Network forensics DDoS attack based on deep neural Network

SHEIKH SAZIB

School of Computer and Communications Engineering, Changsha University of Science and Technology
Changsha, Hunan, China.410114.
E-mail ：**mdsazibsheikh380@gmail.com**

DOI: 10.29322/IJSRP.12.10.2022.p13004
http://dx.doi.org/10.29322/IJSRP.12.10.2022.p13004

Paper Received Date: 20th August 2022
Paper Acceptance Date: 24th September 2022
Paper Publication Date: 6th October 2022

Abstract: The safety as well as reliability of computers networks and systems, essential modern infrastructures, are now urgently threatened by DDoS attacks. DDoS attack detection must be accomplished even before prevention strategies can be implemented. However, because to a flaw in ML/DL-based systems known as the Open Set Recognition (OSR) problem, full-scale success is still out of reach. An ML/DL-based system handling new instances that aren't taken from either the distribution system of a training data in this situation. It occurred when comes to DDoS attack detection because these attacks' technology is always changing, traffic patterns are shifting. The study looks into how the OSR issue affects the ability to identify DDoS attacks. We present a new DDoS detection system, a Gaussian Mixture Model (GMM), and bi-directional long short-term memory (BI-LSTM) in response to this issue. Traffic engineers discriminate and identify an unknown information collected by the GMM, which is then supplied back to the infrastructure as more training data. The findings of the experiment demonstrate that perhaps the suggested BI-LSTM-GMM can obtain recall, precision, and accuracy up to 94% through using dataset sets CIC-IDS2017 and CIC-DDoS2019 in training, testing, and assessment. The proposed architecture appears to offer a promising means of detecting unidentified DDoS attacks, according to experiments.

Keywords: DDoS Attack, Machine Learning, Regression method, SDN

CHAPTER 1

1.1Introduction

Rapid technological development over the decades has been focused on efficiently enhancing our way of life in an effort to address particular, targeted human requirements. With the help of technology, we may easily expand our way of life to greater plains and higher levels of complexity. The use of initiatives that seek to successfully share data from one person to another has substantially advanced thanks to the Internet's widespread use and integration. This utilization, simplicity, accessibility of its type, minimal rates of transaction, and confidence in communications platform have all been credited for this implementation and acceptance; all of which keep increasing its reputation, adoption convenience, and utilization. This popularity has attracted spams as well, a well-organized enterprise that uses messages without users' permission to generate revenue. Their services include the dissemination of malware known as spam, phishing,

and unsolicited advertisements. Unwanted or unsolicited texts are known as spam. Security experts have expressed tremendous alarm about the increase in spam(Bhaya & Manaa, 2014).

The amount of time people spend online is skyrocketing these days. In January 2021, there were 4.66 billion Internet users worldwide. Concessions like these, intended to circumvent security, obfuscate data privacy, and erode network infrastructure, have grown to be a major worry with detrimental effects on the uptake of technology. That provides customer denial, access, cyber attacks on data, theft of private data, and outages. Studies continue to demonstrate network intrusions that successfully assault any site at any moment. Denial of service attacks are just one example of how the exponential rate of attacks is as varied as the breadth of useful technology itself. This necessitates the urgent necessity to put an end to the attacks as quickly and closely as is practical. Most of these network resource attacks are planned, directed at a client system, then executed against with the servers through a variety of other compromised machines(Hoque et al., 2015a). It shows that there are many computers and systems that are interconnected with the outside world, raising important security issues. Security elements like Intrusion Detection Systems (IDS) must be added because no system is completely secure. An attack detection system that operates without human assistance is known as an IDS. The main issue that has caused problems even before the first IDS was released is misclassification brought on by the poor attack detection accuracy and inability to recognize contemporary attacks. Researchers have concentrated on this problem since it makes security analysts' jobs more difficult. Analysts may accidentally disregard serious cyber attacks as a result of this kind of responsibility. IDSs must be aware of every attack type and deal with it separately. The Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks are among the most significant of these assaults. Due to sending a lot of requests to a target at once, DoS attacks take place when a lot of resources (such memory and CPU time) have been used. As a result, it prevents legitimate users from using the service. DDoS assaults, as opposed to DoS attacks, happen when multiple computers or devices simultaneously produce a large quantity of service demand. The attacker is in control of these machines, often known as botnets. In DDoS attacks, the attacker's main goals are to consume network bandwidth and buffer memory(Hoque et al., 2015b).

Researchers have worked on enhancing DDoS detection systems or other IDS by implementing deep neural networks in order to address the aforementioned problems. Deep neural networks are used in intrusion detection systems to acquire information from enormous data sets in order to identify unusual network activity. The data sets' collected data can be utilized to improve detection systems. The algorithms can be trained to achieve this, giving security experts the level of satisfaction they want on the misclassification. In order to overcome the mentioned issues, researches have been done on improving DDoS detection systems or other IDS by incorporating deep neural network approaches. In order to discover unusual network behavior, intrusion detection systems employ deep neural networks to gather data from massive data sets. The obtained data from the data sets can be used to enhance detection methods(Baig et al., 2013). To accomplish this, the algorithms can be taught, providing security specialists with the level of satisfaction they desire regarding the misclassification.

A specific type of denial-of-service attack is distributed denial of service, or DDos. The security and integrity of PC companies and data systems, which are important pillars of modern society, are now under severe threat from these attacks. Before any measures of moderation can be used, it is challenging to identify DDoS assaults. Furthermore, ML/DL (Machine Learning/Deep Learning) has indeed been successfully used to predict the location of DDoS attacks. Here, an ML/DL-based framework ignores fresh cases not taken from the preparatory information distribution model. This problem is particularly important for spotting DDoS assaults since these attacks are becoming more innovative and have shifting traffic characteristics. DDoS attacks are growing and rising in size, frequency, and sophistication together with the increase and advancement of hazardous Web advancements. Organizational risks that might have

a significant impact on a company's operations include professional private time, information breaches, and sometimes even financial demands from programmers(He et al., 2017).

1.2 Fewer False Alarms and Efficient Identification

In order to uncover unusual invasion occurrences, DDoS detection algorithms employ stochastic analysis and make use of the entropy of network data. Traffic rate-limiting as well as screening mechanisms are implemented whenever an intrusion signal is triggered to lessen its effects. Any mitigation measures, nevertheless, that are implemented carelessly will harm legal traffic. Although the victim doesn't really experience a lot of traffic, a poor response could refuse access to authenticated traffic. Therefore, the detection technique ought to be able to tell the difference between attack traffic and legal traffic in order to be allowed to identify that somehow a DDoS attack activity is occurring(Gupta, 2018b).

Machine learning techniques are being used more and more in DDoS detection to categorize and detect suspicious traffic. Despite having to explicitly specify legitimate and harmful operations, these approaches are capable of learning the fundamental data properties. Although machine learning-based solutions show potential, the majority of methods concentrate upon offsite traffic monitoring and have difficulty capturing the constantly changing traits of DDoS attacks. Lastly, the detection technique should reduce false alarms, which can also harm reliable sources inadvertently. Therefore, the defensive system does not only stop attack traffic but also ensures that legal traffic is delivered to end users in a dependable manner(Ranjan & Sahoo, 2014).

1.3 Instantaneous mitigation and efficient inline inspection

Large businesses frequently use the scrubbing center technique to lessen the effects of DDoS attacks. All traffic is directed to a specified centralized information purification facility, known as a scrubbing center, wherein additional traffic analysis and protection are implemented whenever a questionable DDoS attack is found. Hazardous traffic is prevented while valid traffic is allowed to the system for transmission. Every traffic is rerouted using routing updates once the attack is over. This approach works well to shield victims from volumetric traffic, but it has three significant drawbacks. First of all, latency is typically increased. All traffic directed at the victim is diverted to the cleaning facility. In contrast to the detour-related latency, the scrubbing unit may also be a bottleneck and contribute significantly to latency, particularly if its bandwidth is constrained. Additionally, a complicated system is required for the detour strategy to correctly redirect traffic, which might be quite expensive in terms of both money and computation. Another benefit of utilising a scrubber centre is that monitoring is limited to inbound traffic (Peng et al., 2016). For companies and service providers that wish to prevent themselves from being used as an inadvertent DDoS attack platform, thus a strategy is insufficient. A perfect defense system would respond instantly and without any delays. We require an integrated assessment and immediate response system to do this. With inline analysis, all incoming and outgoing traffic across the protected network is examined. Without any need for route optimization, any detected attack traffic would then be simply blocked. Although inline inspection is the best option, it presents a significant challenge: how to do deep packet inspection at cable velocities while adhering to the strict memory and complication requirements? The majority of inline assessments now use packets or traffic monitoring algorithms to deal with this problem. Recent studies have nevertheless demonstrated that sampling techniques alter traffic patterns and reduce the detection performance. This substantial problem is still open(Osanaiye et al., 2016).

1.4 Defense that is Dynamic, Multilingual, and Cooperative

A common DDoS attack originates from numerous infected devices that unintentionally host attack applications. These infected devices cooperate with one another to direct attacking traffic toward the victim network's objective. The source of the attacking packets is therefore widely spread. As IoT and smartphone technology have grown, attackers are increasingly using a large number of potentially weak and unsecured devices as weapons. The origins of the assaults are not only widely spread but also might be located everywhere in the world. The frequency of DDoS assaults also shows that the malicious traffic also targets the companies that the victims depend on, such cloud service providers and Internet service providers (ISPs). Recent assaults appeared to be well-planned, very effective, and widespread. We argue that to effectively fight against DDoS assaults before they have a chance to substantially hurt both the model was used to identify and the Web infrastructure, a broad and collaborative defensive strategy is necessary. Based on an examination of attack patterns, experts separated the deployment of detection methods into three key areas: the origin system, the core-end network, as well as the victim-end connection. Each prospective deployment location has benefits and drawbacks of its own. The source-end network is the greatest place to limit attack traffic because there, it may be halted before it enters the World wide web core and consumes shared resources. It is difficult to tell the difference between genuine traffic and attack traffic at this location.

As already said, DDoS attack data is very dispersed, and the detection technique sees insufficient data at the source end to make a reliable identification. The victim-end network serves as a current perspective for DDoS suspicious network identification in comparison to the source-end network. Irrespective of the assault's nature or the perpetrators' location, since the detection method monitors the aggregated attack traffic close to the target, malicious activity and poor application servers are comparatively easy to see(Mahjabin et al., 2017). Sadly, it frequently happens that it would be too late to lessen the impact at this place. Deployed defense systems in the core-end network can also view the aggregated traffic, which puts them in a stronger way to implement impede the assault traffic.

It is obvious that a networked and cooperative infrastructure was necessary for an efficient defense system. Close to the victim end might be used to detect DDoS attack traffic, and upstream could be informed of the discovered attack traffic signature. As a result, it is possible to restrict the harmful traffic as far from the victim is practicable. Furthermore, by exchanging specific traffic data among several deployment locations, a distributed and cooperative mechanism can aid in early detection. The detection accuracy is improved by having a wider traffic monitoring area. A decentralized defence system also offers scalability, allowing for quick and affordable defensive updates in response to DDoS threat evolution in the future(Shieh et al., 2021a).

1.5 Forms of DDoS

There seem to be two main forms of DDoS attacks: I An attacker attempts to swamp an infrastructure with customer request that emission or saturate Computational time, energy, bandwidth, etc., finding it challenging for some other customers to reach the above assets (i.e. flooding); and (ii) An attacker attempts to generate a full volume of network attacks to a domain controller (i.e. protocol attack). If the attacker spoofs the source ip address to disguise packets and making it impossible to distinguish between legitimate and malicious data, a DDoS attack can avoid detection(Tavallaee et al., 2009). According on where they are deployed, detection systems are typically categorized as follows:

- Whenever a customer device (also known like an access point) is equipped with security features that allow it to identify and filter potentially harmful data in outgoing packets This finding, according to sources, was found just at site of the remote users that started the DDoS assault. The detector minimises harm to the system and other legitimate, unharmed packets of data while attempting to halt DDoS as fast and closely as feasible to the source of attack (best practise).

- Whenever a penetrated system notices an inbound malicious packet, it may easily discriminate between legitimate "incorruptible" packets and "vulnerable" attack packets using either an anomalous detection scheme or the wrongful use of intrusion. We refer to this as a victim-end detection. As well as an attack packet that hits a victim may experience bandwidth saturation and service denial.
- Whenever a network device has the freedom to freely attempt to identify a hazardous signal through percentage upon information in an effort to preserve overall performance of detection and the bandwidth consumption of an activity. When assaulting and legal packets are received at the routers, its network data is mixed by applying a proportion including all traffic information, which makes it easy to identify the attack's origin. Commonly, this is known as core-end identification.

Deep learning, which resembles the human brain, has become a hot topic of research in recent decades. Voice recognition software, image analysis, and translation software seem to be just a several of the areas wherein DL had proven useful in addition to the IDS industry. Despite the fact that the current DL model is faster and more accurate than prior models, a DL algorithm by itself cannot effectively invert multidimensional attribute links. In order to create the innovative technique known as the CLSTMNet, the present study combines two of the most popular deep learning algorithms, the Deep Neural Network (DNN) and the Long Short-Term Memory neural network (LSTM). DNN was used to automatically choose characteristics. LSTM was used to make predictions(Gupta, 2018a).

1.6 Motivation

The security of today's networks is not flawless, and as new technologies are developed and network infrastructure is continuously altered, new security issues arise. Multiple levels of security must be developed securely in order to address these issues, i.e., a suitable defense-in-depth architecture must be put in place. Network Intrusion Detection System is one of these security levels. An IDS aids in alerting whether a sophisticated attack is currently underway. In contrast, if an attack was made previously and by whom, suggesting that it somehow aids in locating the enemy and its movements.

In order for intrusion detection systems to stay current and be able to identify new attacks, they must be constantly improved. There are indeed difficulties in developing highly effective intrusion detection systems, despite the fact that many investigations have attempted to do so. In order to determine the effectiveness of attack detection and the optimum strategy for this sort of attack, study on IDSs must focus specifically on one form of assault(Alghazzawi et al., 2021). Furthermore, the primary aim of this project is to use deep learning algorithms to detect DDoS attacks.

1.7 Objective

- This study will propose a DDoS discovery model and defence framework in light of the environment for deep neural networks.
- This study offers a consolidated learning method for developing a DDoS model that takes use of the neural network learning's capacity for growth and adaptation.
- For DDoS recognition, a joint technique combining neural networking and information presentation is presented.
- To explore the ways by which the DDoS could be prevented
- To utilize the neural networking calculation to foster a model for DDoS, an information diagram framework which makes the model expandable and adaptable

1.8 Problems

(Dispersed) Denial of Service (DDoS) intrusions which can be induced by enhanced network connections, manipulation of network protocols weak points, or interfering with the Operating System (OS)/Firmware, resulting in a non-functional unified state as bricks, are a threat to the continuous production of a deep neural network. A further common IoT strike that really can impact Neural Network is electricity exhaustion, which is a limited resource that can be harmed by unreasonably increasing the burden of the network as well as its detectors, which can be accomplished whether through dedicated cyber-attacks or as a metabolic end (consequence) of DoS and DDoS attacks. Moreover, because deep Neural networks communicate data among base stations, surveillance and manipulation of data attacks such as Man-In-The-Middle (MITM) can compromise their security and integrity.

1.9 Significance of research

This study emphasizes on the investigation of detection of DDoS attacks with implementation of deep neural networks and had proposed solutions for these attacks. We had presented a unique network forensic framework based on deep neural networks for the processing and tracking of attacks. To establish the credibility of building network forensics, we had used a variety of machine/deep learning approaches with network data sources as network forensics models. In addition to this, we had investigated attack events and their traces in attack based on deep neural Network by analysing large data sets and constructing effective network forensics models. Lastly, we had evaluated and compare network forensics for detecting attacks in IoT networks using deep/machine learning.

1.10 Research Questions

- How to prevent DDoS attacks?
- How to recover DDoS attacks?
- What is the best detection approach for DDoS?

CHAPTER 2

2.1 Associated Works

Different ML approaches are often used to detect DDoS attacks, primarily as classifiers. The Naive Bayes Classifier, Random Forest (RF), Density-Based Spatial Aggregation of Application with Noise (DBSCAN), Artificial Neural Networks (ANNs), and Support Vector Machines (SVM) are a few examples of these. With SVM, a hyperplane is built in the transform coefficients to categorize unseen collected data on labeled training data. Cheng has developed a IAI (Ip Interactions Features) model that distinguishes between regular and abnormal traffic flows and aids in quickly and accurately identifying attack patterns. K nearest neighbors of the incoming data are found in KNN. The categorization of the entering data is determined by the bulk of these k neighbors. In order to categorize network state according to each step of a DDoS assault, Vu employed KNN and got excellent results. A Nave Bayes classifier is a classification method that relies on the Bayes theorem and the assumption of predictor independence. A Nave Bayes classifier makes the assumption that the existence of one feature in a class has nothing to do with the availability of any other features. Based on average as well as standard deviation of ethernet frames, Fadil used the NB approach to forecast the occurrence of DDoS attacks and obtained accurate findings. A decision tree collection is an RF. The classification is determined by the majority of independent decision tree results(Cheng et al., 2009). Wang et al. demonstrated that an RF technique may achieve an acceptable classification efficiency and an ideal feature subset with well-computed essential features in DDoS data.

It works well with data that has clusters with a comparable density. Dincalp dealt with the diversity of attack vectors by using a DBSCAN clustering technique. In their experiments, the suggested system performed admirably with the selected qualities. ANNs are simulations of biological brain networks. Using the back-propagation process, ANNs train the transformation matrix using label data. Ahanger suggested a DDoS detection strategy based on ANNs that had a 99.8% detection accuracy for identifying DDoS attacks. For DL, DDoS detection success stories exist as well. To identify DDoS attacks, Li and Lu utilized Bayesian techniques and Long Short-Term Memory (LSTM). The LSTM is appropriate for time-domain events with large gaps and delays. In the other terms, LSTM can decide whether information should be erased or maintained for an extended amount of time. In order to increase the accuracy of the detection process, the author employed the Bayesian technique in combination with LSTM to determine the confidence index of DDoS attacks(Vu et al., 2008). The auto encoder was used by Yang et al. to detect DDoS attacks. A number of neural network with just an unsupervised training process is called an autoencoder. During the training phase, it keeps the most important information while removing less important information and noise.

In their detection method called LUCID, which includes a dataset-independent preprocessing process and an activation analysis, Doriguzzi Corin et al. used CNN. The processing time of the suggested LUCID has been reduced by 40 times, making it suitable for situations with limited resources. In order to create safe solutions for IoT networks, Yong et al. employed deep learning models to identify webshell. These machine learning models' results are enhanced by ensemble methods such random forest (RF), random decision trees, and voting. According to their research, the Voting approach works well for heavyweight IoT scenarios whereas RF and ET are suited for lightweight IoT scenarios. An effective malware detection method deep learning based is proposed by Hemalatha et al. By addressing concerns with unbalanced data, the method uses an iteratively class-balanced prediction error in the Dense Net model's final classification layer to significantly enhance classification performance for malware. Extensive testing on establish a strong malware datasets revealed that the suggested technique performs better than the competition with a greater detection rate and reduced computing cost(Fadlil et al., 2017).

Despite the success of ML/DL-based systems, a crucial problem the OSR problem is ignored. Without appropriate safeguards, ML/DL could forcibly categorize cases from a new sample space into the incorrect category, as noted by Bendale et al. and Sabeel et al. The right course of action is to distinguish fresh cases from training samples, and this is the tactic used by. They calculated the hyper distance between entering data and known classes, and if the distance surpassed a certain threshold, they tagged the data as a new instance. Open Max, a brand-new model layer, is presented; it calculates the likelihood that an input belongs to an unidentified class. Furthermore, the Extreme Value Machine (EVM) is put forth, which builds models from training data using the Weibull function. For classification purposes, EVM will calculate overall insertion probability for every existing class given a fresh sample. A new instance that is distinct from all other classes is thought to belong to that class(Wang et al., 2017). Due to this trait, EVM is a strong contender for OSR issues, like the identification of recognized DDoS attacks.

2.2 Distributed Denial of Service attack

Critical oil and gas industrial installations are increasingly the target of network intrusion threats today. Industrial units have been made aware of a current industrial attack utilizing the malware Stuxnet on Iranian nuclear plants, which compromised active industrial process applications to cause damage or render them inoperable. Shamoon, Mirai, Wannacry, and other malware are only a few examples of those that have given threat profiling in several industries new depth. In the area, recent examples of oil industry behemoths like Saudi ARAMCO and others are common. Due to this, the production corporations have begun investing millions of dollars in preventive measures to safeguard their vital infrastructure. Denial of Service (DoS) attacks are a very popular form of network extortion computer

attack in today's world of the internet. In these attacks, an attacker disables a computer or backfires the network system and exhausts its legitimate users' systems, whether temporarily or permanently, preventing them from connecting to a host or the Internet(Ahanger, 2017). In a distributed denial of service (DDoS) attack, the incoming network traffic of the victim network is flooded by a variety of online victimized devices on the internet (numbers can reach millions), known as bots, and in such a situation, it will be impossible to prevent the attack by obstructing a few of the incoming data sources.

Botnets are groups of interconnected computers and other devices that communicate with one another over the Internet. Whenever made as just a targeted attack employing botnets, DDoS attacks are incredibly difficult to halt. Recurrent neural networks monitor network traffic and, using statistical analysis of genuine network activity, can identify attacks. Machine learning is a growing method for detecting attacks with improved performance and refining the detection based on past data(Li & Lu, 2019). These machine learning techniques do, though, have a limited model presentation restriction. Criminals frequently use DDoS attacks on high-profile servers that provide web services, including financial institutions, virus protection providers, website services businesses (OVH, that also maintains as well as broadcasts numerous essential internet sites), gaming internet sites, and online payment gateways, with the goals of retaliation, extortion, and social movements.

2.3 DDoS attack commonly used types

2.3.1 UDP flood

UDP flood is a type of distributed denial-of-service attack. Additionally, the connectionless User Datagram Protocol (UDP) protocol. In this form of attack, the host looks for apps that are connected to the data grams, and if it doesn't find any, it sends a "Destination Unreachable" packet return to sender. Many of the workstations send large amounts of UDP packets to the target system. As a result of these flood attacks, the system will be overburdened and unable to handle legitimate traffic(Yong et al., 2022).

2.3.2 Flood in SYN

Due to a flaw in the three-way handshake portion of the TCP protocol, a SYN flood is a specific kind of DDoS assault. When a SYN demand to create a Network interface with a server is accompanied by a SYN-ACK reply from the host, which is then verified by an ACK reply from the requester, a 3 handshake will also be carried out. When the host's SYN-ACK is silent or the requester uses a forged IP address to send several SYN requests, the situation is known as a SYN flood. We didn't have 3 handshakes since number three was forgotten, as a consequence. Eventually, DoS will happen since the host will keep waiting for each request's confirmation, which will prevent them from creating a new connection.

2.3.3 Death Ping

A "POD" attack occurs when a machine receives repeated damaging or malicious pings. IP packets have a header and are 65,535 bytes long. The maximum frame size may be limited by the Data Link Layer (DL). For instance, the frame limit on an Ethernet network is 1500 bytes. In this case, the destination host would have to separate a large number of IP packets (also known as fragments) into smaller packets before reassembling the packets. Due to malicious change of fragment content, the reassembled packet ends up being larger than 65,535 bytes, which is what happens in the "ping of death" situation. Due to an excess of storage buffers generated for packets, this may result in denial of service for legitimate packets.

2.3.4 Smurf assault

A DDoS attack that happens at the network layer is known as the Smurf attack. Smurf attacks are one type of amplification-shape attack that broadcast networks use an ICMP request to address. ICMP is typically used to define the operational status of nodes and for data

exchange. The attacker spoofs the victim's IP address using an ICMP request. Because ICMP does not use a handshaking protocol, the end node cannot verify the legitimacy of the source node. Every connected device in the network will immediately receive the request once the router receives it. Attackers are assured of success once they hear these replies from victims. Due to a high volume of ICMP, the server will deny access to its services.

2.3.5 HTTP influx

The application layer is the target of this DDoS attack type. A web server or application can be attacked using an HTTP GET or HTTP POST request. Due to the fact that an HTTP assault can be launched without the use of reflection technology, distorted packets, or spoofing, it is very challenging to identify and stop. In comparison to other assaults, this one requires less bandwidth to bring down the targeted server(Bendale & Boult, 2016). One of the most sophisticated unstable security concerns is the HTTP flood attack, which exploits the common URL request and makes it exceedingly difficult to distinguish between legitimate traffic.

2.4 Methods for Detecting DDoS

To identify DDoS attack traffic, many different approaches have been developed. The early methods concentrate on stochastic analysis to track the behavior of network traffic flows and take advantage of network traffic entropy to recognize typical behavior and find anomalous intrusion events. Utilizing machine learning techniques to categorize and identify malicious traffic has become popular recently as a method of detecting DDoS attack activity. We examine both conventional and intelligence based detection approaches in this part.

2.4.1 Customary Detection Methods

Network traffic is characterized by conventional identification techniques employing statistical and information theoretic analysis. Typically, they take a preset model of the normal state as a given. The data of any monitored traffic are periodically inferred and compared to the predetermined model. Any traffic that doesn't conform is viewed as an attack. In, the authors suggested detection strategies that track the ratios of packets transmitted to and received from the guarded network. They contend that for a given protocol, the observed proportion for valid traffic should be lower than a particular threshold. Then, DDoS attack data is recognized appropriately. In, the article looks at the significant association that either an attack or legal traffic presents from several angles. They analyze network traffic metrics using correlation coefficients, compare the observed changes, and come to choices. In, the authors make the assumption that typical user behavior, such as the frequency of web page requests and the duration of browsing, generally follows certain distributions(Sabeel et al., 2019). The frequency distribution of the sampled flows is calculated periodically by the detection technique. The flow will be recognized as attack traffic if the estimated value is sufficiently high.

In DDoS detection, information-based metrics are very widely used. With a specified time window, entropy can be calculated for a variety of variables, including real networks, supplier IP addresses, packet counts, supplier IP ports, etc. Various information parameters are utilized to assess the variations in the generated entropy values, which are then used to detect DDoS attack traffic. The difficulty these methods encounter comes from the requirement to define exactly a standard profile and distinguish between typical and abnormal behaviour. The statistical behavior of authorized users can be easily imitated with a large enough number of active hacked workstations. As a result, the main premise of the proposed works will be broken, and the defense strategies will be misled(Yulita et al., 2017). Furthermore, the deployment location of these monitoring systems might have a significant impact on their performance.

2.4.2 Methods for Intelligent-Based Identification

- Utilizing machine learning approaches has yielded promising results in recent DDoS detection studies. These methods circumvent the drawbacks of conventional detection methods by being able to confidently learn the properties of the underlying data with no need to explicitly identify good and bad behavior. DDoS detection issues are conceptually characterized as binary

classification issues, where monitored traffic is either categorized as genuine or attack traffic. On well-known benchmarks, various categorization methods have been tested and put to use. Researchers have also looked into various feature sets to increase detection accuracy while reducing false alarmsDeep learning, a sizable percentage of machine learning techniques, has been used to solve challenging problems in various of industry sectors, such as machine learning, networking site sorting, electronic games, applied linguistics, language processing, health insurance, computational biology, computer vision, and drug discovery. However, it hasn't been widely employed for DDoS detection (Gupta, 2018a). In this part, we highlight many recent important initiatives on this subject. A number of studies use autoencoder, an unsupervised learning approach, to derive non-linear models from the input data, and then they use a classification methodology to distinguish between malicious and legitimate traffic.

For network intrusion detection, the authors of integrate a machine with just a soft-max regression layer. Both in binary and multi-class text categorization, the suggested technique performs admirably. In order to detect DDoS attacks, the authors then expand on their approach by stacking two auto encoders. In order to understand the intricate correlations between features, the developers stack two auto encoders. They integrate the tightly packed machine with a Classification Algorithm for intrusion detection, arguing that the soft-max surface is slower than conventional classifiers. In addition to using auto encoder for feature learning, the authors also use it to cut down on the amount of random variables taken into account. An number of layers, that reflects the condensed characteristics, is being used as the classifier's input rather than the auto encoder's output layer. The classifier is an SVM (Support Vector Machine). According to the authors, the SVM shows the best performance of reliability than other traditional classifiers. The suggested techniques successfully deal with the issue of feature selection, however they do not deal with the difficulties of feature extraction. RNN is a well-liked alternative to auto encoders for intrusion detection. An extension of a traditional neural network known as an RNN is frequently used to model sequential data and address time-series issues(Rudd et al., 2017).

The majority of methods use KDD99 and NSL-KDD, that employ manually created flow-level feature engineering, to implement RNN algorithms to well-formatted datasets. Each record in these datasets represents a particular network flow and is identified by a number of variables, including time, amount of frames, number of data packets, etc. The majority of proposed approaches consider each set of attributes as sequential data that is utilized the inputs for RNN models. These studies employ several RNN model types to enhance classification performance and other assessment criteria. It adopts a straightforward RNN. The developers use LSTM RNN and concentrate on choosing the smallest possible collection of characteristics. Gated Recurrent Unit RNN was utilized by the authors. Although these proposed systems perform better than conventional machine learning techniques, they fall short in solving the feature extraction problem. The assumption of order of events between variables in RNN networks is also debatable. RNNs have a memory cell that can retain data received previously as well as retain the temporal relationship among data, which makes them advantageous in modeling sequential data.

CHAPTER 3

Machine Learning

3.1 Utilizing Machine Learning to Identify DDoS

3.1.1 Machine-learning strategies

Modern machine learning techniques are employed, particularly just at phase of intrusion detection system, to identify and defend against DDoS. Svm Algorithm, Neural Network, K-Nearest Neighbor, Decision Tree, and Naive Bayes are some of the methods that are now in practice. Utilizing predetermined mathematical factors or predetermined guidelines to analyze and train in accordance with the recorded rules from the predefined database, network traffic is initially retrieved and processed. For subsequent analysis, characteristics are selected and eliminated from the traffic (for example, the method, byte frequency, and packets percentage in the stream of data). In the following phase, the features are normalized to prepare them for training. The learning methods will be used to train the neural network, and packets from network traffic will be separated out depending on whether they are from an attacker or a genuine source. The algorithm will modify the database because of its screening criteria and remove the recognized DDoS attacker packets (illegal traffic) from the network traffic(He et al., 2017).

3.2 Machine learning Types

Supervised learning, unsupervised learning, and semi-supervised learning are the three primary categories of machine learning. Subtypes can be created within each of these three categories.

3.2.1 Supervised learning

A machine can be trained using supervised learning utilizing labelled datasets. As the name suggests, it denotes that many of the labeled data has been associated with the right response. With supervised learning, one can forecast unexpected outcomes by retraining the labeled data. For instance, if one wants to teach a machine to estimate the travel time between two points, relevant information must be obtained and examined. In particular, such information may consist of the route taken, time of day, and weather conditions. These specifics are all regarded as inputs. Presumably, everyone believes that this will take longer to get to the destination if this is raining, however a machine uses statistics. Therefore, particular data like the entire time required from a start place as well as corresponding data which requires time, weather, route, and so on are needed to construct a dataset which can be trained.(Blomstrom & Kokko, 1998) The program would be able to determine the relationship between various variables and forecast the amount of time it will take to travel between two locations using the provided information.

3.2.2Un supervised learning

The machine learning does not need to be supervised when using the unsupervised learning technique. Unsupervised learning utilizes unlabeled datasets because the model will attempt to find the information on its own. The computer can find any kind of data pattern in unsupervised learning, and it also aids in figuring out the features required to classify the data.

3.2.3 Semi-supervised learning

A different kind of learning is known as semi-supervised learning. Either supervised learning as well as unsupervised learning are used in this method. During the training phase, it blends a small amount of labeled data with a vast amount of unlabeled data.

3.2.4 Categorization

Characterization is used to predict the numbers that will be provided. The basic objective of this kind of machine learning techniques is to create a model that can accurately divide an input vector into classes that are available and accessible depending upon that labeled and training data. Each sample typically has just one input class. Assessment edges and decision borders are used to partition the input

space into decision regions. Data pre - processing, retraining, and categorization are the procedures that must be completed in that order during the classification process.

Training: Data are trained as the model enters this phase to achieve the maximum level of predictability possible.

The given input is assigned to being one of the learnt algorithms, that generates a judgment based on previously developed decision rules, to start the classification step.

There are three major components to the classification issue. First, the distribution of potential classes and class frequency of the input data. The relationship among input and output is established in order to define distinguishing traits. Thirdly, figuring out the loss function will help to lower the overall cost of penalizing incorrect predictions. Its probability theory-based categorization issues can all be found in real-world scenarios. By constructing a vector that represents the likelihood of each probable class, prediction models are being used to demonstrate and describe uncertainty. The ultimate learning model is used for forecasting, enabling the learned model to shed light on an unidentified input type (test data)(Gupta, 2018a). Despite the fact that this is true, every classification algorithm that is created and constructed aims to store training data in memory. Due to the training process, the classifier's efficiency may degrade when applied to new data, but it will continue to be accurate when applied to training data. This problem is referred to as over fitting.

3.3 Deep Learning

In deep learning systems that employ conventional machine learning techniques, features play a significant role. To choose such features, domain expertise and fine-tuning are necessary, and they contribute to the success of conventional machine learning methods. A collection of methods known as representation learning are useful for categorizing and making predictions about almost anything. The ability of the learning process known as "deep learning" to learn from several levels of representation by combining various non-linear transformations sets it apart from other learning methodologies. The rationale behind stacking numerous layers of transformations is the idea that the data is made up of a number of fundamental components that are connected to one another in a hierarchical structure(Alghazzawi et al., 2021).

When referring to the layers closest to the input, the bottom layer is often used. This is predicated on the idea that although low-level qualities, like gradients and edges, are shown by levels nearest to the input, or even at the 'least' level, more complex characteristics, like faces and objects, are indicated by levels nearest to the output, or even at the 'highest' level. Two of the most major differences among machine learning and data mining techniques are data size and approaches to problem-solving. In contrast to machine learning, deep learning usually makes use of big data. Deep learning will complete a problem from beginning to end, in contrast to machine learning, which employs a split-and-handle methodology. This is the second distinction among machine learning techniques. Additionally, deep learning enables sequential layer parallel processing, whereas machine learning only permits single layer parallel processing. One of the most popular deep learning techniques is called a deep neural network (DNN)(Siddiqi & Pak, 2020)

CHAPTER 4

4.1 Research design

In this study qualitative methodology had been adopted. To investigate the maximum capacity of the proposed system there had been some examination headings that merit further examination: approval on more datasets, auto-setup. This strategy had propelled the current practice and issues in DDoS identification.

4.2 An expandable and adaptable Deep Learning strategy

In this exploration, an expandable and adaptable neural learning network interruption recognition framework had been introduced .Writing on building a prescient model for circulated forswearing of administration assault (DDoS) is rich, yet entirely the high level DDoS doesn't cover the expandability and adaptable way of behaving of the interruption recognition model. Versatility is be-coming progressively expected for the present network interruption discovery.

4.3 Suggested Framework (GMM)

This study suggests a system that integrates a Gaussian Mixture Model (GMM), supervised learning, and bi-directional long short-term memory (BI-LSTM) to address the issue of DDoS assault detection. The BI-LSTM is used for the distinction between routine data as well as DDoS attacks using learning algorithm. The GMM is used to create a distribution model for the training data that is based on an idea similar to unsupervised learning. If fresh attacks or new valid traffic extend beyond a certain bounds of the built distribution model, they can be intercepted using GMM. The traffic engineer must identify and categorize newly captured traffic before using it to upgrade the BI-LSTM and the GMM in the gradual process of learning. The BI-LSTM can be thought of as a detector for something like the differentiation of positive and negative traffics, and the GMM can be thought of as a classifier for differentiating learnt samples and novel cases. Our system creates models that will be used to identify network using deep learning by using past traffic data for training. We also add the Gaussian mixture modules to the model to enable detection of previously undetected attacks. Finally, specialists will identify and categorize unidentified attacks or traffic categorised by the Gaussian mixture model, and the supervised neural model will be upgraded through incremental learning.

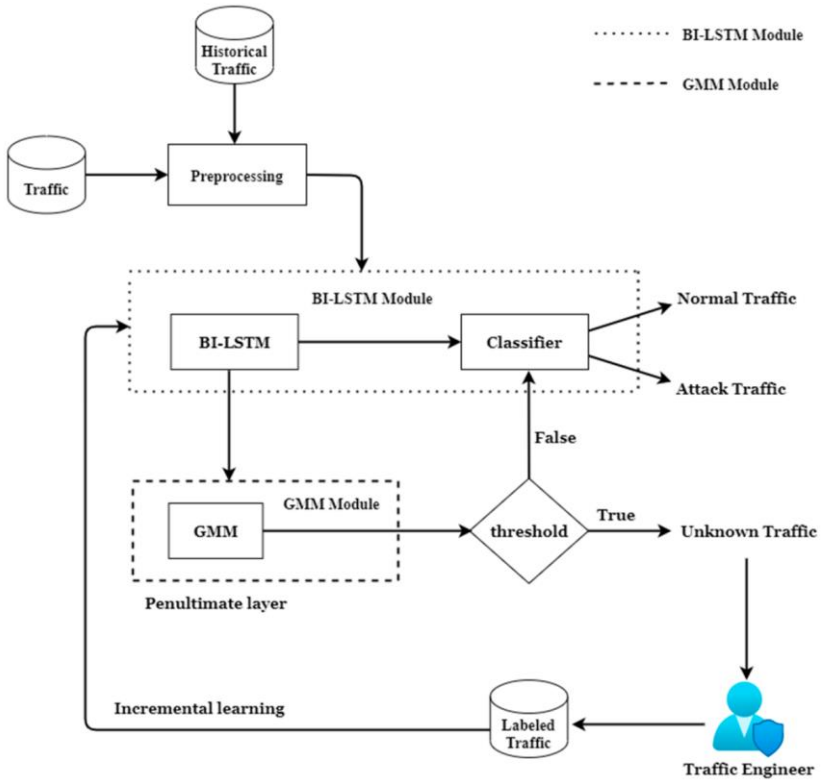


Figure 1 The BI-LSTM-GMM functional block diagram that has been suggested.

4.4 Module BI-LSTM

The LSTM is a Recurrent Neural Network (RNN) design featuring gates for explicit input, output, and forget control and a memory cell. Because of its distinctive form, LSTM is especially well suited for applications involving data sequences. An LSTM's operation is controlled by (a), which includes the following:

$$it=\sigma(Wt, [ht-1,xt]+bi)$$

$$ot=\sigma(Wo, [ht-1,xt]+bo)$$

$$ft=\sigma(Wf, [ht-1,xt]+bf)$$

$$Ct\tilde{}=\tanh(Wc, [ht-1,xt]+bc)$$

$$Ct=ft \cdot Ct-1+it \cdot Ct\tilde{}$$

$$ht=ot \cdot \tanh(Ct)$$

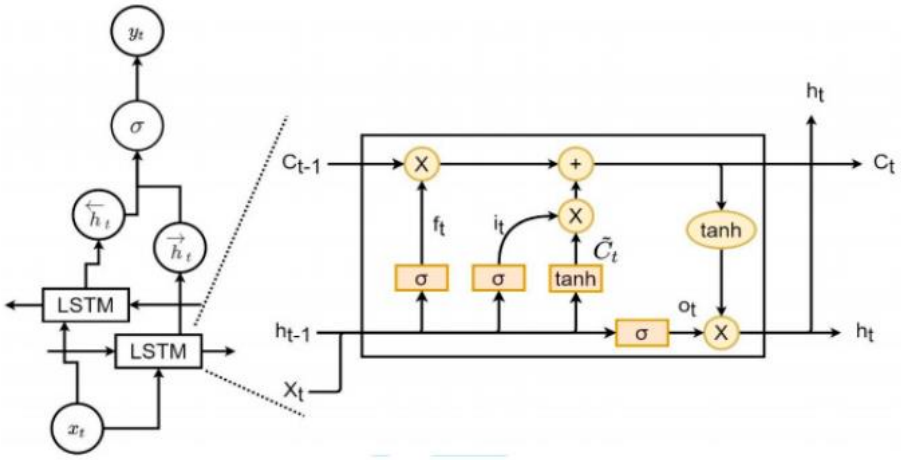


Figure2 The LSTM stage's inner architecture

where Ct and Ct are the current and last states, respectively; t is the index of the LSTM stage; xt is the inputs; ht is the outputs; W is the mass; b is the bias; and () is the sigmoid function. A DL design with two LSTMs—one for forward motion and another for backward motion—is known as a BI-LSTM. As with data sequences, a BI-LSTM is now a non-causal network. A desirable quality again for identification of DDoS assaults is its ability to grasp correlations along both sides of the time axis. The BI-output LSTM's is sent to a deep network that acts as a classifier to distinguish between legitimate and malicious data(Shieh et al., 2021c). Only when the GMM indicates that the given instance falls inside the learning distribution of data is the output of the classifier counted.

4.5 Module GMM

An ML/DL-based system must be able to recognize its limitations. We use the GMM in our design, an extension of the single Gaussian model, to give the DDoS assault detection system the capacity to detect instances not drawn from either the estimation method of the training dataset. Clustered data, a common case in DDoS attack detection, is modeled using GMM by using several Gaussian probability density functions. In the feature space, many valid traffic types and DDoS attacks cluster together according to similarities they share. GMM can create a model that matches the distributions of the training sample using unsupervised learning. A statistical model made up of K Gaussian probabilities is known as a GMM. Each kernel's weighting factor (wk), mean (k), and covariance matrix (k) can be described as follows:

$$\Lambda=\{\lambda k=(wk,\mu\rightarrow k,\Sigma k)\},\ k=1,\ldots,K$$

A particular instance x's likelihood of becoming a part of a GMM can be calculated using the following equation:

$$p(x\rightarrow||\Lambda)=\sum k=1KwkN(x\rightarrow||\mu\rightarrow k,\Sigma k)$$

The expectation-maximization approach is the method most frequently used to produce an estimate of the mixture model using a testing dataset and K. The GMM could be used to exclude unknown instances once the model has been created(Sarker et al., 2020). A failed occurrence would be logged, subject to data engineers' identification and labeling.

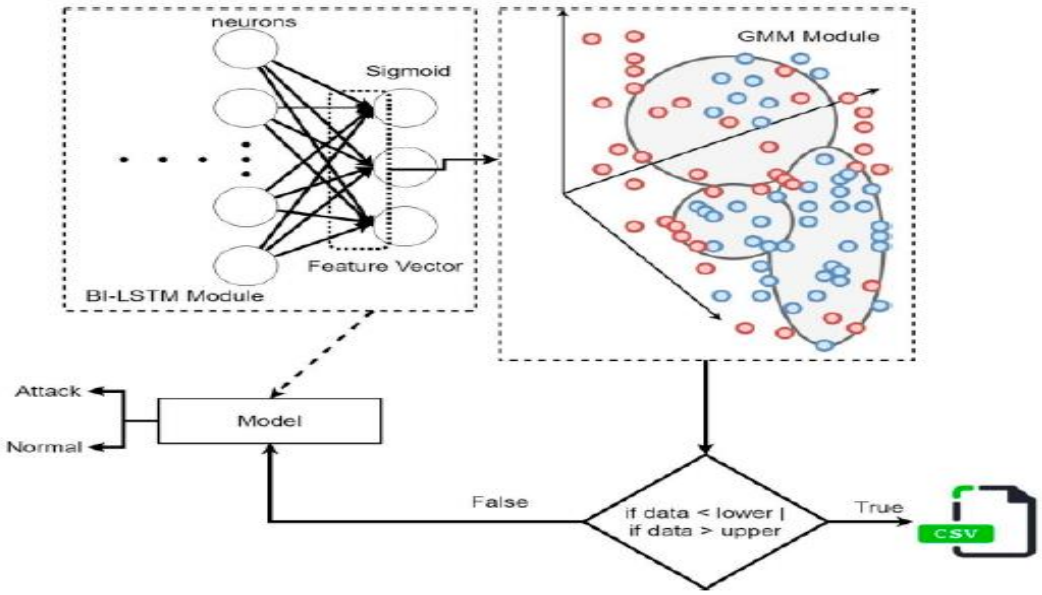


Figure 3 The GMM module's control flow.

4.6 Continuous Learning

If a thing can keep learning, it can keep getting better. Retraining the program from scratch is a possibility but a time-consuming operation given the current training set and newly acquired training data. Incremental learning, on the other hand, is a more logical and effective choice. The process of continuously improving an old model with fresh data is known as incremental learning. Given the constant evolution of both the assaulting technology and the accompanying traffic, this feature is crucial for DDoS attack detection. Traffic engineers will log and classify any traffic that the GMM module rejects within the suggested framework(Sarker et al., 2020). The tagged information would be used as training data for the BI-LSTM and the GMM during the incremental learning stage.

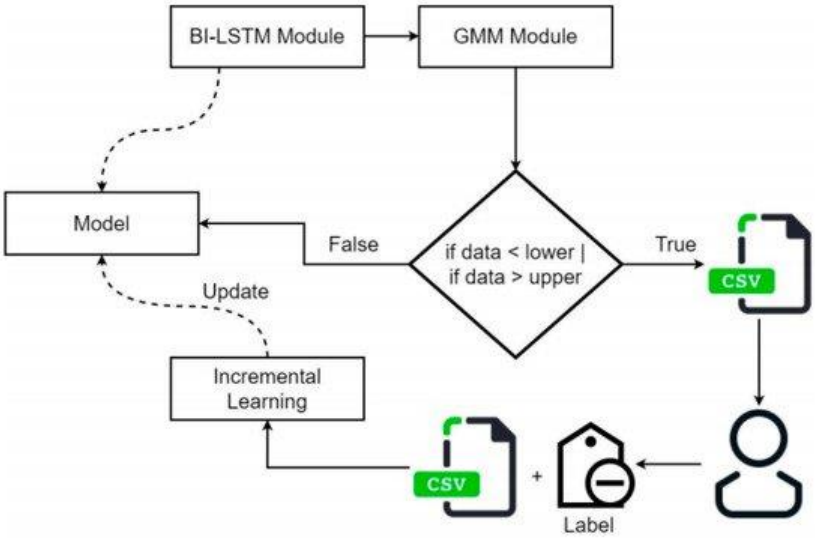


Figure 4 An incremental learning process's control flow

4.7 Framework for Detecting DDoS Attacks Employing Multiple Linear Regression

The main research goal of the suggested approach is to develop a machine learning model relying on multiple linear regression modeling and to visualize data by taking coefficient of determination and fit charts into account. The suggested method's goal is to investigate the viability of using multiple linear regression analysis on the CICIDS 2017 dataset, the benchmark dataset that has been extensively utilized in a number of the most notable development research publications. The goal is to first use the feature selection technique to identify the crucial characteristics that make superior outputs for the prediction model. For carrying out feature selection in the current approach, as shown in fig, we adopted the Information Gain approach method. A common paradigm used in many data mining-based applications is the information gain technique. The behavior of the chosen and retained significant attributes of the CICIDS 2017 dataset is analyzed by examining the fit charts and residual plots after the selected features are taken into consideration for carrying out multiple linear regression analysis. The experiment results are analyzed using the suggested methodology in the next part utilizing the Friday log files from the most popular CICIDS 2017 study dataset.

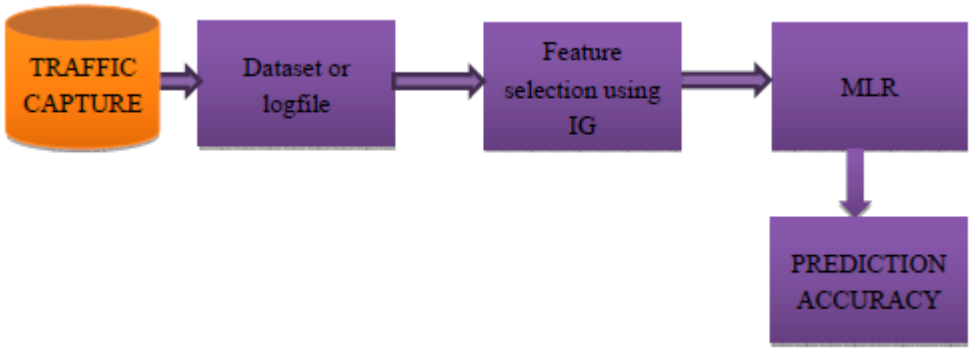


Figure 5 suggested machine learning method for detecting DDoS attacks

4.8 DDoS Attack in SDN

The DDoS assault is among the most harmful ones against SDN. The attacker can quickly generate a large amount of traffic using fake IPs, severely damaging the network and rendering the controller inaccessible to authorized users. Unfortunately, DDoS attacks can

occur at any SDN layer, and these attacks differ from those that are recorded in traditional networks and even from those that are classified as DDoS.

4.8.1 Solutions Based on Deep Learning

Deep Learning (DL) methods now play a major part in anomaly detection methods. Such methods can automatically and without human involvement extract the deep structure as from incoming data. The DL was only seldom used in works to launch DDoS attacks on SDN networks. In order to effectively defend from DDoS attacks in SDN networks, Li et al. constructed DL algorithms. The suggested model was tested on the ISCX dataset using three deep learning (DL) algorithms: CNN, LSTM, and RNN. In both the training and test sets of data, their model effectively attained accuracy levels of 99% and 98%, respectively. The majority of the current studies verified their models using a dataset created based on standard IP networks instead of SDN platforms, demonstrating how greatly the DL techniques may solve the fundamental issues of traditional ML techniques.

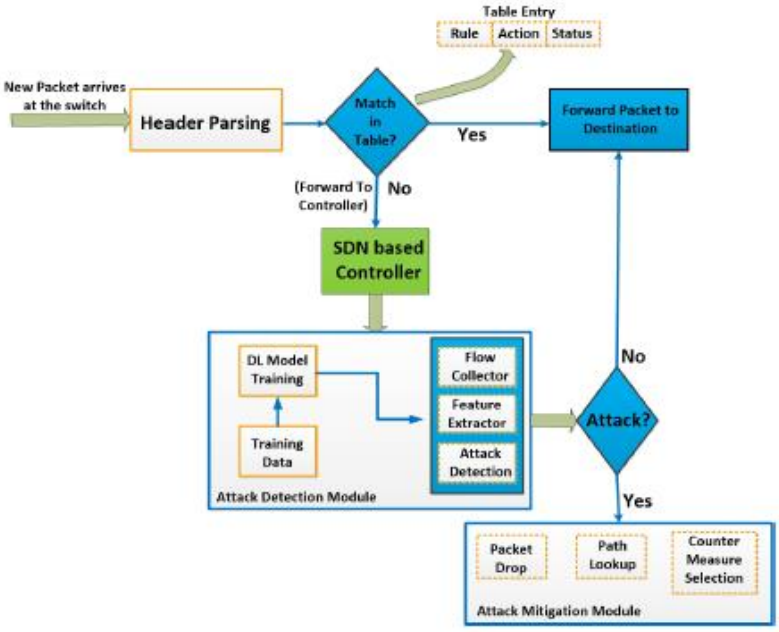


Figure 6 Framework for SDN anomaly detection and mitigation.

CHAPTER 5

Methodology and Results

5.1 Experimentation

To verify the viability and efficacy of the suggested framework, a number of experiments were carried out. The TensorFlow 2.0 and Keras frameworks were used to implement the tests, which were run on an Intel i5-9500 CPU with 32 GB of RAM.

5.2 Data Set

In our studies, two well-known datasets—CIC-IDS2017 [23] and CIC-DDoS2019—are used. The Canadian Centre for Information security used Wireshark in simulated situations to gather the datasets. They are created utilizing different DoS and DDoS attacks, two different consumption profiles, and multistage attacks like Heart bleed. The CIC Flow Meter is then used to pre-process the collected traffic. To provide the various DoS and DDoS traffic data, it contains 80 network traffic features. The final data collection includes CSV files of traffic aspect records. The non-numeric variables are changed using One-Hot encoding to fit within the framework's recommended numeric structure. Then, all fields are normalized in preparation for rescaling their dynamic ranges.

DATA SET	TRAFFIC TYPE	NO OF INSTANCES	RATIO	TOTAL NO OF INSTANCES
CIC-IDS2017/Wednesday	BENIGN	440,030	0.62	692,703
	DoS GoldenEye	10,294	0.013	
	DoS Hulk	231,074	0.334	
	DoS Slowhttptest	5498	0.007	
	DoS Slowloris	5795	0.006	
	Heartbleed	12	1.6×10−5	
CIC-IDS2017/Friday	BENIGN	97,717	0.431	225,745
	DDoS	128,028	0.568	
CIC-DDoS2019/NTP	BENIGN	14,366	0.0119	1,217,007
	DDoS/NTP	1,202,641	0.9880	
CIC-DDoS2019/LDAP	BENIGN	1611	0.0006	2,181,542
	DDoS/LDAP	2,179,931	0.9993	
CIC-DDoS2019/SSDP	BENIGN	764	0.0003	2,611,374
	DDoS/SSDP	2,610,610	0.9996	

Table 1 CIC-IDS2017 and CIC-DDoS2019

The confusion matrix and indeed the Reliability, Proficiency, and Recall are performance indices. Precision makes an effort to determine what percentage of affirmative identifications are in fact accurate. Recall is concerned with the accuracy of genuine positive identification. The percentage of instances that are accurately categorised when they are identified is known as precision.

$$\text{Accuracy} \triangleq \frac{TP+TN}{TP+FP+FN+TN}$$

$$\text{Precision} \triangleq \frac{TP}{TP+FP}$$

$$\text{Recall} \triangleq \frac{TP}{TP+FN}$$

$$\text{F1 Score} \triangleq \frac{2*TP}{2*TP+FP+FN}$$

Predicted	Actual	Attack	Normal
Attack		TP(True Positive)	FP(False Positive)
Normal		FN(False Negative)	TN(True Negative)

Table 2 Jumbled matrix.

5.3 Preprocessing

One of the most important steps in both machine learning and data mining methodologies is data preparation. Preprocessing can be helpful in assisting you in overcoming a variety of data issues, as well as difficulties once you have a large dataset and need to get a more precise output. The preprocessing method guarantees that raw data is updated, preprocessed, and transformed into acceptable, suitable, and pure data, boosting accuracy.

5.4 Module BI-LSTM

We developed a BI-LSTM design with setup and parameter settings after making some investigative efforts. L2 regularization imposes constraints on the kernels, biases, and activation. During the training and testing, we used a 10-fold cross-validation approach. The pullout process was used to get around the over-fitting issue.

Layer (type)	Output Shape	Param #
Hidden_Layer_1 (Dense)	(None, 28)	1596
Hidden_Layer_2 (Dense)	(None, 10)	290
Output_Layer (Dense)	(None, 1)	11

Total params: 1,897
Trainable params: 1,897
Non-trainable params: 0

Table3 Settings for the BI-LSTM

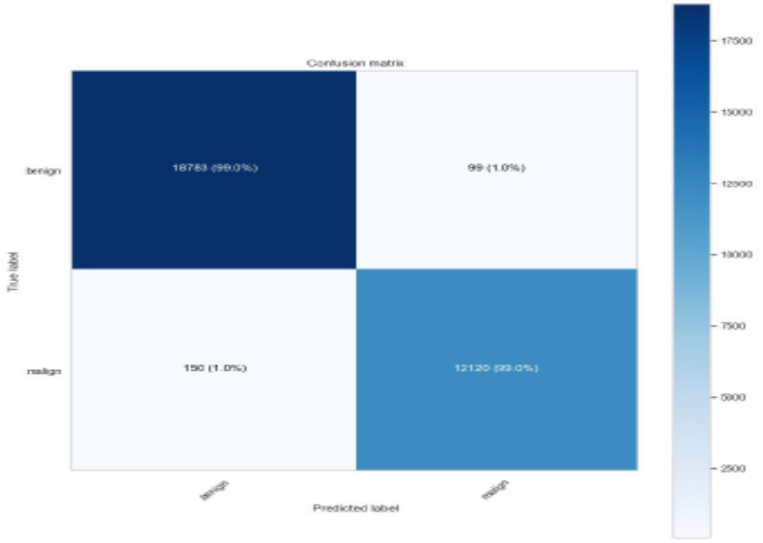


Figure7 confusion matrix

Parameter	Setting
Epoch/size of batch	500/1023
Clipnorm	0.8
Learning rate	0.00858
Momentum	0.88
Decay	0.001
Bidirectiional Layer	2

Table4 Setting up the BI-parameters. LSTM's

First, we looked at the BI-performance LSTM's as just a DDoS detector. The "CIC-IDS2017/Wednesday" data set was used to test the BI-LSTM. The BI-LSTM was tested using the same data set, "CIC-IDS2017/Wednesday," after training. The BI-LSTM was properly trained and behaved, as can be seen in initial row of Table, in accordance with its intended use. The training data set's malicious and legal traffic may be distinguished with great ease by the BI-LSTM.

Test Data set	Recall	Precision	Accuracy	AUC	F1
CIC-IDS2017/Wednesday	0.997	0.971	0.988	0.985	0.984
CIC-IDS2017/Friday	0.413	0.985	0.663	0.704	0.583

Table 5 Effectiveness of the BI-LSTM trained on "Wednesday, CIC-IDS2017"

Our focus is now on the consequences of the OSR issue. In other words, we looked at how a well-trained BI-LSTM responds to novel occurrences that it has never encountered before. The BI-LSTM trained with "CIC-IDS2017/Wednesday" was tested using "CIC-IDS2017/Friday". The performance, recall, and accuracy significantly decreased to about half, which is effectively random guess, as in the second row of Table. This issue arises because BI-LSTM lacks an internal method for determining something does not grasp. The BI-LSTM compulsorily chooses the currently serves on a mismatched model for novel instances not derived from the distribution model of the training set, which typically yields inaccurate classification. Notably, accuracy does not suffer noticeably. Due to this, the distribution of legitimate traffic in "CIC-IDS2017/Friday" and "CIC-IDS2017/Wednesday" is identical in feature space.

5.5 Module GMM

Our preferred approach to the OSR issue is GMM. With GMM, a provided training data set's distribution is modeled by weighting K Gaussian probability distribution functions. A crucial system parameter is K, the amount of Gaussian distributions. The computational expense for calculating the K value and model correctness must always be balanced. The AIC (Akaike Information Criterion), which measures information entropy, is frequently used to assess the accuracy of fitting. The accuracy improves as the AIC decreases.

$$AIC \triangleq 2K - 2\ln(L)$$

We used "CIC-IDS2017/Wednesday" to build the GMM model, and we looked at how the AIC changed for various K values. It is obvious that greater precision is caused by a higher K value. But at the other end, a higher computing cost is implied by a bigger K value. Consequently, in following studies, a reduced K value of 5 and its corresponding =12.23 and =2.20 were observed.

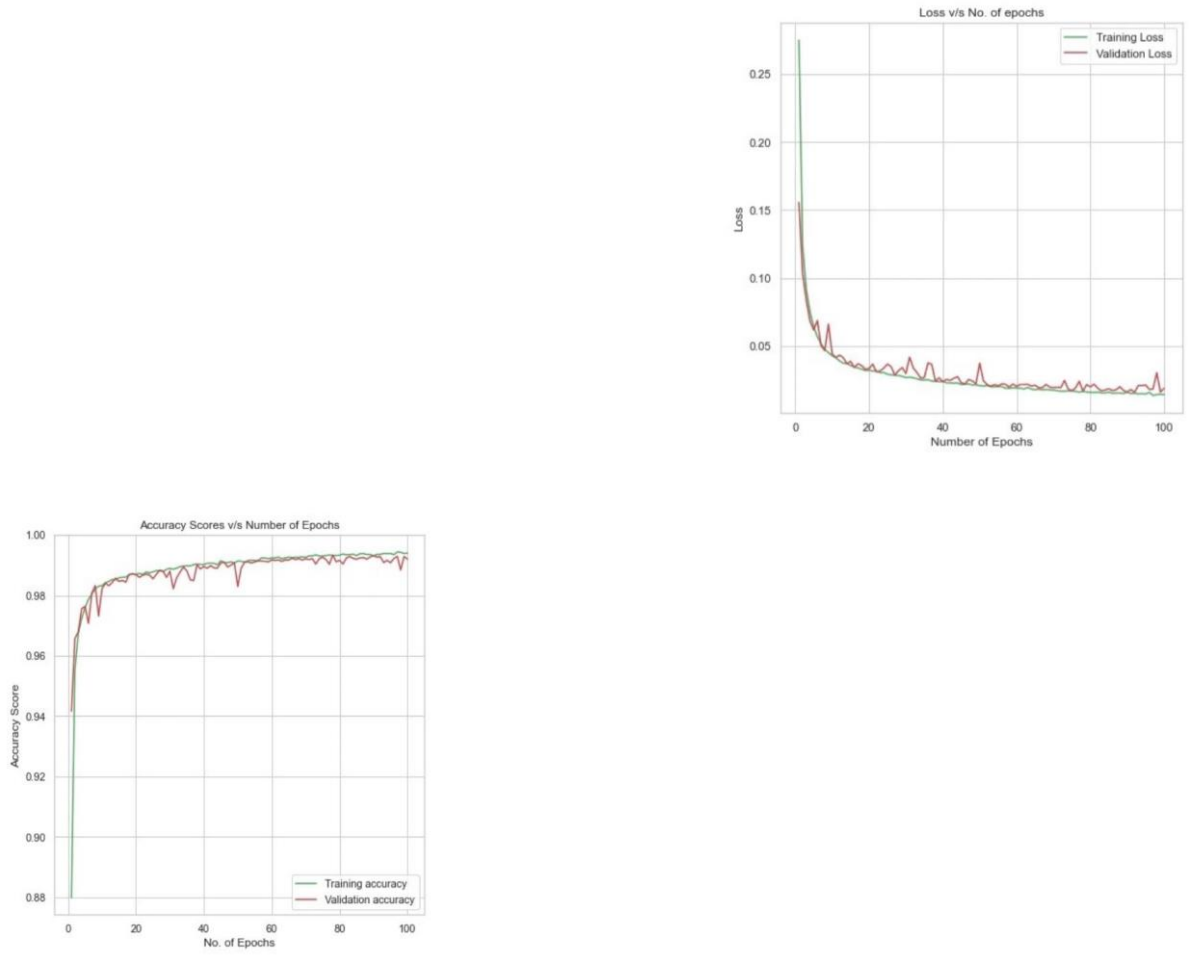


Figure 8 showing accuracy and no of epochs

5.6 BI-LSTM-GMM with Continuous Learning

With GMM, unlearned traffic can be captured and documented. An occurrence that deviates from the mean by three times, or x3 or x+3, is referred to under the proposed framework as a novel instance. Traffic engineers will record it and label it. Even though an unique occurrence may be malicious traffic or, more likely, a novel sort of legitimate traffic, data engineers must intervene. With in incremental learning phase, collected, tagged fresh instances are returned directly to the BI-LSTM and the GMM. After the loop has been completed, the results are shown in Table 5. The "CIC-IDS2017/Friday" has a significantly high detection accuracy of unknown traffic when compared to Table 4, both in terms of recall and accuracy.

Test Data set	Recall	Precision	Accuracy	AUC	F1
CIC-IDS2017/Wednesday	0.952	0.896	0.943	0.931	0.923

CIC-IDS2017/Friday	0.999	0.978	0.981	0.965	0.989

Table 6 Effectiveness of the incremental learning BI-LSTM-GMM.

Remain aware of the modest degradation in the detection of the previous attack, CIC-IDS2017/Friday. The result of progressive learning is this. An progressive training data set will slant an existing model and cause it to shift in the higher dimensional space. In "CIC-DDoS2019," we also used data sets to simulate hypothetical attacks in order to analyze the proposed framework's general operation. The outcomes are outlined. The biggest cause of misclassification is unknown traffic. The pure BI-LSTM, as shown in Table, is unable to appropriately counter unknown threats. The recall indicator drastically decreases as a result of unknown traffic. There is a persistent trend that the BI-LSTM by itself was unable to manage ambiguous traffics effectively. The GMM can, however, filter out new cases that weren't previously learned.

Model	**Test Data set**	**Recall**	**Precision**	**Accuracy**	**AUC**	**F1**
BI-LSTM	CIC-DDoS2019/NetBIOS	0.988	0.998	0.897	0.852	0.947
BI-LSTM-GMM	CIC-IDS2017/Wednesday	0.994	0.735	0.867	0.835	0.846
BI-LSTM-GMM	CIC-DDoS2019/NetBIOS	0.983	0.998	0.981	0.966	0.991
BI-LSTM	CIC-DDoS2019/NTP	0.363	0.994	0.367	0.607	0.562
BI-LSTM-GMM	CIC-IDS2017/Wednesday	0.986	0.751	0.876	0.851	0.932
BI-LSTM-GMM	CIC-DDoS2019/NTP	0.931	0.986	0.923	0.928	0.975
BI-LSTM	CIC-DDoS2019/LDAP	0.391	0.998	0.391	0.567	0.562
BI-LSTM-GMM	CIC-IDS2017/Wednesday	0.998	0.871	0.945	0.908	0.930
BI-LSTM-GMM	CIC-DDoS2019/LDAP	0.957	0.995	0.952	0.947	0.975

Table 7 BI-LSTM and BI-LSTM-GMM efficiency with "CIC-IDS2017/Wednesday" as the old traffic and "CIC-DDoS2019" as the new traffic.

An Open Set Recognition (OSR) difficulty in identifying unknown attacks can be successfully solved by integrating BI-LSTM with GMM. Identified new versions are supplied returned towards the BI-LSTM as well as the GMM enabling incremental learning with the aid of traffic engineers. The revised model can then gently and accurately handle both the new and the old traffic. The CIC-DDoS2019/NetBIOS performance decrease is not severe. Perhaps it has a similar traffic pattern like CIC-IDS2017/Wednesday. Both CIC-DDoS2019/NTP and CIC-DDoS2019/LDAP perform significantly better. However, overall productivity indicators rebound to adequate levels with the assistance of the suggested BI-LSTM-GMM architecture and the incremental learning strategy.

The suggested approach has produced encouraging findings in the identification of unidentified DDoS attacks. However, a number of difficulties necessitate further study to ensure its general application. The test datasets are the first problem. Prior to drawing a firm conclusion, further datasets must be validated. The second problem is that, for the current architecture, there is some trial-and-error involved in determining the BI-LSTM arrangement as well as the amount of kernels inside the Gaussian Mixture model. This vital system parameters' automatic setting will be a profitable area for investigation. The third issue is the continued need for traffic engineers' assistance. A global hub for the collecting of attack traffic that is constantly changing could be one solution to this problem.

Figure 9 visualization of continuous features

5.7 Data preparation

Before handling, data is arranged being an input at this phase. Although noisy, fragmented, or inconsistent input data would be present in the vast majority of cases, preprocessing is essential because it is frequently important to clean up the basic data any significant processing can begin. You could work on numerous data-related tasks while performing these tasks, including feature extraction, data transformation, and data cleaning.

5.8 Analysis of the experiment's results

The five-day log recordings from Monday through Friday were the focus of the experiment's dataset, which was in csv format. We took into account the Friday afternoon log file, which additionally contained two class labels, for experiment analysis. The classification names are DDoS and Benign (Normal) (attack). 225,746 traffic packets were included in the log file's total number of trac packets.

5.8.1 Experimental Assessment for the Log File with DDoS (Attack) and Benign (Normal) Class Labels

The Friday afternoon log file initially has 78 attributes, with the class label serving as the final attribute, making 79 dimensions total. The use of the feature selection procedure, which is based on the computing of mutual information for each of the features in the dataset, served as the initial step in the modeling process. Other characteristics in the set of attributes are dropped in favor of keeping the top 16 attributes. Regression analysis was done on the file system with these 16 attributes in order to execute multiple linear regression calculations for the purposes of mathematical modeling. Following first analysis, the characteristics at measures 1, 5, 6, 7, 9, 11, 13, 35, 36, 53, 54, 55, 56, 64, 66, and 67 are those that are kept. Thus, the analysis is carried out utilizing the reduced dimension log file that contains the aforementioned 16 properties. The linear regression model's absolute percentage mean error is calculated to be 0.2621. As a result, the multiple linear regression model's accuracy is calculated to be 73.79%, or 0.7379.

The residual plot in each of the 16 variables of the CICIDS 2017 database with respect to the Friday afternoon log file is shown in the figure. The fit chart and residual plot for the total multiple linear regression model are shown in Figure. Following the development of the initial model, we removed 10 attributes that are not statistically meaningful and kept the following 6 attributes. Thus, there are now only six qualities instead of the previous nine. The dimensions of the property are 1, 9, 13, 53, 54, and 64. Then, on these statistically significant attributes, the multiple regression evaluation is done out once more. The fit chart, which depicts the actual label as well as anticipated label, is shown in Figure 8. Figure 9 displays the residual plots for these six attributes, and Figure 10 displays the residual plot for such model. As a result, the machine learning model's accuracy using these six criteria is 71.6%, making it abundantly evident that the first model's 16 features are significantly superior.

The fit chart obtained by performing multiple regression analysis on the CICIDS 2017 input data with the top 16 variables allows the measurement through the information gain-based feature selection method is superior to the fit chart obtained by taking into account six attributes of the CICIDS 2017 dataset, as can be observed from the experiment result analysis. This is due to the fact that the discrepancy between the real label and projected label is readily obvious in the subsequent fit chart. As a result, the first model with 16 features is more effective in detecting DDoS attacks.

ANOVA					
	df	SS	MS	F	Significance F
Regression	16	29778.18946	1861.136841	23832.54328	0
Residual	225733	25640.72297	0.113588722		
Total	225749	55418.91243			

	Coefficients	Standard Error	t Stat	P-value	Lower 95%	Upper 95%	Lower 95.0%	Upper 95.0%
Intercept	1.48016318 1E0	1.214964E-3	1.218277377E3	0	1.477781883E0	1.48254448E0	1.477781883E0	1.48254448E0
Destination Port	-7.9586E-06	5.07513E-08	-1.568157955E2	0	-8.05807E-06	-7.85913E-06	-8.05807E-06	-7.85913E-06
Total Length of Fwd Packets	0	0	6.5535E4	#NUM!	0	0	0	0
Total Length of Bwd Packets	3.09845E-06	6.49544E-08	4.770183338E1	#NUM!	2.97114E-06	3.22576E-06	2.97114E-06	3.22576E-06
Fwd Packet Length Max	8.64604E-06	1.1702E-06	7.388505992E0	1.48996E-13	6.35248E-06	1.09396E-05	6.35248E-06	1.09396E-05
Fwd Packet Length Mean	0	0	6.5535E4	#NUM!	0	0	0	0
Bwd Packet Length Max	2.86673E-06	7.00377E-07	4.093126063E0	#NUM!	1.49401E-06	4.23945E-06	1.49401E-06	4.23945E-06
Bwd Packet Length Mean	7.06531E-05	3.26813E-06	2.161880723E1	1.5235E-103	6.42476E-05	7.70585E-05	6.42476E-05	7.70585E-05
Fwd Header Length	0	0	6.5535E4	#NUM!	0	0	0	0
Bwd Header Length	-5.97066E-4	5.71258E-06	-1.045177417E2	#NUM!	-6.08262E-4	-5.85869E-4	-6.08262E-4	-5.85869E-4
Average Packet Size	2.81867E-4	4.07743E-06	6.912847606E1	0	2.73875E-4	2.89858E-4	2.73875E-4	2.89858E-4
Avg Fwd Segment Size	-1.49379E-4	4.89145E-06	-3.053887974E1	2.0805E-204	-1.58966E-4	-1.39792E-4	-1.58966E-4	-1.39792E-4
Avg Bwd Segment Size	0	0	6.5535E4	#NUM!	0	0	0	0
Fwd Header Length	4.1925E-4	7.04326E-06	5.952493054E1	#NUM!	4.05445E-4	4.33054E-4	4.05445E-4	4.33054E-4
Subflow Fwd Bytes	-3.81369E-06	4.82535E-07	-7.903452827E0	2.72494E-15	-4.75944E-06	-2.86793E-06	-4.75944E-06	-2.86793E-06
Subflow Bwd Bytes	0	0	6.5535E4	#NUM!	0	0	0	0
Init_Win_bytes_forward	-1.24761E-05	9.91958E-08	-1.257727911E2	#NUM!	-1.26706E-05	-1.22817E-05	-1.26706E-05	-1.22817E-05

Figure 10 confidence interval and p-values

Concise summary of dataset

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 104345 entries, 0 to 104344
Data columns (total 23 columns):
 #   Column        Non-Null Count   Dtype
---  ------        --------------   -----
 0   dt            104345 non-null  int64
 1   switch        104345 non-null  int64
 2   src           104345 non-null  object
 3   dst           104345 non-null  object
 4   pktcount      104345 non-null  int64
 5   bytecount     104345 non-null  int64
 6   dur           104345 non-null  int64
 7   dur_nsec      104345 non-null  int64
 8   tot_dur       104345 non-null  float64
 9   flows         104345 non-null  int64
 10  packetins     104345 non-null  int64
 11  pktperflow    104345 non-null  int64
 12  byteperflow   104345 non-null  int64
 13  pktrate       104345 non-null  int64
 14  Pairflow      104345 non-null  int64
 15  Protocol      104345 non-null  object
 16  port_no       104345 non-null  int64
 17  tx_bytes      104345 non-null  int64
 18  rx_bytes      104345 non-null  int64
 19  tx_kbps       104345 non-null  int64
 20  rx_kbps       103839 non-null  float64
 21  tot_kbps      103839 non-null  float64
 22  label         104345 non-null  int64
dtypes: float64(3), int64(17), object(3)
memory usage: 18.3+ MB
```

Figure 11 concise summary of dataset

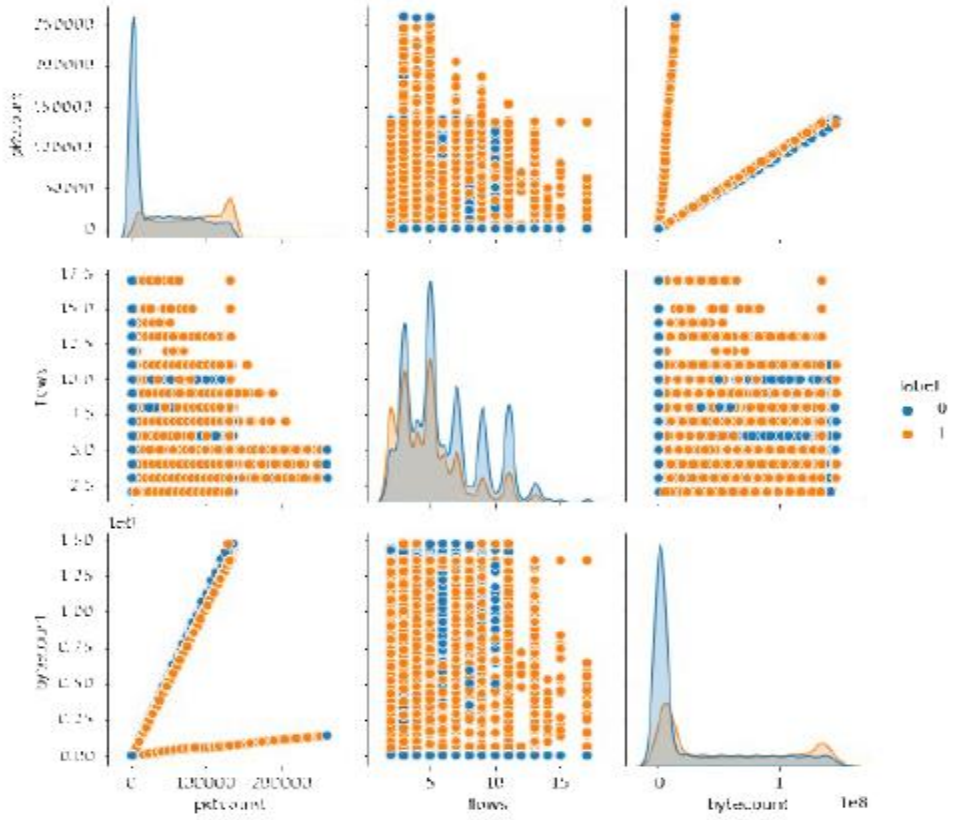


Figure12 Pair plots of selected features

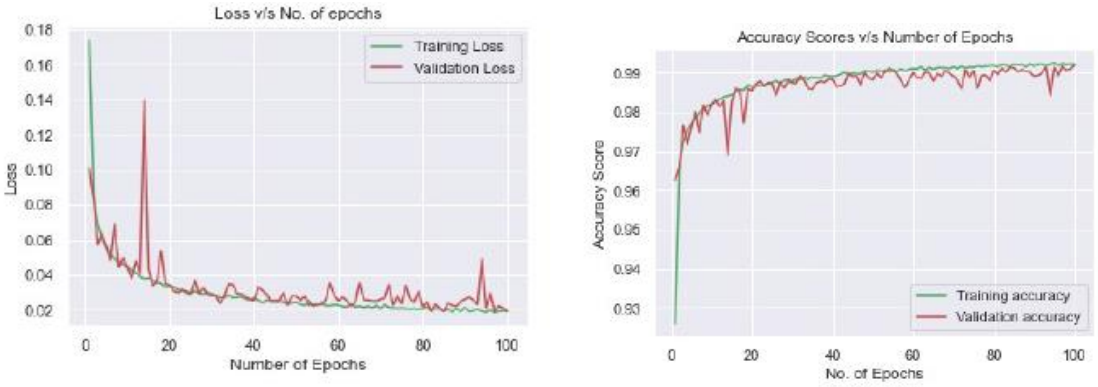


Figure 13 regression approach

5.9 Methodology

This section goes into great detail about our experimental procedure, the datasets we utilized to evaluate our experiment, how we chose the features, and how we went about detecting DDoS. We investigate the possibility of DL methods for SDN environments DDoS attack detection.

5.9.1 Dataset Description

The caliber of the training datasets is crucial to the development of a successful anomaly detection-based IDS. However, a major issue is the lack of readily available high-quality databases for malware detection as well as network traffic in general. Numerous high-quality datasets from several application domains, including translation software and computer vision, are accessible to the public online. On

the other hand, network data may contain private customer data, and it is prohibited from making such information public. As a result, the performance of the classifier models is significantly changed by the fact that the majority of different datasets for vulnerability scanning are anonymized payload data. Furthermore, the vast majority of datasets are out-of-date, have low traffic variety, and are useless for current attack detection methods. Since network traffic is fluid and businesses might constantly alter the used protocols, using incompatible datasets can result in a rigid model and a contradiction between both the models and the new technology. For instance, until 2010, the most well-known video companies, such as YouTube and Netflix, used the Flash as well as Silverlight protocols extensively. The HTML5 protocol has taken their place at the moment. As a result, a number of research projects have been suggested to simulate new databases for the research. One of the important centers in the world, the Canadian Institute for Cybersecurity (UNB), has helped to provide trustworthy and validated datasets. Utilizing network typologies that resemble the actual network data centers, the UNB developed intrinsic datasets that are accessible to the general public. Although the statistics created by UNB are widely used in many SDNs research projects, they were not created using SDNs but rather traditional or conventional IP networks. The Internet protocol network and SDN, however, operate very differently, as was previously covered in section II. In addition, the network becomes vulnerable to new attack vectors that are distinct from those observed in conventional IP networks as a result of the decoupling of the management plane from the routing protocol. Detaching the SDN controller from the connected devices, for instance, raises the likelihood that an attacker may launch different kinds of attacks against data communications systems or the SDN controller itself. Due to the attacker's authorized connection to the target server, these assaults are difficult to identify. Therefore, utilizing an inappropriate dataset can trick the detection algorithm and lead to a significant number of false alarms. Along with the above mentioned issue, to the best of our knowledge, there is currently no publicly accessible dataset for assessing and testing IDS in the context of SDNs. The majority of the work done on anomaly identification in SDNs has utilized typical datasets created using the traditional network. We utilized the InSDN dataset to examine the effectiveness of the suggested DL models to address each of these issues. This work additionally uses the CICIDS2017 and CICIDS2018 databases for additional examination. The following is a discussion of the three separate datasets:

InSDN: The InSDN dataset takes into account the updated SDN network structure. It was made with the aid of four virtual servers (VMs). The first VM served as an SDN controller, or ONOS, and the second one served as just an Open Virtual Switches (OVS). Kali Linux, the third virtual system, served as the invader machine, and Metasploitable2, among other venerable programs, was loaded on the last virtual machine. A mininet emulator program was also used to construct four internal virtual machines to impersonate both legitimate users and some malicious hosts. A number of attack classes from both within and outside the SDN network were thus simulated by the dataset. Several application services, including HTTPS, DNS, SSH, FTP, email, and others, were reflected in the usual flow in the InSDN dataset. Some internal hosts are permitted to access the Internet for this reason and gather intrinsic traffic from many websites, like YouTube, Facebook, SKYPE, and others, to simulate real-world traffic.The InSDN dataset has a total of 361; 317 occurrences, with samples for the normal and attack classes having sizes of 69; 423 and 291; 894, correspondingly.

CICIDS2017: The dataset included five days' worth of network traffic that was created between Monday, July 3 and Friday, July 7, 2017. A network connectivity topology with several devices, including router, ports, firewalls, and various operating systems platforms, was used to develop the CICIDS2017. The usual flow in the datasets was created by the authors using the notion of profiles. The data set was accessible to the general public online in PCAP and.CSV formats. The overall number of instances in the CICIDS2017 is 2,830,743, and the size of the attacks made up 19.7% of the entire data.

CICIDS2018: To create a fresh realistic database in a scalable way, the authors expanded the CICIDS2017 project. A total of 16,233,002 instances from the CICIDS2018 footprints were collected in 10 days, and the size of the attacks accounted for 17% of the overall data. The authors created the normal and attack classes using the same concept of profiles, but they did so using the Amazon Web Services

(AWS) infrastructure rather than the outdated network infrastructure. With much more over 80 network flow characteristics in the form of CSV files, the 3 dataset features were created through using CIC Flow Meter program. Different attack classes can be found within the three datasets. Other attack classes are not included in our study because this work exclusively focuses on DDoS attacks. We take all labels classified under the regular and DDoS classifications because the InSDN sample is considerably smaller than other datasets. However, just the Friday afternoon (July 7) data from the CICIDS2017 dataset is chosen for our studies, whereas the CICIDS2018 dataset uses the Wednesday (February 21) file.

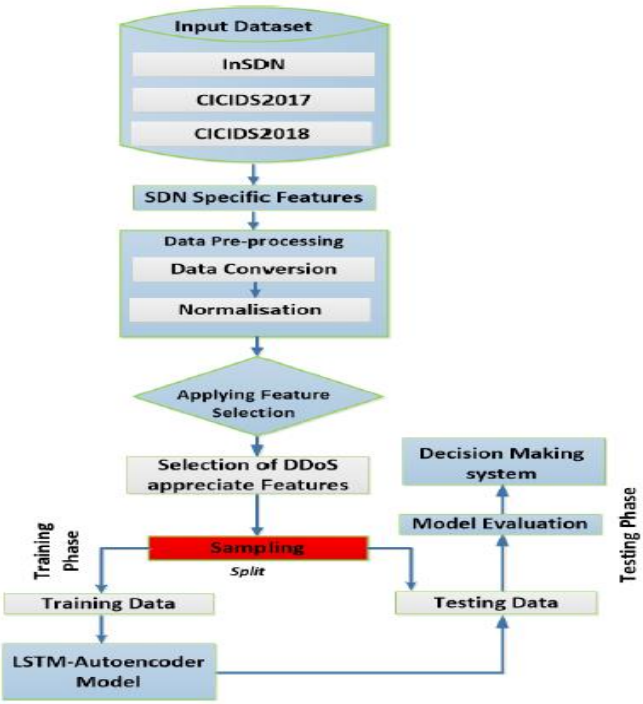


Figure 14 The flow diagram of the DDoS Detection mode

5.9.2 Features Particular to SDN

In this part, we'll go over a few typical features that apply to SDN networks. The application software CICFlowMeter tool was used to obtain the characteristics of the three datasets. And over 84 flow features are generated by the CICFlowMeter. Several of these capabilities, nevertheless, are extractable within the SDN context. Utilizing Open Flow calls towards the SDN switches, only statistical features in SDN may be collected from the SDN controller. We employ the same architecture to achieve this goal in order to locate the sub-features, which can be quickly obtained either directly from the SDN controller source or by competitive analysis of the flow statistics. For instance, we can manually compute various attributes like standard deviation, minimum and maximum values, and mean of flow features. The table shows the features in the InSDN dataset mapped to the derived features from the SDN environment. Additionally, Table displays extra functionality that can be estimated from the set of 50 variables that were used for their research goal. Only a portion of the 48 features are utilized in this post because the source and destination Intrusion detection system are not used in our research. The two characteristics are interchangeable between networks, and an attacker can also employ a valid user's IP address.

As a result, utilizing these features to train a classifier model can bias the model toward particular socket features, leading to an overfitting issue. The obtained SDN features are shown in Table.

	switch	flows	Pairflow	port_no
0	1	3	0	3
1	1	2	0	4
2	1	3	0	1
3	1	3	0	2
4	1	3	0	3
5	1	3	0	1
6	1	3	0	4
7	1	3	0	1
8	1	3	0	2
9	1	3	0	4

Table8 Discrete numeric features

5.9.3 Data Preparation

To create an accurate detection system, data preprocessing must be done on the input data before model training. The original data cannot be used to create or train ML/DL models, hence the following procedures are taken to make the input dataset intelligible and readable:

- Many infinite & missing (nan) values can be found in the CICIDS2017 as well as CICIDS2018 datasets. Data cleaning is the first step in creating an accurate model. In actuality, there are two different approaches that may be used to manage the missing and infinite values in a given column. We have two options: either we eliminate these data or we compute the mean and then add the results. Because the two datasets in this study have sufficient sample sizes, we can eliminate every null and infinity values without significantly affecting the model's effectiveness.
- Because ML/DL approaches are built on mathematical equations, categorical data must be transformed into numerical values in order to preserve the integrity of the equations. The designated column's content is changed to a number using the OneHotEncoder class. These experiments categorize DDoS or regular attacks using simply binary classification. As a result, DDoS attacks are coded as 1, whereas regular traffic takes the number of 0.
- The varying sizes of the dataset's characteristics can interfere with the DL model in several ways. For instance, some features or columns in a dataset may have a narrow range of values, whereas other columns may accept values that are higher than those in other columns. Utilizing feature scaling, we restrict the variety of variables so that the common ground may be used for comparison. Scaling can be done in two main ways: normalization and standardization. The standardization transforms the input variables into a new scale with a zero mean () and a one standard deviation () whereas the normalization adjusts the attributes between 0 and 1. For all datasets in this study, we used the standardization approach in accordance with Eq. 1.

$$x(i) = \frac{x(i) - \mu(x(i))}{\sigma(x(i))}$$

- Using test train split from the sklearn module, we divided the dataset into a 75:25 ratio, which implies that 70percent of total of the dataset was used for training and the remaining 30% was set aside for the model test to see how well we could predict it. Graph depicts the total number of testing and training samples for all datasets.

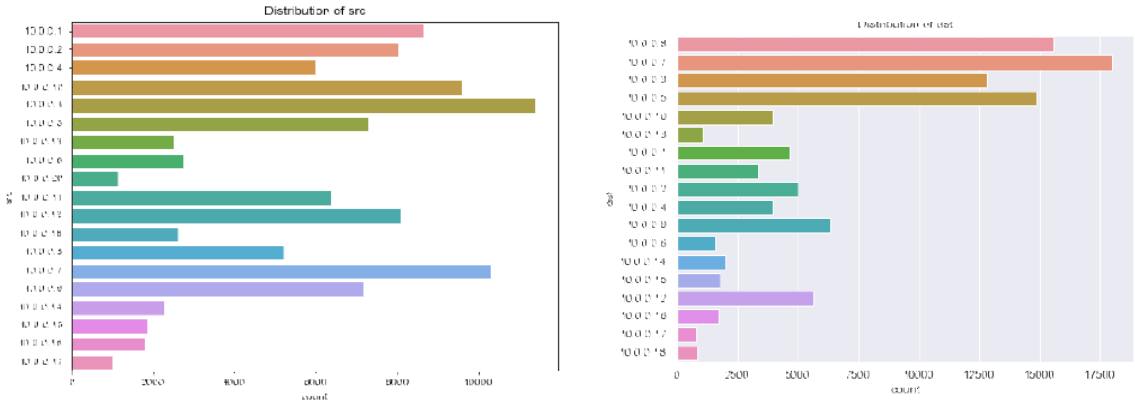


Figure 15 Showing frequency distribution

5.9.4 Deep Learning Classifier

Recently, DL methods have become more and more popular for a wide range of applications, including natural language processing, computer vision, and language translation. Multiple hidden layers are used in DL approaches to handle increasingly complex issues that are challenging to solve with a linear function. The traditional approaches frequently involve extensive feature engineering, whereas the DL methods have the ability to automatically extract the features from input data without human intervention, addressing the limitations of standard ML algorithms. When dealing with data points with high levels of nonlinearity, DL performs noticeably well. As a result, it is anticipated to enhance cybersecurity trends like IDSs. This section outlines the DL strategy for addressing the issue of DDoS attacks in SDN networks.

5.10 The suggested DL Model

The suggested deep learning model is based on our prior model, DDoSnet, which combines autoencoder and recurrent neural networks (RNN). In compared to conventional ML algorithms, the suggested model successfully recognized DDoS attacks with outstanding performance and few false alarms. To get around the issue of vanishing gradient, we improve model performance in this article by employing LSTM, a particular sort of RNN, rather than simple RNN. A neural network's weight values are updated via gradients. However, as the gradient value backpropagates through time, it does not significantly contribute to learning when it is exceedingly small. Small gradient modifications hurt the RNN, particularly in earlier layers. As a result, it cannot retain the information for lengthy sequences.

New weight = weight - learning rate * gradient

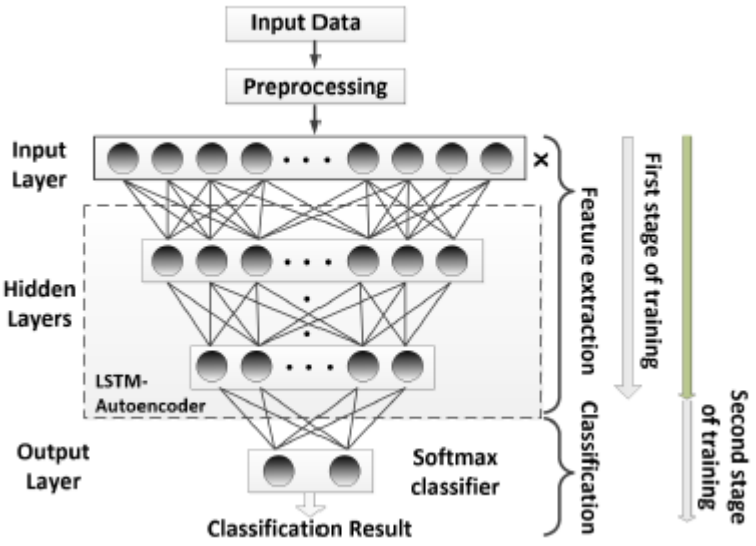


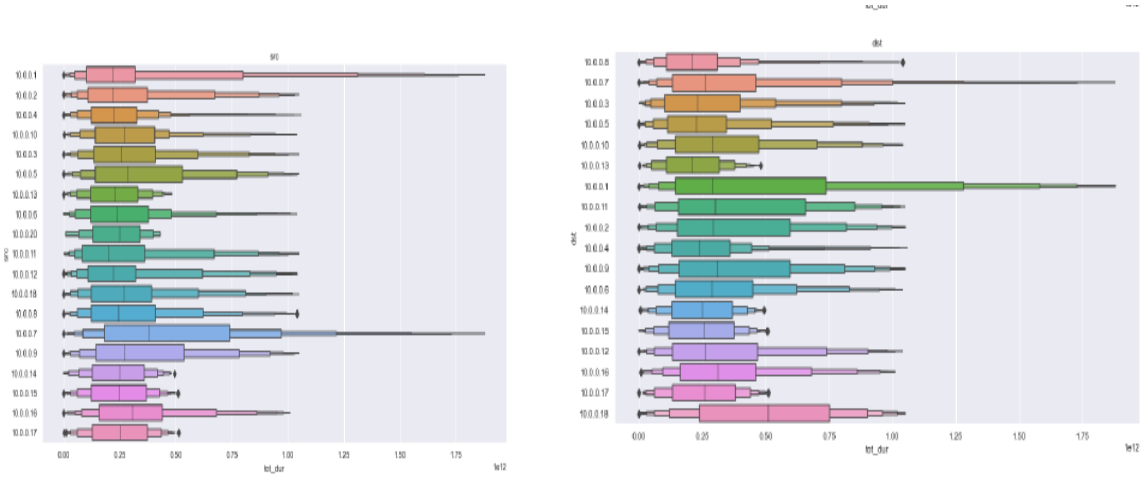
Figure 16 DL method structure

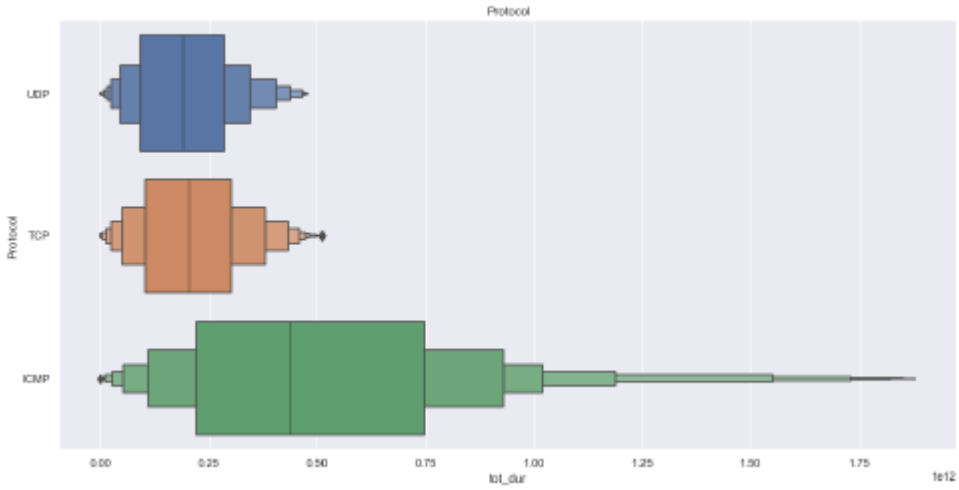
The model has two phases: the first employs unsupervised learning for the basic phase, while the second uses supervised learning for the fine-tuning phase. Without any labels, unsupervised learning is used in the first stage to identify distinguishing characteristics in the raw data. The autoencoder converts representational information from its native space into a different one. To enhance model performance, an LSTM layer is added in place of each dense layer in the original autoencoder. As temporal correlations of input data creates sequential traffic, the main concept driving the use of LSTM in DL method is the character of network traffic. As a result, using such techniques to develop DL models will remove loss because the output of each layer is independent of both its input and output history.

Setting for the experiment: Due to the absence of theoretical support, tuning hyper-parameter values is one of the major obstacles in DL training. Consequently, there is no magic formula for selecting the ideal hyper-parameter values. To determine the optimal amount of network layers, neurons in each layer, iterations, batch size, etc., numerous tests as well as variations based on trial and error have been carried out. We examine the effectiveness of the DL model by evaluating several values of learning rate (), i.e. 0:001, 0:01, and 0:1, in order to show the optimal values of hyperparameters. We analyze the model's performance for every value of for several other parameters. Whenever the amount of hidden layers is equivalent to three, the classifier accuracy is at its best. Repeating trials numerous times revealed that using the hyper-parameters listed in Table VI led to the best performance for the supplied data. Through one dataset to another, hyper-parameter values can be altered. Since the fluctuation in the findings can be disregarded, we utilized the very same hyper-parameters as listed in Table for all samples to keep things simple.

The 3 hidden layers at the encoder phase successively lower the input dimensions to 32, 16 and 8 pixels. Compressed input data is the phase's end result after encoding. With 8, 16, and 32 channels, respectively, the decoded phase is performed in the opposite sequence of the encoded phase. The hierarchy features are collected from the unlabeled data after the model has been built and the best weight and bias values have been determined. In the second step, labeled data is used for fine-tuning to optimize the network and train the top layers of the network. Finally, by including the softmax activation function at the output nodes, the model's output is obtained. For each output class, the softmax layer produces an output in the region of (0, 1), with a probability of 1 for all output classes. We only looked at samples from traditional and DDoS attacks in this study. Therefore, binary categorization assigns a value of 0 to legitimate attacks and a digit of 1 to malicious or DDoS attacks. The softmax layer uses categorical cross-entropy,

and the Adam optimiser is utilized to iteratively update the network weights. 100 periods of history and 128 batches were used to train the model.





5.10.1 Feature identification algorithm

This section makes use of feature selection algorithms to identify the pertinent DDoS attack attributes in each dataset independently. The feature selection algorithms only consider the features that are most pertinent to each class label, disregarding any redundant or unimportant features. In order to create a classifier that is lightweight and resistant to overfitting, it may be helpful to train the detection model using a limited set of features. Additionally, the lightweight model can be quickly implemented in a network platform without having a substantial negative impact on system resources [8]. Although the approach to determining the most important features varies depending on the feature selection algorithm, we used two alternative techniques: Random Forest (RF) and Information Gain.

IG: One of the most widely used algorithms to determine total amount each variable affects the choice is Information Gain (IG). It falls under the genre of filter methods and uses the idea of information theory to determine the significance of features. Shannon entropy is a frequently used indicator of information. The entropy assesses each feature's uncertainty in terms of how important it is for classifying data. The following equation can be used to determine the entropy for a certain attack class H(C):

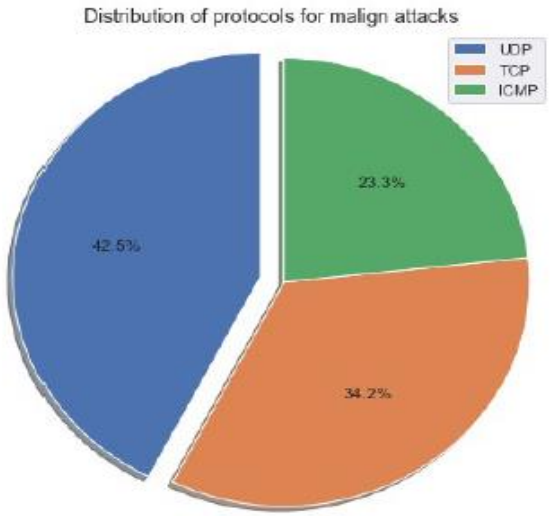
$$H(C) = -\sum_{i=0}^{n} \rho(i) log \rho(i)$$

By calculating the information weight of every feature and removing the extraneous information for each class label, the IG employed a straightforward attribute rank. A feature with a tiny information gain also has a small impact on the categorization of the data and can be disregarded without harming the model's performance. The entropy reduction is calculated using the following equation to determine the IG for every single given input F in the dataset:

$$IG(C;F) = H(C) - H(C|F)$$

where IG(C;F) is the feature F's information gain, taking taking C's class characteristics into consideration, and H(C—F) is C's average conditional entropy.

Random Forest: Has an excellent predictive performance and is less prone to overfitting, and is frequently employed to address the issue with individual Decision Trees. It falls under the umbrella of Embedded methods, a category that mixes filter and wrapper methods. The main goal of the RF is to quantify the contribution of each feature to the prediction. This is a crucial variable if change is substantial. In a similar vein, a modest change indicates that the feature doesn't offer much insight. The RF is a mix of thousands of built-in decision trees based on random dataset observation and random separating characteristics. Because the trees are de-correlated, the amount of features can vary from one to the next, protecting the model from overfitting. Each variable's weight is determined by the RF in one of two ways. The basis for the first metric is the reduction in Gini distortion when a variable is selected to separate a node. The second metric looks at how much accuracy is lost when the variable is taken out.

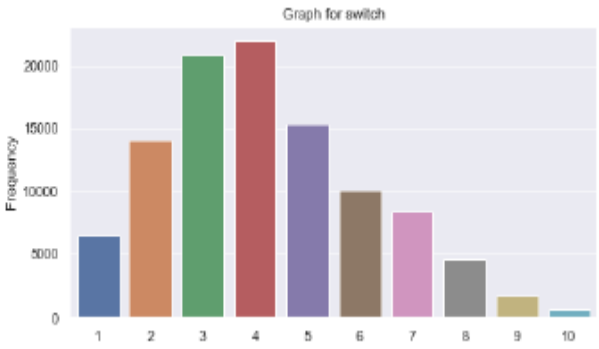


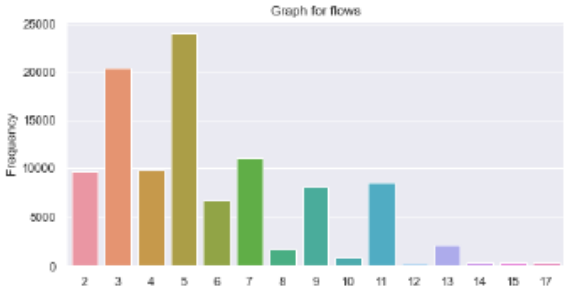
5.11 Experimental Findings

The experimental findings of the suggested strategy are covered in this section. The percentages of Precision, Recall, as well as f1-score for samples with various subsets of features are shown in Table IX. When 48 subfeatures were utilized in the training phase, the model produced the best results; however, when only 10 subfeatures were used for the RF and IG algorithms, the model's performance somewhat decreased. The performance reduction can be disregarded, allowing us to create a lightweight model with fewer features. Making a model that is lightweight will use fewer resources and render it better suited for the SDN framework. Additionally, it should be noted that the IG algorithm performs somewhat better overall than the RF approach. We further verify our assertion and show how using different datasets generated in various contexts can dramatically reduce the model's performance.

The proposed DL technique is trained using the CICIDS2017 dataset during the training phase, but its performance is evaluated using the test portions of the InSDN and CICIDS2018 databases. The model is initially trained and tested using all 48 subfeatures, and only the top 10 RF and IF features, which were previously chosen in light of the CICIDS2017 dataset, are then used. The results demonstrate that the measurement methods are extremely greater and very close to those presented in Table if we test performance of the model on the CICIDS2018 dataset using a subset of 48 features. On the InSDN dataset, performance has, however, drastically decreased. The stated accuracy is 99.61% for the CICIDS2018 dataset and 55.18% for the InSDN dataset, respectively. Additionally, both datasets' performance is greatly decreased when the model has been trained on just 10 features, although the decline for InSDN is much worse than it is for CICIDS2018. For the InSDN dataset, the model was unable to detect any DDoS entities. The overall accuracy for the RF technique is 88.21% and 36.22% for the CICIDS2018 and InSDN datasets, respectively, whereas the overall accuracy for the IG approach is 89.09% and 32.94%. We display the model's execution time for all dataset, to further assess the model's performance and demonstrate the impact of the component selection procedure. The graph demonstrates that the model's execution time for CICIDS2018 is very high while it is minimal for InSDN.

This is as a result of the unusually large sample sizes in CICIDS2018 when contrasted to other datasets. Once all 48 subfeatures are included, it is also observed that the model took a long time to train, although the execution time was very short for RF but significantly longer for IG.





CHAPTER 6

CONCLUSIONS

In this work, a Gaussian Mixture Model and Deep Learning (DL) are used to demonstrate how DDoS attacks can be detected (GMM). The suggested system uses GMM, incremental learning, and bi-directional prolonged short-term memory (BI-LSTM) to address the Open Set Recognition (OSR) issue in DDoS detection. The Bi-LSTM has proven to be an effective method for differentiating between

malicious and lawful traffic samples taken from either the dispersion of the training data. The GMM has proven to be a useful tool for differentiating between untrained samples and trained cases. The GMM can record unknown traffic, label it with the help of data engineers, and afterwards feed that labeled information back to the BI-LSTM and GMM for learning algorithms. The upgraded model can accurately and gracefully manage both the new and the old traffic. A number of tests using the data sets CIC-IDS2017 and CIC-DDoS2019 have verified the viability and efficacy of the proposed framework.

BI-LSTM is completely capable of carrying out the tasks for which it has been trained, such as identifying recognized DDoS attacks. However, the system performance suffers greatly when faced with innovative attacks. The recall decreases for Dataset CIC-IDS2017/Wednesday and Friday from 99.8 to 41.2 percent. GMM allows for the capturing of unknown traffic, which traffic engineers can later annotate. For Dataset CIC-IDS2017/Wednesday and Friday, incremental learning then raises detection rates to 95.3 percent and 99.8 percent, respectively. Some research topics need more attention if the suggested BI-LSTM-GMM architecture is to be used to its full potential. These directions include validation on other databases, auto configuration of BI-LSTM as well as GMM, as well as the removal of involvement transportation systems.

It is crucial to identify assaults that result in Cloud services being unavailable since DDOS attack detection has increased in frequency in dispersed environments like the cloud. Machine learning models have the potential to train and test attack detection datasets in order to recognize such attacks. As an alternative, we can employ the regression analysis method by using multiple linear regression analysis, one of its crucial forms. The goal of this study's research is to create a machine learning model that combines feature selection with regression analysis and information acquisition. The dataset taken into consideration for the experimental study was from the well-known CICIDS 2017 dataset. In particular, the Benign, Bot, and DDoS classes in the Friday morning and afternoon logfile are taken into account. This ensemble model for the Friday morning dataset has been seen to attain a predictive performance of 97.86%. Similar results were obtained for the Friday afternoon log file, where the prediction accuracy for 16 characteristics derived from feature selection based on information gain and ML model based on regression analysis was 73.79%. Thus, this study set the road for demonstrating the value of regression analysis in creating ML models. It also illustrates certain crucial visualizations, including such residual plots and fit charts, which demonstrate the significance of the model and demonstrate its applicability for prediction. In this work, we have restricted our analysis to one day's worth of log files; however, in the future, this research may be expanded to take into account all five days' worth of trac log files and produce a prevailing opinion machine learning model.

The complexity of the suggested classifier rises during intrusion detection network system training utilizing a high-dimensional dataset, resulting in long training and classification times. To increase classification accuracy and avoid the curse of excessive computational complexity, pre-processing feature selection approaches are crucial in selecting the significant characteristics from the original dataset. The objective of this research is to eliminate redundant or pointless features while minimizing any negative effects on classification accuracy. Using two widely used feature selection techniques, IG and RF, we have chosen 10 features from the 48 available features. DDoS attacks were employed as a case study, and a redesigned DL model is developed based on LSTMAutoencoder was utilized for experimental reasons. Our method offers a high detection accuracy and a quicker, more effective way to create the model. To determine how the used dataset might influence the effectiveness of the SDN controller, we subsequently tested the trained model on the SDN controller's performance. The outcomes demonstrated that the suggested strategy did not impair network performance. We will examine fresh assault categories for the test evaluation in our next work. Additionally, we intend to test our suggested model on an actual SDN network in order to better understand how well this IDS can respond to an incursion in real-time.

References

Ahanger, T. A. (2017). An effective approach of detecting DDoS using artificial neural networks. *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, 707–711.

Alghazzawi, D., Bamasag, O., Ullah, H., & Asghar, M. Z. (2021). Efficient detection of DDoS attacks using a hybrid deep learning model with improved feature selection. *Applied Sciences*, *11*(24), 11634.

Baig, Z. A., Sait, S. M., & Shaheen, A. (2013). GMDH-based networks for intelligent intrusion detection. *Engineering Applications of Artificial Intelligence*, *26*(7), 1731–1740.

Bendale, A., & Boult, T. E. (2016). Towards open set deep networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1563–1572.

Bhaya, W., & Manaa, M. E. (2014). A proactive DDoS attack detection approach using data mining cluster analysis. *Journal of Next Generation Information Technology*, *5*(4), 36.

Blomstrom, M., & Kokko, A. (1998). MNCs and spillovers. *Journal of Economic Surveys*, *12*, 97–110.

Cheng, J., Yin, J., Liu, Y., Cai, Z., & Wu, C. (2009). DDoS attack detection using IP address feature interaction. *2009 International Conference on Intelligent Networking and Collaborative Systems*, 113–118.

Fadlil, A., Riadi, I., & Aji, S. (2017). Review of detection DDOS attack detection using naive bayes classifier for network forensics. *Bulletin of Electrical Engineering and Informatics*, *6*(2), 140–148.

Geng, C., Huang, S., & Chen, S. (2020). Recent advances in open set recognition: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *43*(10), 3614–3631.

Gupta, A. (2018a). *Distributed Denial of Service Attack Detection Using a Machine Learning Approach*. Graduate Studies.

Gupta, A. (2018b). *Distributed Denial of Service Attack Detection Using a Machine Learning Approach*. Graduate Studies.

He, Z., Zhang, T., & Lee, R. B. (2017). Machine learning based DDoS attack detection from source side in cloud. *2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud)*, 114–120.

Hoque, N., Bhattacharyya, D. K., & Kalita, J. K. (2015a). Botnet in DDoS attacks: Trends and challenges. *IEEE Communications Surveys & Tutorials*, *17*(4), 2242–2270.

Hoque, N., Bhattacharyya, D. K., & Kalita, J. K. (2015b). Botnet in DDoS attacks: Trends and challenges. *IEEE Communications Surveys & Tutorials*, *17*(4), 2242–2270.

Jonker, M., Sperotto, A., & Pras, A. (2020). DDoS Mitigation: A measurement-based approach. *NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium*, 1–6.

Li, Y., & Lu, Y. (2019). LSTM-BA: DDoS detection approach combining LSTM and Bayes. *2019 Seventh International Conference on Advanced Cloud and Big Data (CBD)*, 180–185.

Mahjabin, T., Xiao, Y., Sun, G., & Jiang, W. (2017). A survey of distributed denial-of-service attack, prevention, and mitigation techniques. *International Journal of Distributed Sensor Networks*, *13*(12), 1550147717741463.

Osanaiye, O., Cai, H., Choo, K.-K. R., Dehghantanha, A., Xu, Z., & Dlodlo, M. (2016). Ensemble-based multi-filter feature selection method for DDoS detection in cloud computing. *EURASIP Journal on Wireless Communications and Networking*, *2016*(1), 1–10.

Peng, J., Choo, K.-K. R., & Ashman, H. (2016). Bit-level n-gram based forensic authorship analysis on social media: Identifying individuals from linguistic profiles. *Journal of Network and Computer Applications*, *70*, 171–182.

Pouyanfar, S., Sadiq, S., Yan, Y., Tian, H., Tao, Y., Reyes, M. P., Shyu, M.-L., Chen, S.-C., & Iyengar, S. S. (2018). A survey on deep learning: Algorithms, techniques, and applications. *ACM Computing Surveys (CSUR)*, *51*(5), 1–36.

Pouyanfar, S., Yang, Y., Chen, S.-C., Shyu, M.-L., & Iyengar, S. S. (2018). Multimedia big data analytics: A survey. *ACM Computing Surveys (CSUR)*, *51*(1), 1–34.

Ranjan, R., & Sahoo, G. (2014). A new clustering approach for anomaly intrusion detection. *ArXiv Preprint ArXiv:1404.2772*.

Rudd, E. M., Jain, L. P., Scheirer, W. J., & Boult, T. E. (2017). The extreme value machine. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *40*(3), 762–768.

Sabeel, U., Heydari, S. S., Mohanka, H., Bendhaou, Y., Elgazzar, K., & El-Khatib, K. (2019). Evaluation of deep learning in detecting unknown network attacks. *2019 International Conference on Smart Applications, Communications and Networking (SmartNets)*, 1–6.

Sarker, I. H., Kayes, A. S. M., Badsha, S., Alqahtani, H., Watters, P., & Ng, A. (2020). Cybersecurity data science: An overview from machine learning perspective. *Journal of Big Data*, *7*(1), 1–29.

Shieh, C.-S., Lin, W.-W., Nguyen, T.-T., Chen, C.-H., Horng, M.-F., & Miu, D. (2021a). Detection of unknown ddos attacks with deep learning and gaussian mixture model. *Applied Sciences*, *11*(11), 5213.

Shieh, C.-S., Lin, W.-W., Nguyen, T.-T., Chen, C.-H., Horng, M.-F., & Miu, D. (2021b). Detection of unknown ddos attacks with deep learning and gaussian mixture model. *Applied Sciences*, *11*(11), 5213.

Shieh, C.-S., Lin, W.-W., Nguyen, T.-T., Chen, C.-H., Horng, M.-F., & Miu, D. (2021c). Detection of unknown ddos attacks with deep learning and gaussian mixture model. *Applied Sciences*, *11*(11), 5213.

Siddiqi, M. A., & Pak, W. (2020). Optimizing filter-based feature selection method flow for intrusion detection system. *Electronics*, *9*(12), 2114.

Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). A detailed analysis of the KDD CUP 99 data set. *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, 1–6.

Vu, N. H., Choi, Y., & Choi, M. (2008). DDoS attack detection using K-Nearest Neighbor classifier method. *Proceedings of the 4th IASTED International Conference on Telehealth/Assistive Technologies. Baltimore, Maryland, USA*, 248–253.

Wang, C., Zheng, J., & Li, X. (2017). Research on DDoS attacks detection based on RDF-SVM. *2017 10th International Conference on Intelligent Computation Technology and Automation (ICICTA)*, 161–165.

Yong, B., Wei, W., Li, K.-C., Shen, J., Zhou, Q., Wozniak, M., Połap, D., & Damaševičius, R. (2022). Ensemble machine learning approaches for webshell detection in Internet of things environments. *Transactions on Emerging Telecommunications Technologies*, *33*(6), e4085.

Yulita, I. N., Fanany, M. I., & Arymuthy, A. M. (2017). Bi-directional long short-term memory using quantized data of deep belief networks for sleep stage classification. *Procedia Computer Science*, *116*, 530–538.

Author short profile

SHEIKH SAZIB
Major: computer science and applications
Student number:GJ201963120066
School of Computer and Communications Engineering,

Changsha University of Science and Technology
Changsha, Hunan, China.410114
E-mail： mdsazibsheikh380@gmail.com