dataset includes new attacks that can be carried out using TCP/UDP based protocols at the application layer. Machine and deep learning-based approaches are being conducted to evaluate the detection and identification of DDOS attacks.

3. Dataset Selection

For the DDoS attacks, different datasets are used by numerous researchers that contain information about a variety of attacks. But new attacks are made which poses a security challenge. So, that is why datasets are updated to increase security. We needed a newly released dataset that contains the latest information about Distributive Denial of Service attacks or DDoS attacks. So, for this research, a recently published dataset CICDDoS2019[1] is selected, which contains benign and the most up-to-date realistic background DDoS traffic, which resembles the true real-world data. It also includes the results of the network traffic analysis using CICFlowMeter-V3 with labeled flows based on the time stamp, source, and destination IPs, source and destination ports, protocols, and attacks. For this dataset, the abstract behavior of 25 users based on the HTTP, HTTPS, FTP, SSH, and email protocols was established.

In Section 2, as explained that there exist no other datasets that have captured modern reflective DDoS attacks. The new reflective DDoS attacks are NTP, NetBIOS, SSDP, UDP-Lag, and TFTP. The important part of analyzing the network packets is to keep the payloads while anonymizing the traffic. The above datasets anonymized the traffic but removed the payloads which shows the datasets discussed in 2 were not complete and the selected dataset CICDDoS2019 for this research is better concerning the factors complete traffic, attack diversity, data source heterogeneity, complete interaction, and complete capture. A graphical representation of different DDoS attacks and their types can be seen in Table 1 which was introduced by [24] by Iman Sharafaldin in 2019. In this dataset, there are different modern reflective DDoS attacks such as PortMap, NetBIOS, LDAP, MSSQL, UDP, UDP-Lag, SYN, NTP, DNS, and SNMP. Moreover, 12 DDoS attacks include NTP, DNS, LDAP, MSSQL, NetBIOS, SNMP, SSDP, UDP, UDP-Lag, WebDDoS, SYN, and TFTP are used on the training day, and 7 attacks including PortScan, NetBIOS, LDAP, MSSQL, UDP,

[1] [48] CICDDoS2019 Dataset Link: https://www.unb.ca/cic/datasets/ddos-2019.html

UDP-Lag and SYN in the testing day. The traffic volume for WebDDoS was so low and PortScan just has been executed in the testing day and will be unknown for evaluating the proposed model.

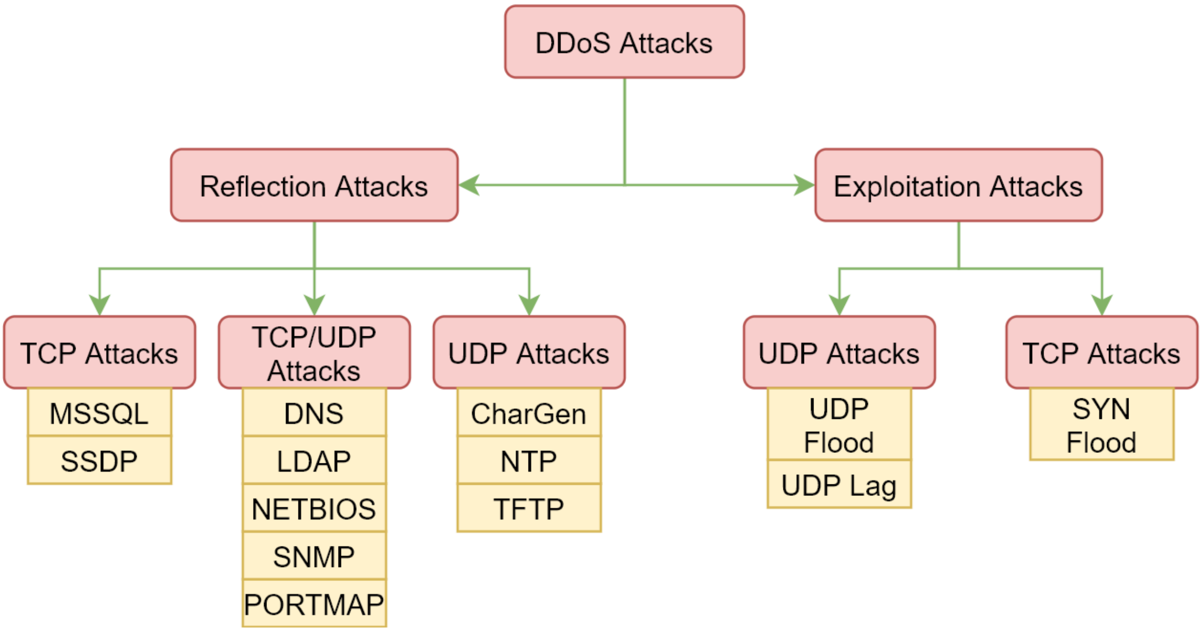


Figure 1: Graphical Representation of DDoS Attacks Hierarchy and Categorization

3.1. Reflection-based DDoS Attacks

Are those kinds of attacks in which the identity of the attacker remains hidden by utilizing legitimate third-party component. The packets are sent to reflector servers by attackers with source IP address set to target the victim & rsquos IP address to overwhelm the victim with response packets. These attacks can be carried out through application layer protocols using transport layer protocols, i.e., Transmission Control Protocol (TCP), User Datagram Protocol (UDP), or through a combination of both. In this category, TCP based attacks include MSSQL, SSDP while UDP based attacks include CharGen, NTP, and TFTP. Certain attacks can be carried out using either TCP or UDP like DNS, LDAP, NETBIOS, and SNMP.

1. **MSSQL Attack:** Microsoft Structured Query Language (MSSQL) injection is an attack that makes it possible to execute malicious SQL statements [49].
2. **SSDP Attack:** An SSDP attack exploits Universal Plug and Play (UPnP) networking protocols to send a large amount of traffic to a victim to overwhelm their computing resources [50].
3. **DNS Attack:** A DNS attack exploits vulnerabilities in the DNS [51].

4. **LDAP Attack:** LDAP injection is an attack used to exploit web-based applications that construct LDAP statements based on user inputs [52].
5. **NETBIOS Attack:** A security exploit in Network Basic Input/Output System (NetBIOS) allows an attacker to see information in computer memory over a network [53].
6. **SNMP Attack:** A Simple Network Management Protocol (SNMP) attack generates a large amount of traffic which is directed at victims from multiple networks [25].
7. **PORTMAP Attack:** PORTMAP is an attack on TCP or UDP port 111 which is a service used to direct clients to the proper port number so they can communicate with the requested Remote Procedure Call (RPC) service [25].
8. **CharGen Attack:** Character Generator Protocol (CharGEN) flooding is an attack that is carried out by sending small packets carrying a spoofed IP of the victim to internet-enabled devices running CharGEN to exhaust computing resources [25].
9. **NTP Attack:** NTP is an amplification attack in which the attacker exploits publically accessible NTP servers to overwhelm the target with UDP traffic [54].
10. **TFTP Attack:** A TFTP attack exploits the buffer overflow vulnerability in a Trivial File Transfer Protocol (TFTP) server [55].

3.2. Exploitation-based DDoS attacks

Are those kinds of attacks in which the identity of the attacker remains hidden by utilizing legitimate third-party component. The packets are sent to reflector servers by attackers with the source IP address set to the target victim & rsquos IP address to overwhelm the victim with response packets. These attacks can also be carried out through application layer protocols using transport layer protocols e.g. TCP and UDP. TCP based exploitation attacks include SYN flood and UDP based attacks include UDP flood and UDP-Lag. UDP flood attack is initiated on the remote host by sending a large number of UDP packets. These UDP packets are sent to random ports on the target machine at a very high rate. As a result, the available bandwidth of the network gets exhausted, system crashes and performance degrades. On the other hand, the SYN flood also consumes server resources by exploiting the TCP-three-way handshake. This attack is initiated by sending repeated SYN packets to the target machine until the server crashes/malfunctions. The UDP-Lag attack is that kind of attack that disrupts the connection between the client and the server. This attack is mostly used in online gaming where the players want to slow down/interrupt the movement of other players to outmaneuver them. This attack can be carried in two ways, i.e., using a hardware switch known as a lag switch or by a software program that runs on the network and hogs the bandwidth of other users.

1. **UDP-Flood Attack:** User Datagram Protocol (UDP) flooding is an attack in which a large number of UDP packets are sent to a victim to overwhelm their ability to process and respond. The firewall protecting the target server is exhausted as a result [56].
2. **UDP-Lag Attack:** UDP-Lag is an attack that disrupts the connection between the client and server [57].
3. **SYN Flood Attack:** SYN flood is a denial-of-service attack in which an attacker sends a succession of SYN requests to a target system in an attempt to consume server resources so the system is unresponsive to legitimate traffic [25].

TCP-based attacks can employ Microsoft Structured Query Language (MSSQL) or Simple Service Discovery Protocol (SSDP) whereas UDP-based attacks utilize CharGen, Network Time Protocol (NTP), or Trivial File Transfer Protocol (TFTP). Certain attacks use a combination of these protocols and include Domain Name System (DNS), Lightweight Directory Access Protocol (LDAP), Network Basic Input/Output System (NetBIOS), Simple Network Management Protocol (SNMP), or PORT MAP. SYN flood is a denial-of-service attack in which an attacker sends a succession of SYN requests to a target system in an attempt to consume server resources so the system is unresponsive to legitimate traffic [25]. WebDDoS is an attack to take down the target website or slow it by flooding the network, server, or application with bogus traffic [58]. A TFTP attack exploits the buffer overflow vulnerability in a Trivial File Transfer Protocol (TFTP) server [25]. A DNS attack exploits vulnerabilities in the DNS [25]. PORT MAP is an attack on TCP or UDP port 111 which is a service used to direct clients to the proper port number so they can communicate with the requested Remote Procedure Call (RPC) service [25]. Microsoft Structured Query Language (MSSQL) injection is an attack that makes it possible to execute malicious SQL statements [25]. LDAP injection is an attack used to exploit web-based applications that construct LDAP statements based on user inputs [25]. NETBIOS is a security exploit in Network Basic Input/Output System (NetBIOS) that allows an attacker to see information in computer memory over a network [25]. NTP is an amplification attack in which the attacker exploits publically accessible NTP servers to overwhelm the target with UDP traffic [25]. An SSDP attack exploits Universal Plug and Play (UPnP) networking protocols to send a large amount of traffic to a victim to overwhelm their computing resources [25]. SNMP is a Simple Network Management Protocol (SNMP) attack that generates a large amount of traffic which is directed at victims from multiple networks

[25]. User Datagram Protocol (UDP) flooding is an attack in which a large number of UDP packets are sent to a victim to overwhelm their ability to process and respond. The firewall protecting the target server is exhausted as a result [25]. UDP-Lag UDP-Lag is an attack that disrupts the connection between the client and server [57]. CharGEN is Character Generator Protocol (CharGEN) flooding is an attack that is carried out by sending small packets carrying a spoofed IP of the victim to internet-enabled devices running CharGEN to exhaust computing resources [25].

4. Proposed Methodology

In this section, we present our proposed *DIDDOS* approach for the detection and identification of DDoS attacks. The proposed approach comprises data normalization, feature extraction, and classification of attacks. Figure 2 summarizes our proposed approach which consists of deep learning classifiers for the classification of multiple types of DDoS attacks. In Figure 2, the detailed methodology can be clearly seen in which firstly the feature extraction and feature normalization takes place. Then the dataset is checked and if it has oversampling problems then the dataset is balanced by using SMOTE with the help of the tool WEKA. After this step, the algorithms are deployed on the datasets to evaluate their performance to detect malware.

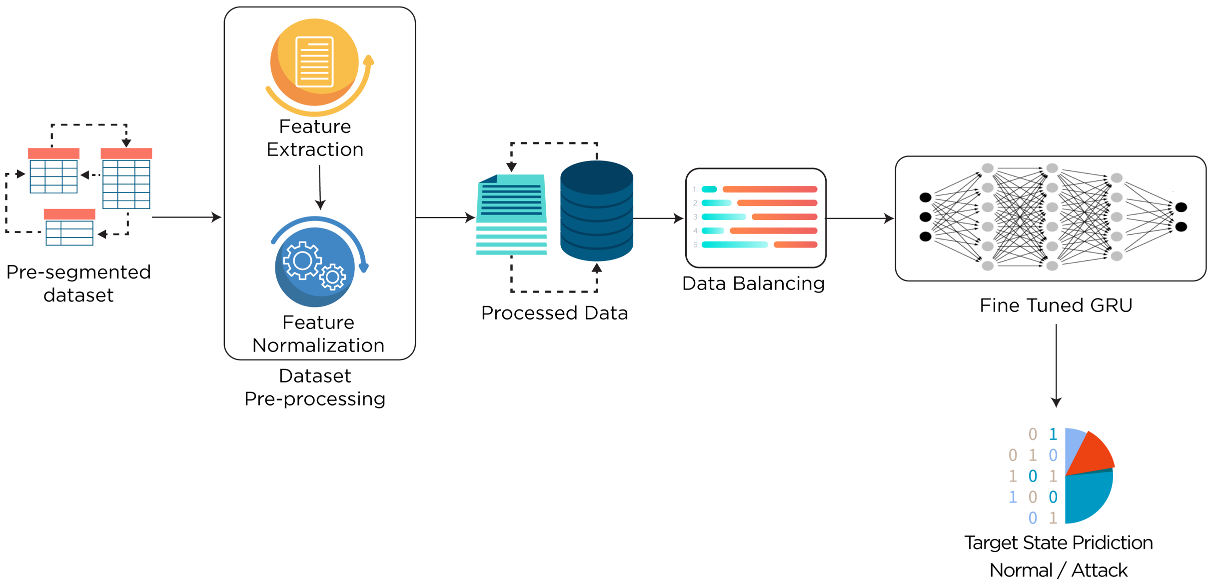


Figure 2: Graphical Representation of the *DIDDOS* Demonstrating the Worflow of the Model

4.1. Pre-Processing

In the Pre-processing stage, the dataset is optimized so that the results could be achieved with the highest accuracy. This includes dealing with NaN and duplicate instances. Typically these instances are removed and then the dataset is normalized and scaled according to the algorithm. In this case, MinMax scaling from [59] is used for feature normalization which used the Equation 1 to normalize the data.

$$X_{norm} = \frac{X_i - X_{min}}{X_{max} - X_{min}} \tag{1}$$

In equation 1 the variable X_i represents the original value of the feature. Then the minimum value of the feature X_{min} is subtracted from the original feature and divided by the difference between the maximum X_{max} and a minimum X_{min} result of the feature.

4.2. Feature Extraction

After pre-processing the raw data, the data is in good shape to extract features. The dataset is distributed in 13 categories each representing a different DDoS attack. These different attacks are NTP, UDP, DNS, LDAP, MSSQL, NetBIOS, SNMP, SSDP, SYN, UDP-Lag, Web-DDoS, TFTP, and Portmap attacks. The dataset CIC-DDoS2019 [48] is a combination of numerous numeric and object types from which only numerical types are extracted. This step is necessary to ensure improvement in the efficiency of classification models.

4.3. Oversampling

Oversampling is achieved by increasing the minority classes using the Synthetic Minority Oversampling Technique (SMOTE) [60]. SMOTE is a statistical technique for increasing the number of instances in a dataset such that all class labels have the same number of instances. It generates new instances from existing minority cases. First, the minority class instance is randomly selected by SMOTE and finds its k nearest minority class neighbors. The synthetic instance is then created by choosing one of the k nearest neighbors b at random and connecting a and b to form a line segment in the feature space. The synthetic instances are generated as a convex combination of the two chosen instances a and b.

4.4. Classification Models

For the classification deep learning algorithms are used such as Gated Recurrent Unit (GRU), recurrent neural network(RNN), and machine learning algorithms are Naive Bayes (NB), Sequential Minimal Optimization (SMO). Below, a brief introduction is provided to each algorithm.

1. **Gated Recurrent Unit (GRU)** aims to solve the vanishing gradient problem which comes with a standard recurrent neural network. GRU can also be considered as a variation on the LSTM because both are designed similarly and, in some cases, produce equally excellent results. In this research, the GRU model is used because it trains the dataset faster, executes faster, and uses less memory.

2. **Recurrent Neural Network (RNN)** is a generalization of a feedforward neural network that has internal memory. RNN is recurrent as it performs the same function for every input of data while the output of the current input depends on the past one computation. After producing the output, it is copied and sent back into the recurrent network [61]. In this research, the RNN model is used because the size of the dataset CICDDoS2019 is large and even if the dataset input size is larger, the model size does not increase.

3. **Naive Bayes (NB)** is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. So, this classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.
4. Bayes theorem checks probability $P(c|x)$ from $P(c)$, $P(x)$ and $P(x|c)$ as shown in equation 2 from [62] and $P(c|x)$ is the posterior probability of class $(c, target)$ given predictor (x, attributes), $P(c)$ is the prior probability of a class, $P(x|c)$ is the likelihood which is the probability of predictor given class and $P(x)$ is the prior probability of predictor [63]. In this research, the NB model is used because the dataset CICDDoS2019 .
$$\frac{P(x|c)P(c)}{P(x)} \tag{2}$$
5. **Sequential Minimal Optimization (SMO)** is an algorithm for solving the quadratic programming (QP) problem that arises during the training of support vector machines (SVM). Instead of an SVM algorithm that uses numerical QP as an inner loop, SMO uses an analytic QP step [64].In this research, the SMO algorithm is used because it is a very fast algorithm and very robust with a high input dimension dataset.

5. Evaluation and Results

The results are evaluated based on Accuracy, Precision, Recall, and F1-score. Accuracy is commonly taken as the performance evaluator in most cases but in the case of dataset imbalance problem, F1-score is the optimal choice to evaluate the performance of the classifier. F1-score is the harmonic mean of precision and recall.

5.1. NTP attacks

The results for NTP DDoS attacks are shown in Table 2 in which the highest accuracy achieved is 99.52% by using the GRU algorithm. Other algorithms: RNN, SMO, and NB achieve the accuracy of 99.35%, 98.89%, and 96.65%. In the case of naive Bayes, the accuracy is 96.65% which is low as compared to other algorithms because it needs more data instances. Other algorithms used are not dependent on the quantity of data. Figure 3a presents the accuracy convergence with respect to epochs and the highest

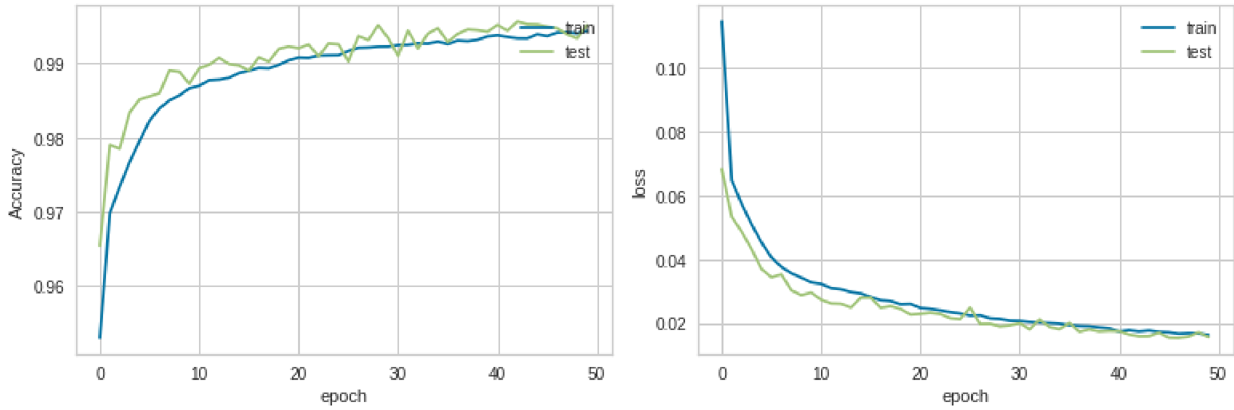
Table 2: Algorithms Proficiency Metrics for Detecting NTP attacks

Model	Accuracy(%)	Precision(%)	Recall(%)	F1-score(%)
GRU	**99.52**	**99.31**	**97.12**	**98.37**
RNN	99.35	99.45	96.50	97.07
SMO	98.89	99.0	98.91	98.90
NB	96.65	97.3	96.75	96.83

accuracy of 99.5% is achieved at 42th epoch. Training accuracy curve begins at 97.5% and goes up to 99.5% and after that the convergence of training accuracy becomes stable. Test accuracy starts at 99.87% and goes up to 99.5%. It slightly went down at the 6th epoch. Figure 3 depicts the convergence of the accuracy with epochs and it achieves the lowest loss of 0.01% at the 44th epoch. Training loss starts at 0.09% and goes down to 0.01%. Then the convergence of training loss becomes stable as shown in Figure 3b.

5.2. UDP attacks

In the case of UDP attacks, the highest accuracy of 99.69% is achieved by using GRU and RNN classification algorithm and the other models were also very accurate in which SMO and NB achieved an accuracy of 99.60% and 99.20% as shown in Table 3. Figure 4a presents the accuracy convergence with respect to epochs and the highest accuracy of 99.76% is achieved at 42th epoch. Training accuracy curve begins at 98.7% and goes up to 99.8% and after that the convergence of training accuracy becomes stable. Test accuracy starts at 99.87% and goes up to 99.7%. It slightly went down at the 15th epoch. Figure 4 depicts the convergence of the accuracy with epochs and it achieves the lowest loss of 0.01% at the 46th epoch.



(a) Model accuracy

(b) Model Loss

Figure 3: Model Accuracy and Loss of DDoS Malware with Respect to NTP Attacks

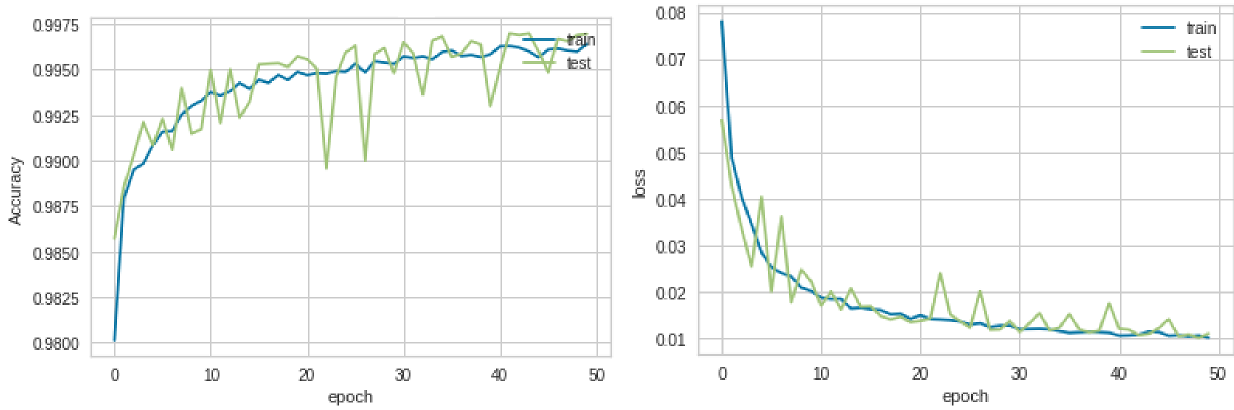
Table 3: Algorithms Proficiency Metrics for Detecting of UDP attacks

Model	Accuracy(%)	Precision(%)	Recall(%)	F1-score(%)
GRU	**99.69**	**98.1**	**98.21**	**98.3**
RNN	99.69	98.41	97.94	98.35
SMO	99.60	99.61	99.61	99.61
NB	99.20	99.31	99.24	99.29

Training loss starts at 0.07% and goes down to 0.01%. Then the convergence of training loss becomes stable as shown in fig 4b.

5.3. DNS attacks

In Table 4, it can be seen that by using the SMO algorithm the highest accuracy achieved is 99.75% and with other algorithm techniques such as GRU, RNN and NB the accuracy achieved is 99.51%, 99.72% and 99.35% for DNS attacks. Figure 5a presents the accuracy convergence with respect to epochs and the highest accuracy of 99.72% is achieved at 46th epoch. Training accuracy curve begins at 98.25% and goes up to 99.6% and after that the convergence of training accuracy becomes stable. Test accuracy starts at 98.65% and goes up to 99.65%. It slightly went down at the 20th epoch. Figure 5 depicts the convergence of the accuracy with epochs and it achieves the lowest loss of 0.01% at the 48th epoch. Training loss starts at 0.06% and goes down to 0.01%. Then the convergence of training loss becomes stable as shown in fig



(a) Model accuracy (b) Model Loss

Figure 4: Model Accuracy and Loss of DDoS Malware with Respect to UDP Attacks

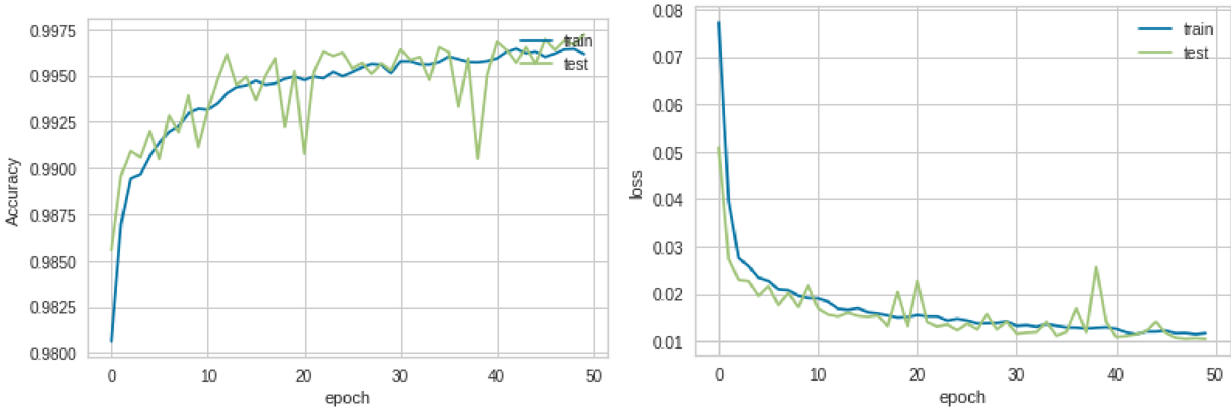
Table 4: Algorithms Proficiency Metrics for Detecting of DNS attacks

Model	**Accuracy(%)**	**Precision(%)**	**Recall(%)**	**F1-score(%)**
GRU	99.51	98.42	97.21	97.20
RNN	99.72	98.12	99.59	99.32
SMO	**99.75**	**99.80**	**99.80**	**99.80**
NB	99.35	99.40	99.40	99.40

5b.

5.4. LDAP attacks

For LDAP attacks, the highest accuracy achieved is 99.96% by using the SMO model. The remaining algorithms were GRU, RNN, and NB through which the achieved accuracy is 99.95%, 99.94%, and 99.82% as shown in Table 5. Figure 6a presents the accuracy convergence with respect to epochs and the highest accuracy of 99.95% is achieved at 4th epoch. Training accuracy curve begins at 99% and goes up to 99.9% and after that the convergence of training accuracy becomes stable. Test accuracy starts at 98.1% and goes up to 99.95%. Figure 6 depicts the convergence of the accuracy with epochs and it achieves the lowest loss of below 0.01% at the 15th epoch. Training loss starts at 0.06% and goes down to 0.005%. Then the convergence of training loss becomes stable as shown in fig 6b.



(a) Model accuracy (b) Model Loss

Figure 5: Model Accuracy and Loss of DDoS Malware with Respect to DNS attacks

Table 5: Algorithms Proficiency Metrics for detecting of LDAP attacks

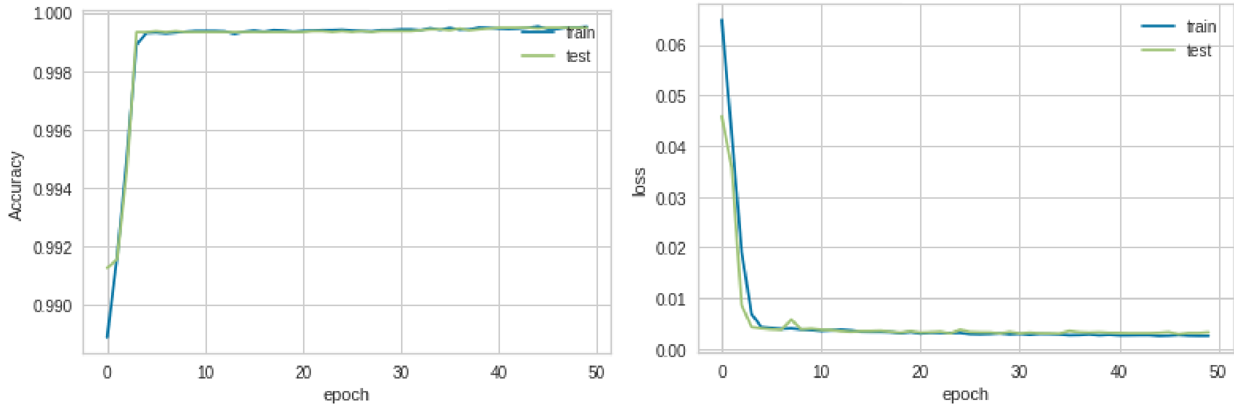
Model	**Accuracy(%)**	**Precision(%)**	**Recall(%)**	**F1-score(%)**
GRU	99.95	99.16	99.88	99.32
RNN	99.94	99.40	99.77	99.48
SMO	**99.96**	**99.71**	**99.91**	**99.87**
NB	99.82	99.80	99.80	99.80

5.5. MSSQL attacks

The MSSQL attack results are shown in Table 6 in which the highest accuracy of 99.94% is achieved by using the SMO algorithm and with GRU, RNN, and NB algorithm the accuracy achieved is 99.82% and 99.83%. Figure 7a presents the accuracy convergence with respect to epochs and the highest accuracy

Table 6: Algorithms Proficiency Metrics for Detecting of MSSQL attacks

Model	**Accuracy(%)**	**Precision(%)**	**Recall(%)**	**F1-score(%)**
GRU	99.82	98.11	99.10	99.06
RNN	99.83	98.04	99.55	99.31
SMO	**99.94**	**99.90**	**99.90**	**99.90**
NB	99.83	99.80	99.80	99.80

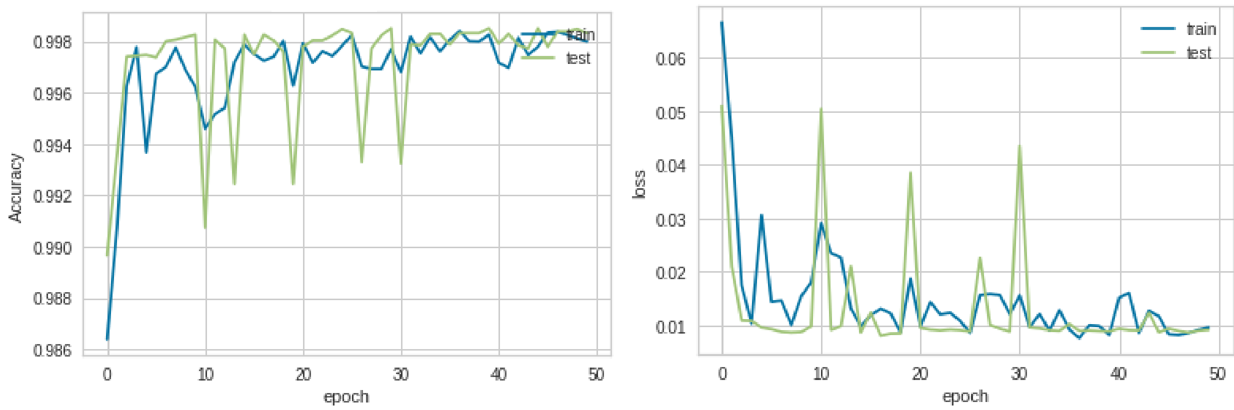


(a) Model accuracy

(b) Model Loss

Figure 6: Model Accuracy and Loss of DDoS Malware with Respect to LDAP Attacks

of 99.83% is achieved at 8th epoch. Training accuracy curve begins at 98.4% and goes up to 99.8% and after that the convergence of training accuracy becomes stable. Test accuracy starts at 98.1% and goes up to 99.95%. Figure 7 depicts the convergence of the accuracy with epochs and it achieves the lowest loss of below 0.01% at the 48th epoch. Training loss starts at 0.08% and goes down to 0.01%. Then the convergence of training loss becomes stable as shown in fig 7b.



(a) Model accuracy

(b) Model Loss

Figure 7: Model Accuracy and Loss of DDoS Malware with Respect to MSSQL Attacks

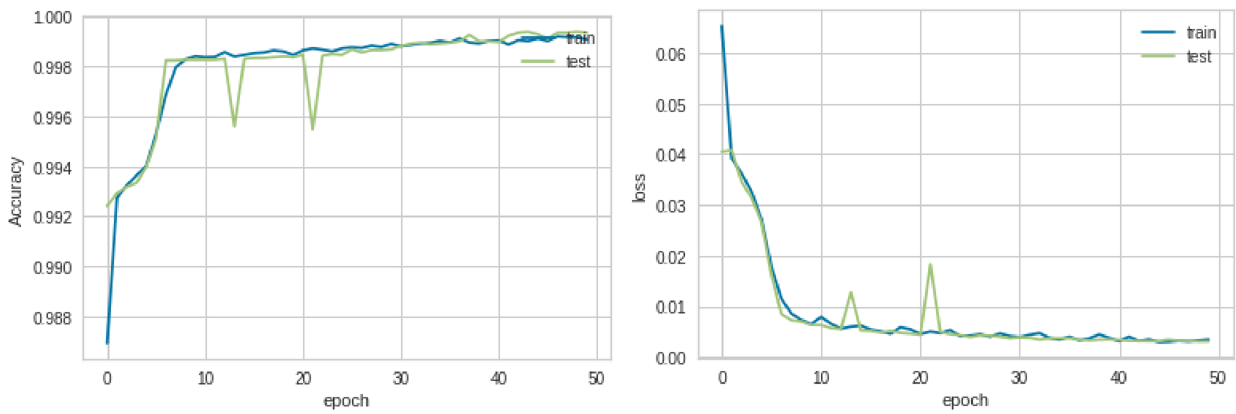
5.6. *NetBIOS attacks*

In the case of NetBIOS attacks, the highest accuracy of 99.94% is achieved by using the GRU algorithm. By using other algorithms: RNN, SMO and NB achieved the accuracy of 99.89%, 99.93%, and 99.87% as shown in Table 7. Figure 8a presents the accuracy convergence concerning epochs and the highest accuracy

Table 7: Algorithms Proficiency Metrics for Detecting of NetBIOS attacks

Model	Accuracy(%)	Precision(%)	Recall(%)	F1-score(%)
GRU	**99.94**	**99.11**	**99.90**	**99.49**
RNN	99.89	98.10	99.81	99.10
SMO	99.93	99.90	99.90	99.90
NB	99.87	99.90	99.90	99.90

of 99.94% is achieved at the 35th epoch. The training accuracy curve begins at 98.8% and goes up to 99.9% and after that the convergence of training accuracy becomes stable. Test accuracy starts at 99.3% and goes up to 99.9%. Figure 8 depicts the convergence of the accuracy with epochs and it achieves the lowest loss of below 0.01% at the 48th epoch. Training loss starts at 0.06% and goes down to 0.004%. Then the convergence of training loss becomes stable as shown in fig 8b.



(a) Model accuracy

(b) Model Loss

Figure 8: Model Accuracy and Loss of DDoS Malware with Respect to NETBIOS Attacks

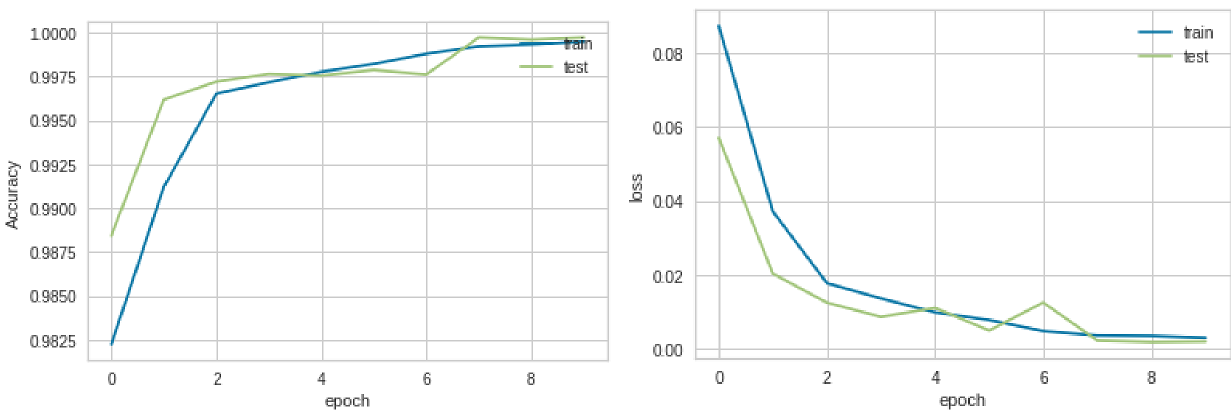
5.7. *SNMP attacks*

For SNMP attacks, the highest accuracy achieved is 99.99% by using the SMO algorithm, and by using GRU, RNN, and NB classification techniques the achieved accuracy is 99.97%,99.79% and 99.87% as shown in Table 8. Figure 9a presents the accuracy convergence with respect to epochs and the highest

Table 8: Algorithms Proficiency Metrics for Detecting of SNMP attacks

Model	**Accuracy(%)**	**Precision(%)**	**Recall(%)**	**F1-score(%)**
GRU	99.97	99.35	99.55	99.67
RNN	99.79	99.42	96.15	97.25
SMO	**99.99**	**99.97**	**99.97**	**99.97**
NB	99.87	99.90	99.90	99.90

accuracy of 99.97% is achieved at 9th epoch. Training accuracy curve begins at 98.25% and goes up to 99.9% and after that the convergence of training accuracy becomes stable. Test accuracy starts at 98.8% and goes up to 99.9%. Figure 9 depicts the convergence of the accuracy with epochs and it achieves the lowest loss of below 0.01% at the 8th epoch. Training loss starts at 0.09% and goes down to 0.006%. Then the convergence of training loss becomes stable as shown in fig 9b.



(a) Model accuracy

(b) Model Loss

Figure 9: Model Accuracy and Loss of DDoS Malware with Respect to SNMP Attacks

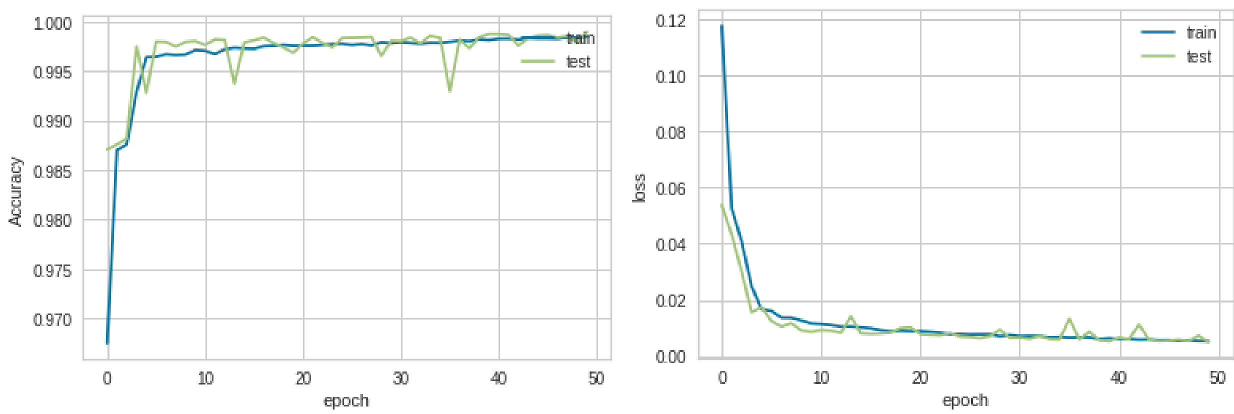
5.8. SSDP attacks

The SSDP attacks results are shown in Table 9 in which the highest accuracy achieved is 99.90% using the GRU algorithm and by using others with algorithms i.e., RNN, SMO and NB the accuracy of 99.87%, 99.89%, and 99.78% are achieved respectively. Figure 10a presents the accuracy convergence with respect

Table 9: Algorithms Proficiency Metrics for Detecting of SSDP attacks

Model	Accuracy(%)	Precision(%)	Recall(%)	F1-score(%)
GRU	**99.91**	**99.83**	**99.79**	**99.69**
RNN	99.87	99.05	99.68	99.81
SMO	99.89	99.90	99.90	99.90
NB	99.78	99.81	99.80	99.80

to epochs and the highest accuracy of 99.9% is achieved at 45th epoch. Training accuracy curve begins at 96.7% and goes up to 99.8% and after that the convergence of training accuracy becomes stable. Test accuracy starts at 98.7% and goes up to 99.9%. Figure 10 depicts the convergence of the accuracy with epochs and it achieves the lowest loss of below 0.01% at the 48th epoch. Training loss starts at 0.12% and goes down to 0.01%. Then the convergence of training loss becomes stable as shown in fig 10b.



(a) Model accuracy

(b) Model Loss

Figure 10: Model Accuracy and Loss of DDoS Malware with Respect to SSDP Attacks

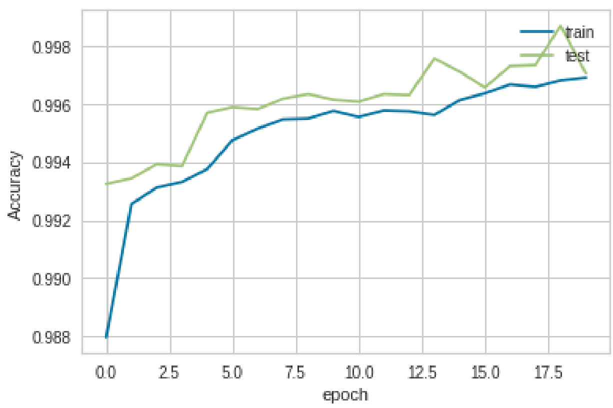
5.9. SYN attacks

In the case of SYN attacks, as shown in Table 10 the highest accuracy of 99.98% is achieved by using the SMO algorithm and by other algorithms such as GRU, RNN and NB, the achieved accuracy is 99.69%, 99.7%, and 99.95% respectively. Table 10 shows these results. Figure 11a presents the accuracy

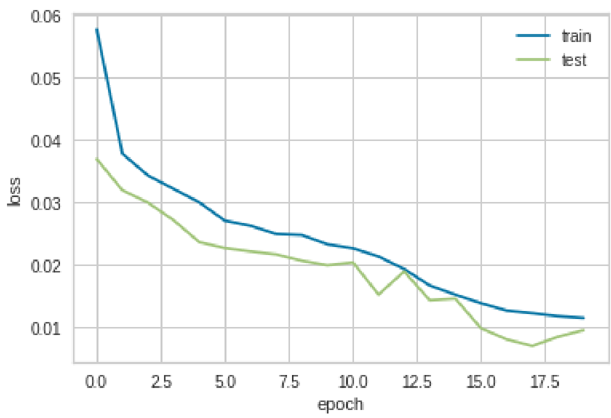
Table 10: Algorithms Proficiency Metrics for Detecting of SYN attacks

Model	Accuracy(%)	Precision(%)	Recall(%)	F1-score(%)
GRU	99.69	99.11	92.43	96.35
RNN	99.70	99.50	92.24	96.31
SMO	**99.98**	**99.94**	**99.94**	**99.94**
NB	99.95	99.91	99.91	99.91

convergence with respect to epochs and the highest accuracy of 99.7% is achieved at 15th epoch. Training accuracy curve begins at 98.8% and goes up to 99.7% and after that the convergence of training accuracy becomes stable. Test accuracy starts at 99.3% and goes up to 99.9%. Figure 11 depicts the convergence of the accuracy with epochs and it achieves the lowest loss of 0.01% at the 17th epoch. Training loss starts at 0.06% and goes down to 0.01%. Then the convergence of training loss becomes stable as shown in Figure 11b.



(a) Model accuracy



(b) Model Loss

Figure 11: Model Accuracy and Loss of SYN Attacks Detection

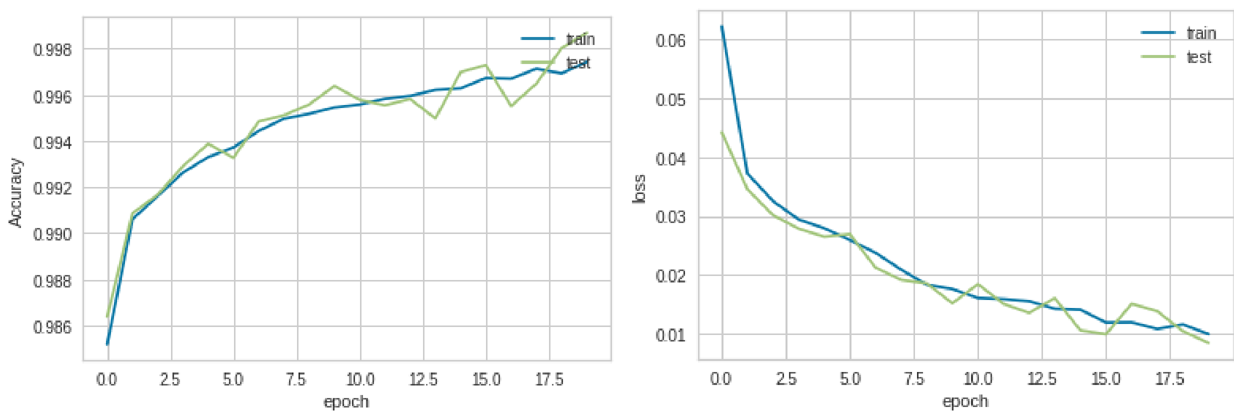
5.10. *UDP-Lag attacks*

As shown in Table 11, the highest accuracy achieved is calculated by using RNN algorithm that is 99.87%. Other algorithms also achieved good results in which GRU algorithm achieved 99.55% accuracy, SMO achieved 99.86% accuracy and NB achieved 96.63% accuracy respectively. Figure 12a presents the

Table 11: Algorithms Proficiency Metrics for Detecting of UDP-Lag attacks

Model	Accuracy(%)	Precision(%)	Recall(%)	F1-score(%)
GRU	**99.87**	**99.57**	**98.67**	**98.36**
RNN	99.55	97.60	90.56	94.22
SMO	99.86	99.9	99.91	99.90
NB	96.63	96.51	96.60	96.60

accuracy convergence with respect to epochs and the highest accuracy of 99.87% is achieved at 18th epoch. Training accuracy curve begins at 98.86% and goes up to 99.7% and after that the convergence of training accuracy becomes stable. Test accuracy starts at 98.6% and goes up to 99.85%. Figure 12 depicts the convergence of the accuracy with epochs and it achieves the lowest loss of 0.01% at the 18th epoch. Training loss starts at 0.06% and goes down to 0.01%. Then the convergence of training loss becomes stable as shown in fig 12b.



(a) Model accuracy (b) Model Loss

Figure 12: Model Accuracy and Loss of DDoS Malware with Respect to UDP-Lag Attacks

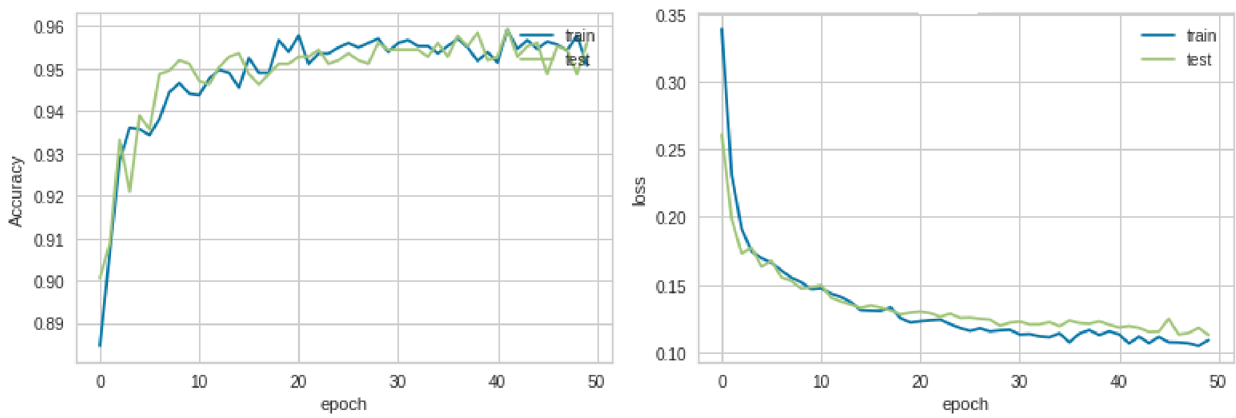
5.11. Web-DDoS attacks

As shown in Table 12, SMO algorithm achieved the highest accuracy of 96.62% for detecting Web-DDoS attacks. Other algorithms such as GRU, RNN and NB achieved the accuracy 95.11%, 95.6% and 68.8% respectively. For Web-DDoS attacks the accuracy achieved with naive bayes classifier is 96.65% which is low as compared to other algorithms because it needs more data instances. Other algorithms used are not dependent on quantity of data. Figure 13a presents the accuracy convergence with respect to epochs

Table 12: Algorithms Proficiency Metrics for Detecting of WebDDoS attacks

	Accuracy(%)	**Precision(%)**	**Recall(%)**	**F1-score(%)**
GRU	95.11	96.14	**99.32**	**97.41**
RNN	95.60	97.44	99.04	98.36
SMO	**96.62**	**96.70**	**96.60**	**96.03**
NB	68.80	92.00	68.90	75.10

and the highest accuracy of 96% is achieved at 40th epoch. Training accuracy curve begins at 89% and goes up to 95.55% and after that the convergence of training accuracy becomes stable. Test accuracy starts at 90% and goes up to 96.0%. Figure 13 depicts the convergence of the accuracy with epochs and it achieves the lowest loss of 0.10% at the 48th epoch. Training loss starts at 0.35% and goes down to 0.1%. Then the convergence of training loss becomes stable as shown in fig 13b.



(a) Model accuracy

(b) Model Loss

Figure 13: Model Accuracy and Loss of DDoS Malware with Respect to Web-DDoS Attacks

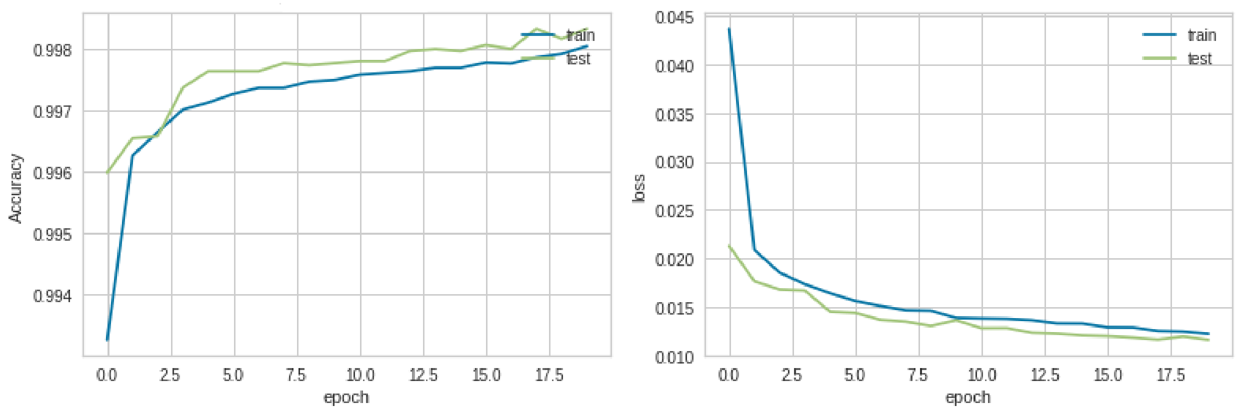
5.12. *TFTP attacks*

The result of TFTP are represented in Table 13 with respect to 4 different classification models. The highest accuracy that is achieved is 99.97% by using the SMO algorithm. The other algorithms: GRU, RNN, and NB also calculated excellent results in which the accuracy is 99.83% for GRU, 99.78% for RNN, and 98.92% for NB algorithm respectively. Figure 14a presents the accuracy convergence with respect

Table 13: Algorithms Proficiency Metrics for Detecting of TFTP attacks

Model	Accuracy(%)	Precision(%)	Recall(%)	F1-score(%)
GRU	99.83	99.08	86.16	92.33
RNN	99.78	98.36	83.18	90.17
SMO	**99.97**	**99.94**	**99.96**	**99.92**
NB	98.92	99.40	98.90	99.10

to epochs and the highest accuracy of 99.83% is achieved at 18th epoch. Training accuracy curve begins at 99.3% and goes up to 99.8% and after that the convergence of training accuracy becomes stable. Test accuracy starts at 99.6% and goes up to 99.83%. Figure 14 depicts the convergence of the accuracy with epochs and it achieves the lowest loss of 0.013% at the 17th epoch. Training loss starts at 0.04% and goes down to 0.013%. Then the convergence of training loss becomes stable as shown in fig 14b.



(a) Model accuracy

(b) Model Loss

Figure 14: Model Accuracy and Loss of DDoS Malware with Respect to TFTP Attacks

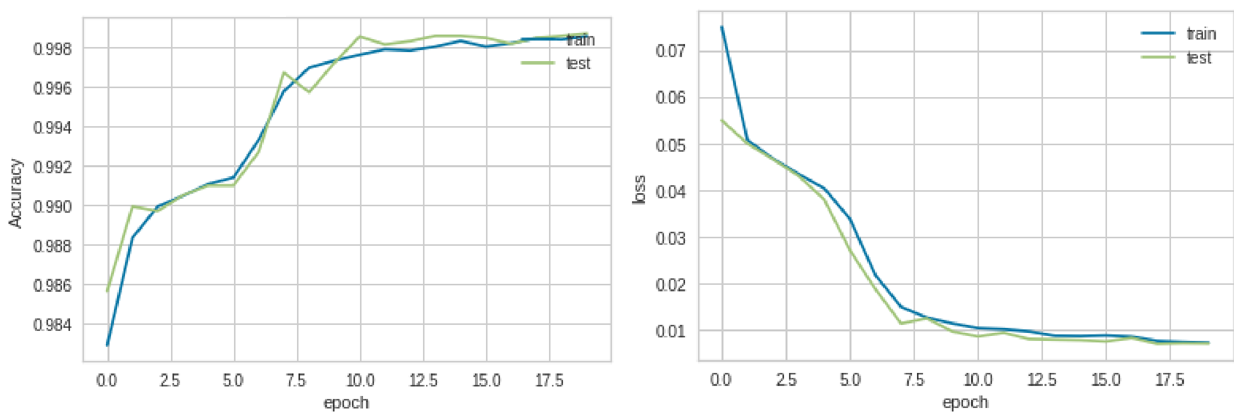
5.13. Portmap attacks

The result of portmap attacks are represented in Table 14 with respect to 4 different classification models. The highest accuracy that is achieved is 99.87% by using GRU algorithm. The other algorithms: RNN, SMO and NB also calculated excellent results in which the accuracy is 99.80% for RNN, 99.86% for SMO and 99.1% for NB algorithm respectively. Figure 15a presents the accuracy convergence with respect to

Table 14: Algorithms Proficiency Metrics for Detecting of Portmap Attacks

Model	**Accuracy(%)**	**Precision(%)**	**Recall(%)**	**F1-score(%)**
GRU	**99.87**	**98.13**	**99.45**	**99.63**
RNN	99.80	97.44	99.07	98.30
SMO	99.86	97.50	99.65	98.50
NB	99.18	85.40	99.70	92.00

epochs and the highest accuracy of 99.87% is achieved at 16th epoch. Training accuracy curve begins at 99.4% and goes up to 99.81% and after that the convergence of training accuracy becomes stable. Test accuracy starts at 99.6% and goes up to 99.87%. It slightly went down at the 17th epoch. Figure 15b depicts the convergence of the accuracy with epochs and it achieves the lowest loss of 0.013% at the 17th epoch. Training loss starts at 0.045% and goes down to 0.014%. Then the convergence of training loss becomes stable as shown in Figure 15.



(a) Model accuracy

(b) Model Loss

Figure 15: Model Accuracy and Loss of DDoS malware with respect to Portmap attacks

6. Comparative Analysis and Discussion

In Table 15, the precision, and recall of different models are compared with other state-of-the-art studies that utilize the CICDDoS2019 dataset for DDoS attack detection and identification. In [24], the ID3 algorithm was used to achieve the highest precision of 78% while other algorithms such as RF, NB, and logistic regression achieved the precision 77%, 41%, and 25% respectively. Similarly, the research [25] evaluated the average performance of classifiers and achieved the highest precision of 96.9% by using the bagging classifier. This research also obtained the results by using other classifiers such as Bayes net, KNN, SMO, and simple logistic which achieved the precision 96.2%, 96.7%, 93.9%, and 93.1% respectively. The research [65] combined DDoS simulators, BoNeSi and SlowHTTPTest with the CICDDoS2019 dataset. They achieved an accuracy of 98.9% using the LSTM algorithm and 99.9% using the CNN algorithm. In [26], 2 scenarios were observed and in the second scenario, they used the dataset CICDDoS2019. By using the CICDDoS2019 dataset the highest precision achieved was 97.89% using the LSTM-Fuzzy algorithm. Other algorithms were also used such as KNN, LSTM-2, MLP, PSO-DS, and SVM which achieved the precision of 89.27%, 96.61%, 94.08%, 81.19%, and 97.74% respectively. In the other comparisons, it is observed that they also achieved good results but our research achieves the highest accuracy of 99.97% using the GRU algorithm for SNMP attacks as shown in Table 16.

Table 15: Comparison of the *DIDDOS* with State-of-the-art Studies

Paper	**Dataset**	**Precision (%)**	**Recall (%)**
[24]	CICDDoS2019	78.00	65.00
[25]	CICDDoS2019	96.90	96.40
[26]	CICDDoS2019	97.89	93.13
This approach	CICDDoS2019	99.83	99.79

In this research, the CICDDoS2019 dataset is passed from a series of steps that include pre-processing, feature extraction, resolving the oversampling problem, and then the data was split into 11 different attack files. Due to a very large dataset, a part of the CICDDoS2019 dataset is used for each attack in this experimentation. Firstly in the pre-processing stage, NAN values, duplicate rows of data are removed, and then the data is normalized with MinMax scaling because the data was less ambiguous with low variance. Then the oversampling problem is solved on WEKA [66] platform by using a supervised classification technique called SMOTE [60]. This technique analyzes the data and generates data instances of the minority class of

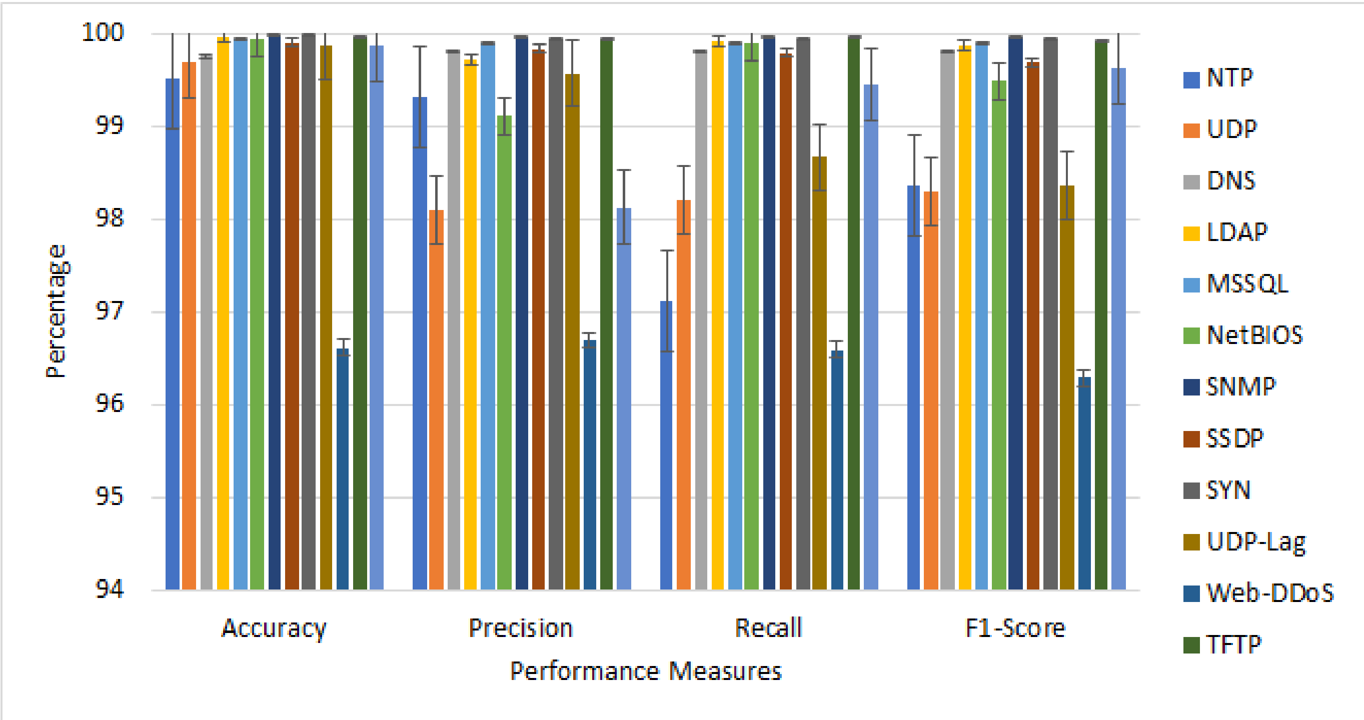


Figure 16: Comparison of Performance Measures with Respect to Each Attack

data to balance the data and avoid over-fitting problems.

7. Conclusion

In this research, an approach *DIDDOS* is proposed to detect and identify DDoS attacks over the network. The *DIDDOS* is evaluated by using the state-of-the-art CICDDoS2019 dataset by using deep learning algorithms i.e., GRU and RNN as well as conventional machine learning algorithms NB and SMO. The experimental results demonstrated that the *DIDDOS* is most efficient for detecting and identifying DDoS attacks. From the experimental result analysis, it is evident that our proposed approach gives very effective performance results based on accuracy, precision, recall, and F1-score. The highest accuracy achieved is 99.91% by using the GRU algorithm in case of an SSDP attack and an average of 99.7 for all other attacks. In addition to this, for SSDP attacks, the precision, recall, and F1-score are 99.83%, 99.79%, and 99.69% respectively. For future work, we plan to use this dataset in an intrusion detection system and that network module can be upgraded to an intrusion prevention system so that the DDoS attacks can be detected and prevented. By the addition of more malware samples, in the near future, we can also make a generic dataset that will contain all different categories and types of malware information. This step will allow different areas to use our generic dataset instead of using multiple datasets for each malware classification. This will contribute a positive security service to the world and help to increase the prevention of DDoS attacks.

References

[1] A. Benmoussa, A. el Karim Tahari, C. A. Kerrache, N. Lagraa, A. Lakas, R. Hussain, F. Ahmad, Msidn: Mitigation of sophisticated interest flooding-based ddos attacks in named data networking, Future Generation Computer Systems 107 (2020) 293–306.

[2] M. Shafiq, Z. Tian, A. K. Bashir, X. Du, M. Guizani, Iot malicious traffic identification using wrapper-based feature selection mechanisms, Computers & Security (2020) 101863.

[3] I. U. Khan, I. M. Qureshi, M. A. Aziz, T. A. Cheema, S. B. H. Shah, Smart iot control-based nature inspired energy efficient routing protocol for flying ad hoc network (fanet), IEEE Access 8 (2020) 56371–56378.

[4] S. I. Imtiaz, S. ur Rehman, A. R. Javed, Z. Jalil, X. Liu, W. S. Alnumay, Deepamd: Detection and identification of android malware using high-efficient deep artificial neural network, Future Generation Computer Systems 115 844–856.

[5] C. Iwendi, Z. Jalil, A. R. Javed, T. Reddy, R. Kaluri, G. Srivastava, O. Jo, Keysplitwatermark: Zero watermarking algorithm for software protection against cyber-attacks, IEEE Access 8 (2020) 72650–72660.

[6] M. Mittal, C. Iwendi, S. Khan, A. Rehman Javed, Analysis of security and energy efficiency for shortest route discovery in low-energy adaptive clustering hierarchy protocol using levenberg-marquardt neural network and gated recurrent unit for intrusion detection system, Transactions on Emerging Telecommunications Technologies (2020) e3997.

[7] M. Alazab, R. Layton, R. Broadhurst, B. Bouhours, Malicious spam emails developments and authorship attribution, in: 2013 Fourth Cybercrime and Trustworthy Computing Workshop, IEEE, 2013, pp. 58–68.

[8] M. Alazab, R. Broadhurst, An analysis of the nature of spam as cybercrime, in: Cyber-Physical Security, Springer, 2017, pp. 251–266.

[9] M. Alazab, S. Venkatraman, P. Watters, M. Alazab, Information security governance: the art of detecting hidden malware, in: IT security governance innovations: theory and research, IGI Global, 2013, pp. 293–315.

[10] A. R. Javed, M. O. Beg, M. Asim, T. Baker, A. H. Al-Bayatti, Alphalogger: detecting motion-based side-channel attack using smartphone keystrokes, Journal of Ambient Intelligence and Humanized Computing (2020) 1–14.

[11] M. Mittal, S. Vijayal, Detection of attacks in iot based on ontology using sparql, in: 2017 7th International Conference on Communication Systems and Network Technologies (CSNT), IEEE, 2017, pp. 206–211.

[12] A. Basit, M. Zafar, X. Liu, A. R. Javed, Z. Jalil, K. Kifayat, A comprehensive survey of ai-enabled phishing attacks detection techniques, Telecommunication Systems (2020) 1–16.

[13] S. P. RM, P. K. R. Maddikunta, M. Parimala, S. Koppu, T. Reddy, C. L. Chowdhary, M. Alazab, An effective feature engineering for dnn using hybrid pca-gwo for intrusion detection in iomt architecture, Computer Communications (2020).

[14] S. S. Silva, R. M. Silva, R. C. Pinto, R. M. Salles, Botnets: A survey, Computer Networks 57 (2) (2013) 378–403.

[15] A. Rehman Javed, Z. Jalil, S. Atif Moqurrab, S. Abbas, X. Liu, Ensemble adaboost classifier for accurate and fast detection of botnet attacks in connected vehicles, Transactions on Emerging Telecommunications Technologies (2020) e4088.

[16] C. Kolias, G. Kambourakis, A. Stavrou, J. Voas, Ddos in the iot: Mirai and other botnets, Computer 50 (7) (2017) 80–84.

[17] P. J. Criscuolo, Distributed denial of service: Trin00, tribe flood network, tribe flood network 2000, and stacheldraht ciac-2319, Tech. rep., California Univ Livermore Radiation Lab (2000).

[18] R. M. A. Ujjan, Z. Pervez, K. Dahal, A. K. Bashir, R. Mumtaz, J. González, Towards sflow and adaptive polling sampling for deep learning based ddos detection in sdn, Future Generation Computer Systems 111 (2020) 763–779.

[19] M. Shakil, A. Fuad Yousif Mohammed, R. Arul, A. K. Bashir, J. K. Choi, A novel dynamic framework to detect ddos in sdn using metaheuristic clustering, Transactions on Emerging Telecommunications Technologies (2019) e3622.

[20] K.-N. Tran, M. Alazab, R. Broadhurst, et al., Towards a feature rich model for predicting spam emails containing malicious attachments and urls (2014).

[21] M. Shafiq, Z. Tian, Y. Sun, X. Du, M. Guizani, Selection of effective machine learning algorithm and bot-iot attacks traffic identification for internet of things in smart city, Future Generation Computer Systems 107 (2020) 433–442.

[22] J. Nazario, Ddos attack evolution, Network Security 2008 (7) (2008) 7–10.

[23] J. Mirkovic, P. Reiher, A taxonomy of ddos attack and ddos defense mechanisms, ACM SIGCOMM Computer Communication Review 34 (2) (2004) 39–53.

[24] I. Sharafaldin, A. H. Lashkari, S. Hakak, A. A. Ghorbani, Developing realistic distributed denial of service (ddos) attack dataset and taxonomy, in: 2019 International Carnahan Conference on Security Technology (ICCST), IEEE, 2019, pp. 1–8.

[25] Y. S. Hussain, Network intrusion detection for distributed denial-of-service (ddos) attacks using machine learning classification techniques (2020).

[26] M. P. Novaes, L. F. Carvalho, J. Lloret, M. L. Proença, Long short-term memory and fuzzy logic for anomaly detection and mitigation in software-defined network environment, IEEE Access 8 (2020) 83765–83781.

[27] The 15 top ddos statistics you should know in 2020, `https://cybersecurityventures.com/the-15-top-ddos-statistics-you-should-know-in-2020/`, accessed: 2020-03-12.

[28] The caida ucsd ”ddos attack 2007” dataset, `http://www.caida.org/data/passive/ddos-20070804dataset.xml`, accessed: 2020-03-12.

[29] A. Asosheh, N. Ramezani, A comprehensive taxonomy of ddos attacks and defense mechanism applying in a smart classification, WSEAS Transactions on Computers 7 (4) (2008) 281–290.

[30] A. Bhardwaj, G. Subrahmanyam, V. Avasthi, H. Sastry, S. Goundar, Ddos attacks, new ddos taxonomy and mitigation solutions—a survey, in: 2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES), IEEE, 2016, pp. 793–798.

[31] M. Masdari, M. Jalali, A survey and taxonomy of dos attacks in cloud computing, Security and Communication Networks 9 (16) (2016) 3724–3751.

[32] U. A. Butt, M. Mehmood, S. B. H. Shah, R. Amin, M. W. Shaukat, S. M. Raza, D. Y. Suh, M. Piran, et al., A review of machine learning algorithms for cloud computing security, Electronics 9 (9) (2020) 1379.

[33] K. Singh, P. Singh, K. Kumar, Application layer http-get flood ddos attacks: Research landscape and challenges, Computers & security 65 (2017) 344–372.

[34] A. Patel, M. Taghavi, K. Bakhtiyari, J. C. JúNior, An intrusion detection and prevention system in cloud computing: A systematic review, Journal of network and computer applications 36 (1) (2013) 25–41.

[35] A. R. Javed, M. Usman, S. U. Rehman, M. U. Khan, M. S. Haghighi, Anomaly detection in automated vehicles using multistage attention-based convolutional neural network, IEEE Transactions on Intelligent Transportation Systems (2020).

[36] Q. Zhao, J. Sun, S. Zhang, A hybrid and hierarchical nids paradigm utilizing naive bayes classifier, in: Canadian Conference on Electrical and Computer Engineering 2004 (IEEE Cat. No. 04CH37513), Vol. 1, IEEE, 2004, pp. 145–148.

[37] M. Roesch, et al., Snort: Lightweight intrusion detection for networks., in: Lisa, Vol. 99, 1999, pp. 229–238.

[38] D. Jing, H.-B. Chen, Svm based network intrusion detection for the unsw-nb15 dataset, in: 2019 IEEE 13th International Conference on ASIC (ASICON), IEEE, 2019, pp. 1–4.

[39] J. H. Anajemba, C. Iwendi, M. Mittal, T. Yue, Improved advance encryption standard with a privacy database structure for iot nodes, in: 2020 IEEE 9th International Conference on Communication Systems and Network Technologies (CSNT), IEEE, 2020, pp. 201–206.

[40] M. Mittal, L. K. Saraswat, C. Iwendi, J. H. Anajemba, A neuro-fuzzy approach for intrusion detection in energy efficient sensor routing, in: 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), IEEE, 2019, pp. 1–5.

[41] C. Yin, Y. Zhu, J. Fei, X. He, A deep learning approach for intrusion detection using recurrent neural networks, Ieee Access 5 (2017) 21954–21961.

[42] P. Du, G. Roussos, Adaptive time slotted channel hopping for wireless sensor networks, in: 2012 4th computer science and electronic engineering conference (CEEC), IEEE, 2012, pp. 29–34.

[43] J. R. Jain, A. Asaduzzaman, A novel data logging framework to enhance security of cloud computing, in: SoutheastCon 2016, IEEE, 2016, pp. 1–6.

[44] F. Meng, Y. Fu, F. Lou, Z. Chen, An effective network attack detection method based on kernel pca and lstm-rnn, in: 2017 International Conference on Computer Systems, Electronics and Control (ICCSEC), IEEE, 2017, pp. 568–572.

[45] J. Kim, H. Kim, et al., An effective intrusion detection classifier using long short-term memory with gradient descent optimization, in: 2017 International Conference on Platform Technology and Service (PlatCon), IEEE, 2017, pp. 1–6.

[46] S. Bhattacharya, R. Kaluri, S. Singh, M. Alazab, U. Tariq, et al., A novel pca-firefly based xgboost classification model for intrusion detection in networks using gpu, Electronics 9 (2) (2020) 219.

[47] K. Bajaj, A. Arora, Improving the intrusion detection using discriminative machine learning approach and improve the time complexity by data mining feature selection methods, International Journal of Computer Applications 76 (1) (2013) 5–11.

[48] Ddos evaluation dataset (cic-ddos2019), `https://www.unb.ca/cic/datasets/ddos-2019.html`, accessed: 2020-09-09.

[49] Attackers using new ms sql reflection techniques, feb. 2015., `https://blogs.akamai.com/2015/02/plxsert-warns-of-ms-sql-reflection-attacks.html`, accessed: 2020-01-08.

[50] Stupidly simple ddos protocol (ssdp) generates 100 gbps ddos, `https://blog.cloudflare.com/ssdp-100gbps/`, accessed: 2020-01-08.

[51] The most popular types of dns attacks., `https://securitytrails.com/blog/most-popular-types-dns-attacks`, accessed: 2020-01-08.

[52] J. M. Alonso, R. Bordon, M. Beltran, A. Guzmán, Ldap injection techniques, in: 2008 11th IEEE Singapore International Conference on Communication Systems, IEEE, 2008, pp. 980–986.

[53] Ms03-034 flaw in netbios could lead to information disclosure, `https://support.microsoft.com/en-us/help/824105/ms03-034-flaw-in-netbios-could-lead-to-information-disclosure`, accessed: 2020-01-08.

[54] Ntp amplification ddos attack, https://www.cloudflare.com/learning/ddos/ntp-amplification-ddos-attack/, accessed: 2020-01-08.

[55] Tftp server buffer overflow, https://fortiguard.com/encyclopedia/ips/10268, accessed: 2020-01-08.

[56] F. Lau, S. H. Rubin, M. H. Smith, L. Trajkovic, Distributed denial of service attacks, in: Smc 2000 conference proceedings. 2000 ieee international conference on systems, man and cybernetics.'cybernetics evolving to systems, humans, organizations, and their complex interactions'(cat. no. 0, Vol. 3, IEEE, 2000, pp. 2275–2280.

[57] I. Sharafaldin, A. H. Lashkari, S. Hakak, A. A. Ghorbani, Developing realistic distributed denial of service (ddos) attack dataset and taxonomy, in: 2019 International Carnahan Conference on Security Technology (ICCST), IEEE, 2019, pp. 1–8.

[58] F. N. Jones, M. E. Nichols, S. P. Pappas, Organic coatings: science and technology, John Wiley & Sons, 2017.

[59] S. Patro, K. K. Sahu, Normalization: A preprocessing stage, arXiv preprint arXiv:1503.06462 (2015).

[60] L. Lusa, et al., Smote for high-dimensional class-imbalanced data, BMC bioinformatics 14 (1) (2013) 106.

[61] Understanding rnn and lstm, https://towardsdatascience.com/understanding-rnn-and-lstm-f7cdf6dfc14e, accessed: 2020-06-12.

[62] A. McCallum, K. Nigam, et al., A comparison of event models for naive bayes text classification, in: AAAI-98 workshop on learning for text categorization, Vol. 752, Citeseer, 1998, pp. 41–48.

[63] Nb explained, https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/, accessed: 2020-03-12.

[64] J. Platt, Sequential minimal optimization: A fast algorithm for training support vector machines (1998).

[65] Y. Jia, F. Zhong, A. Alrawais, B. Gong, X. Cheng, Flowguard: An intelligent edge defense mechanism against iot ddos attacks, IEEE Internet of Things Journal (2020).

[66] Waikato environment for knowledge analysis (weka), university of waikato, new zealand, https://www.cs.waikato.ac.nz/ml/weka, accessed: 2020-06-12.