National College of Ireland

An SDN based Machine Learning and Deep Learning model for DDoS attack detection on IoT networks.

MSc Research Project
Cybersecurity

Akash .
Student ID: X20258194

School of Computing
National College of Ireland

Supervisor: Michael Pantridge

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: ……Akash . ……

Student ID: …X20258194……..……

Programme: …Cybersecurity……**Year:** 2021-2022…

Module: …MSc Research Project……..……

Supervisor: …Michael Pantridge……..……

Submission Due Date: …15/08/2022……..……

Project Title: …An SDN based Machine Learning and Deep Learning model for DDoS attack detection on IoT Network……

Word Count: …7625…… **Page Count**……23……..……..

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: …Akash . ……

Date: …14/08/2022……

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

An SDN based Machine Learning and Deep Learning model for DDoS attack detection on IoT networks.

Akash.
X20258194

Abstract

IoT has been around for more than 30 years now and has been a significant factor in people's lives, and that will continue to be the case in the future as well. Among the most advanced and growing technologies in the world is the Internet of Things. It is important to note that a lot of IoT devices are connected to the internet, and these internet nodes contain a lot of resources. These resources might contain sensitive information, making IoT devices a prime target for cybersecurity attacks. Many attempts have been made to combat the dangers of IoT and improve the network's adaptability and speed. The problem with these techniques is that they have not provided a complete defence to the IoT network from attacks and there has not been any combination of Software based networking with machine learning and deep learning discussed profusely for IoT environment and this reason is why this research proposal is focused on finding a better way to stop or minimize these attacks by using machine learning and deep learning with SDN to make IoT systems adaptive and faster.

The bot-Iot dataset offered by University of New South Wales is used for the proposed model. The model is trained with only DDoS labelled traffic from dataset and goes through feature selection and PCA for better results. Upon testing the model with different algorithms of Machine learning and Deep Learning. It was found that they all produce a similar accuracy with Decision Tree providing 86.359% and all other algorithms were compared. The model also achieves its aim to produce null or negligible False Negative.

Keywords: DDoS attack detection, CNN, Decision Tree, Random Forest, Gaussian NB, IoT networks

1 Introduction

The Internet of Things (IoT) network is a network of interconnected wireless or distributed devices that could conduct activities utilizing the Internet as a channel of communication. The Internet of Things may also be defined as a network that includes physical objects that have been provided with technologies for communication, processing, and storage that would otherwise have limited capabilities. From digital computing machines to smart appliances that are human interactive, these machines can do anything. Basically, any device with a processor and a connection to the internet that sends and receives signals is termed as an IoT device. Sensor data generated by IoT devices is in their raw form unable to be understood until it is processed. The system then uses the processed signal for completing tasks. Together, these machines create an automated system that simplifies the human task,

reducing labour costs and improving the supply chain of businesses by further simplifying it. It is estimated that by 2025, there will be 40 billion IoT devices in the world, up from 30 billion today. Data is collected and sometimes processed by IoT devices before being exchanged through sensors, ports, and network connectors.(Stellios et al., 2018)

There are several types of devices that make up IoT devices, most of which are consumer items that people use every day and are part of their everyday life in some way. As an example of IoT devices, they can be anything from smart devices such as fire alarms, light bulbs, air-conditioner, projectors, refrigerators, ovens, and also for office/organizational use such as sensors-based pulse detector, motion sensor devices, parking lots, etc. They can also include devices such as smart meters and cameras. It has been found that the devices make it easier for clients to access services even from a far distance and in a very efficient manner. There are numerous applications of the Internet of Things, including applications in different industries, such as the building and construction sector, agriculture, the military, home appliances, and personal health care.(Mosenia and Jha, 2017)

The Internet of Things is also applicable to other fields, such as manufacturing, medical, retail, transportation, and more. A few challenges have arisen due to the emergence of Internet of Things (IoT) devices, which include maintaining these devices, managing data collected through these devices, storing collected data, exchanging data among these devices, maintaining the security of these devices, maintaining confidentiality, and many others. In recent years, many different types of research by many researchers have been conducted in order to deal with the challenges posed by these limitations or shortcomings. There is a considerable amount of research that has been conducted on architectures, applications security, privacy, and protocols. Many researchers, however, are concerned about the security, privacy, and satisfaction of consumers, which has been a leading cause of concern for them. As IoT devices have become more popular, the number of threats and attacks against these devices has increased despite the introduction of new technologies like cloud computing, SDN and SD-WAN. IoT-based devices also work over the internet and are based on the same protocol for information exchange as other internet-based services that operate using the internet. Through the internet, devices are linked by their private IP addresses, which serve as a common platform for exchanging messages. This research was motivated by the fact that devices that are part of the Internet of Things network are encountering difficulties when it comes to the routing table entries as the number of IoT devices increases around the world. As IoT-based machines grow exponentially, routing becomes more difficult, leading to inefficient routing leading to delays in their response. In order to compensate for the lack of improvement in routing, it is possible to size up the routing table, but deploying these solutions can be more expensive.(Ammar et al., 2018)

1.1. SOFTWARE-DEFINED NETWORK AND IOT

The software-defined network distributes the control and data flow of the machines, by detaching the control and data forwarding modules to introduce a new entity called a network controller to monitor the traffic that is being routed. The forward plane is located on the bottom of the stack that deals with the hardware interaction of the network side for the

device. The device itself is not capable of filling the routing and forwarding tables present on the forward plane, the network logic relocates the controller layer. The controller layer provides resource leveling operation, it also provides network topology and state information in order to validate the decisions additionally the controller also oversees the north and southbound communication via APIs and provides accessibility to the forwarding tables. SDN is a networking framework that is used for improving flexibility, scalability, and security of IoT devices, a number of researches have been developed that have supported SDN for the networking model for IoT devices, however, the security of this framework needs to be addressed as it is a major concern for this approach (Patnaik et al., 2020)

1.2. SECURITY ISSUES IN SDN

The biggest advantage of an SDN architecture is its capability to program a network i.e., Network Programming. Using network programming we can achieve access to physical devices through the controller. The programming feature would help us in accessing the potential uses of the input points and logic installed on the IoT machine and it further helps in the innovation of the network by improving the software versions of the product.

Security threats are one of the key challenges in these programmable devices, the threats in the SDN network can be critical as exposure of a single device may result in an intrusion attack that would affect all of the devices connected in the network. OpenFlow is a tool that is used to detect the vulnerabilities present in IoT devices, from the analysis it was obtained that the number of threats present to a device was relatively higher than non-programmable devices indicating the need for a security infrastructure for these devices. The flow of rules created by the developer for the devices can be different which may result in conflicts that might further reduce the device security. We are proposing a model to make IoT devices more secure and faster by improving their response time and the overall security of the infrastructure. Routing is difficult to scale due to the large number of routing tables and schema. Therefore, logarithmic network scaling is needed to improve routing, since protocols are challenging to replace. For the existing system to be automated, machine learning techniques would have to be applied.(Kaspersky, 2020)

2 Related Work

Several research papers have been proposed by authors that involve machine learning and deep learning algorithms using supervised and unsupervised learning to detect the attack vectors in IoT devices. The main advantage offered by AI and Machine learning algorithms is that they can observe and process millions of communication logs on a daily basis that might happen within a large network. In this section, we would be discussing the different approaches followed by a number of researchers in order to detect anomalies in dataset that is developed for testing the IoT-SDN framework that researchers may propose in their research. Some of the research that has followed machine learning and deep learning methods to secure their model IoT network are given below (Lima Filho et al., 2019)

A denial-of-service detection system was proposed by (Bhunia and Gurusamy, 2017) based on SDN for IoT devices, their research claimed that they were capable of accurately stopping 98 percent of the DoS requests that were directed towards their test PC. A statistical solution was proposed that would approximate the results that were obtained from the analysis of the network traffic against a test packet that was benchmarked as malicious. If a certain

approximate was achieved that the new packet was similar to the benchmarked packet, the packet stream was classified as malicious, and the request was rejected.

Research that focused more on the metadata encryption on the IoT devices was proposed by (Chakrabarty and Engels, 2016) in which they proposed an SDN-based architecture for the secure payload and metadata encryption framework called blackSDN. In this method, all the components and information that was routed in and out of the IoT device were encrypted so that network sniffing was not possible. the research and the framework that was developed were effective, but they had a downside that increased the complexity of time. The source and the header information of the data packet were also encrypted which increased the computing power per packet and the complexity of time as well.

In this research,(Fiore et al., 2013) proposed a methodology that would use an unsupervised learning algorithm for network anomaly detection. A restricted Boltzmann machine was developed to configure an intrusion detection system. the results that were obtained were not promising as the number of false positives was high during the test data whereas the dataset on which their model was trained was offered by NSL-KDD for knowledge data mining showed promising results. The` following study was an independent article published in 2018 by (Dawoud et al., 2018), in which he published the calculations and methodology for the development of an intrusion prevention and detection system that would follow the SDN controller architecture for intrusion detection. He also proposed a restricted Boltzmann machine for the configuration of an intrusion prevention system. The Boltzmann machine that he proposed was to be deployed on IoT devices that were used in large smart city applications, A three-tier architecture was developed like an SDN approach in which detection of malicious traffic by the help of a deep learning model. To identify the potential malware on an android based IoT device, (Alam and Vuong, 2013) used an ensembled learning model in conjunction with the random forest classification, in his approach he used android apps that were affected by malware in order to test the classifier's accuracy. During the testing phase, the author looked at the criteria of the malware using the random forest classification to identify the malicious data sources. From the initial testing, it was revealed that the classifier could achieve more than 99 percent accuracy, and the misclassification and false positive rates were much lesser than other machine learning algorithms. Using their experimental settings, it was found that more than 16 was the ideal depth of the tree as the number of false positives was minimized and greater identification was achieved in their experimental settings.(Zhou and Yu, 2018) proposed a malware detection method that detected the spread of malware from one IoT device to the system of connected network devices. The authors proposed a cloud-based model that would run an SVM classifier in its backend for malware detection and transmission.

After extensively studying the various machine learning approaches, (Ozay et al., 2015) explored the working of several algorithms such as unsupervised learning, feature space fusion, and supervised learning algorithms for attack detection. Following conducting their research, the authors separated the networks into smaller and bigger groups and concluded that SVM performed well in terms of attack detection accuracy, whereas k-NN performed better in large networks. The authors emphasized on machine learning techniques for detecting the attacks since their computational difficulty is often lower than that of set learning algorithms. But each of these algorithms performed consistently. A method was proposed by (Ham et al., 2014) in which he highlighted the use of multiple models separately and in conjunction to evaluate the accuracy of detection. The detection models which were used are the Deep belief network and Support vector machine, when both models were used

separately, they returned a mean accuracy of 88 percent and 90 percent. But when these models were used in a hybrid approach, 93 percent accuracy was achieved. The deep belief network functioned as a dimensionality reduction matrix for the SVM classifier in this method. A comparative study was conducted between the anomaly detection models that are used, the study compared the five different decision-making algorithms and two different decision boundaries against the datasets that contained sequential, static, and spatial data as an IoT device can generate different types of data, it was important to test all possible data types against the machine learning algorithms. From the research, it was identified that SVM returned the highest detection accuracy, which was followed closely by the principal component analysis theorem (Servida and Casey, 2019).For IoT network, (Rathore and Park, 2018) suggested a semi-supervised machine learning approach. NSL-KDD was used for the experimental examination of their work. According to the research, the attack may be effectively identified by utilizing the Fuzzy C Means technique, which is a machine algorithm also referred as ESFCM that works by establishing the infrastructure. The capability of ESFCM to handle labeled data sets it apart, increasing the detection accuracy of potential attacks.

One of the primary requirements to improve the security of IoT devices is the authentication of the device and enabling access control protocols within the network. As the network uses a system of distributed computing sources to manage and communicate data, therefore it is important that the user access control of the device metadata is applied. To enable the authentication system for the IoT devices (Liu et al., 2012) proposed a method in their research that involved an authentication system based on machine learning technique. To examine the strength of the physical security, Liu used a game theory to identify the IoT users from malicious users, the technique increases in utility by using a zero-sum game so that the frequency of attack can be measured to attain a state of equilibrium. A projected layer to be utilized for authentication by machine learning was presented by(Das et al., 2018) to achieve authentication in IoT devices. The authors employed physical security in this technique, such as evaluating the signal intensity. Furthermore, Zhang employed a machine learning strategy based on a game-theoretic approach to distinguish criminals from benign (Normal) IoT users. The Zero-Sum was used to improve the strategy which utilized the channel frequency response to raise the attack frequency which further build Nash Equilibrium. Using an authentication method based on ANN-Model in IoT devices network was (Chatterjee et al., 2018) solution to authentication issues in wireless nodes. They have employed the PUF to do this, that performs by examining the transmitters physical attributes and eventually eliminating any transmitter in the network found to be duplicate. The Author’s study also included a machine learning technique " in-situ” for grouping all transmitters. This method was implemented around the receiver, and transmitter classification was done at the sensitivity. After analyzing the accuracy, the authors concluded that the detection error from a specific transmitter had a minor overhead; nevertheless, the specific receiver required two neural networks, which typically add roughly 3% to 5%.

Research Gap

All the research papers cited in this work were very helpful in advancing our research. Though, the papers had achieved high accuracy and have used latest technologies, there were still some grey areas, and our proposed model will try to cover those grey area. The shortcomings of some important papers have been discussed and also how the proposed model will mitigate these issues.

Research Paper	Approach	Dataset	Algorithm	Shortcomings	Proposed Model upsides.
(Diro and Chilamkurti, 2018)	Paper uses Deep Learning for DDoS attack detection in IoT	NSL-KDD	Customised Deep Learning Model	Dataset is based on KDD'99 (Outdated). Dataset not IoT focused but model is for IoT network	Dataset is IoT based and has been released recently. Model trained with DDoS traffic only to reduce False negative.
(Saharkhizan et al., 2020)	Detecting DDoS attack using LSTM	Modbus-TCP (Own Dataset)	LSTM	The model complexity is high. Dataset not IoT focused.	Model is very clean and simple. Dataset is IoT based and has been released recently. Model trained with DDoS traffic only.
(Rathore and Park, 2018)	Semi-supervised learning model proposed to detect DDOS in IoT network	KDD'99 dataset	ESFCM	Dataset is dated from 1999 and not based on IoT network. Accuracy is not the best among the other papers.	Dataset is IoT based and has been released recently. Accuracy is obtained from recent dataset. Model trained with DDoS traffic only.
(Ravi et al., 2017)	RNN, LSTM and GRU based IDS	KDD'99	RNN, LSTM and GRU	Dataset is dated from 1999. Accuracy is not the best among the other papers.	Dataset is IoT based and has been released recently. Accuracy is obtained from recent dataset. Model trained with DDoS traffic only.
(Bhunia and Gurusamy, 2017)	Attack detection in SDN-IoT environment using SVM	Mininet (Own dataset)	Linear and Non-Linear SVM	Accuracy is not the best among the other papers	Model trained with DDoS traffic only. Model will be using both Machine learning and Deep Learning algorithm.
(Torres et al., 2016)	Recurrent Neural Networks for Botnet detection	MCFP	RNN	Slow computation will affect performance of model. False Positive is significant.	Model will be using both Machine learning and Deep Learning algorithm. Model trained with only DDoS traffic only i.e., botnet.
(Lima Filho et al., 2019)	DDoS Attack detection using Machine Learning in IoT (smart devices)	CIC-DoS, CICIDS2017, CSE-CIC-IDS2018 And Own Dataset	RF, LR, SGD, Dtree	Dataset not IoT focused. Accuracy is not the best among the other papers	Model trained with DDoS traffic only. Model will be using both Machine learning and Deep Learning algorithm.

Table 1 Research Gap

3 Research Methodology

The primary research we did for the development of the proposed model is that we conducted a literature review of related research associated with the same or equally motivation that would be addressing the concern we plan to mitigate. The literature review highlighted the relevant research that had similar methodologies based on deep learning and machine learning for IoT security. These papers gave us a base for our model to build upon and thus a steppingstone for us to further progress of our model.

3.1 Dataset Selection

As the prevention of attacks on IoT networks is the main goal of our methodology. We decided on a dataset containing network traffic generated by IoT devices. In order to improve

the model's accuracy as it learns the types of traffic that pass across an IoT network, an IoT dataset with devices seems more appropriate as a normal dataset generated by a network environment might include or exclude some features that could be crucial when detecting an attack.

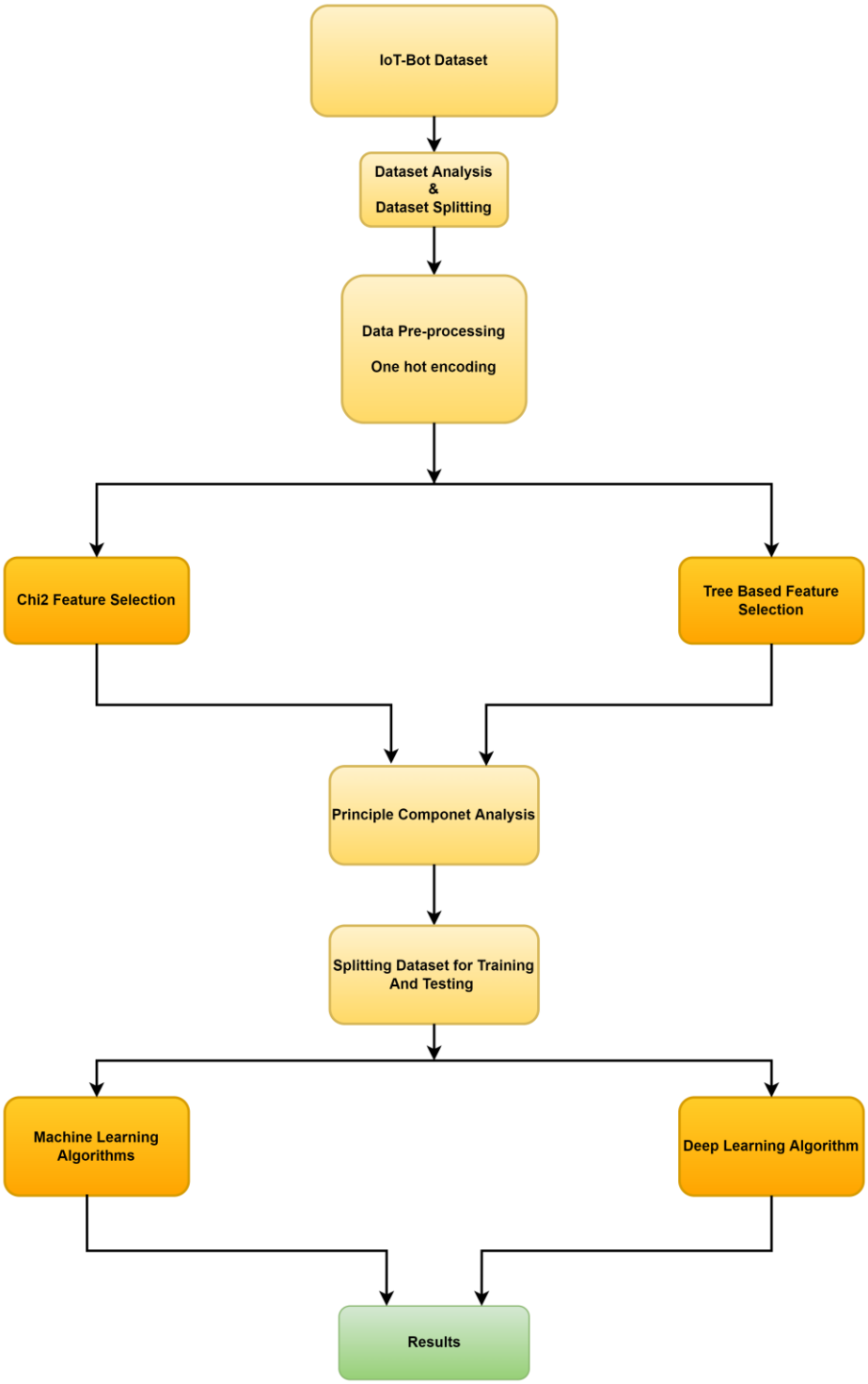


Figure 1 Model's Process Flow diagram

Further study led us to a dataset made available by the University of New South Wales. The University of New South Wales's "Intelligent Security Group" had created a realistic network

environment that included both IoT devices and Botnets. As the IoT devices generated legitimate network traffic, the Botnets sought to disrupt the services by utilizing DoS and DDoS attacks. The archives maintained by the university include several supporting files, such as PCAP files and argus, which assisted in the development of the dataset, hence proving the authenticity of the dataset hosted by them.("The Bot-IoT Dataset | UNSW Research," n.d.) ("CloudStor - CloudStor is powered by AARNet," n.d.)

3.2 Dataset Extraction and Loading

The model being build will be built or planned in a way where the model will only be trained by the Malicious Traffic i.e., DoS and DDoS traffic and will be tested with a dataset comprising of both benign and DDoS traffic. The reason behind the concept of training the model with two different dataset is to make the model more robust. Exposing the model to Normal traffic while training it might help in accuracy with the same dataset but when the model is exposed to a new set of traffic which would be an ideal case of an attack detection system the model might fail to detect an attack because the attack might show some similarity to a certain traffic in the training model and consider it to be normal. Though this might be same if we train with only DDoS traffic that it detects a normal traffic to be a DDoS traffic, but that prediction or assumption is less likely to harm a network and will only create minor inconvenience while the latter is not true. The numerous paper we reviewed choose to train and test the model with the same dataset and thus we choose a different approach to train and test our model.

To achieve the same, we went through the dataset provided by the university and was deemed suitable by them for training and testing. We found few of the dataset with only malicious traffic and one of them with both normal as well as malicious traffic. We choose to then create two separate dataset and treat them independently while one will be used to train the model the other would be used to check the authenticity and accuracy of the model.

3.3 Dataset pre-processing and labelling

The datasets went through a lot of pre-processing before being considered for training and testing. The first step was to replace any blank spaces in the dataset. The datasets were also checked for any null and infinity values which was then replaced by 0 as the values hindered the processing of dataset. Both the datasets were found to be having String values instead of Integers which were becoming a hindrance. The String values were then converted to integers and float. The model also converted the datasets from 64-bit data type to 32-bit reducing the memory usage. Later it was found that both source and destination address were categorical and thus were difficult for our model to interpret and the model proceeds converting them from categorical in nature to numeric.

The traffic of the dataset was already labelled but to make the data more presentable and the model to be more precise we labelled all the DoS and DDoS traffic as DDoS traffic and rest of the traffic to be "Normal" traffic. Using the label encoder these were labelled as 0 and 1 as these nominal variables will cause hindrance hence converting them into integer values.

3.4 Feature Selection

The dataset has total of 46 columns out of which 44 are independent variables (Excluding Traffic labels). These 44 variables or features if used in the model will cause uneven results

as there is a high chance of many of these features to be redundant in nature. Since individually selecting these features would fail the motive of the model as well as there might be few features which we feel redundant, but the model might find them useful.

To achieve the same the model employs some feature selection techniques to reduce the feature we were moving forward for our model. The model includes feature selection techniques Chi2 and Tree-based feature selection. These feature selection techniques were set to choose only 10 of the features they deemed best.
The model had then two different feature techniques with each technique shortlisting ten features. The model then combined all these features and dropped any repetitive feature selected by these three techniques combined. Leaving us with total of 19 features.

Feature	**Description**
pkSeqID	Sequence Number
Stime	Start Time
Flgs	Flow Flags
flgs_number	Flags represented by their numerical value
Proto	Protocol type
proto_number	Protocol represented by their numerical value
Saddr	IP Address: Source
Sport	Port number: Source
Daddr	IP Address: Destination
Dport	Port number: Destination
Pkts	Packet count
Bytes	Bytes Count
State	Transaction state
state_number	State represented by their numerical value
Ltime	End Time
Seq	Sequence Number: Argus
Dur	Duration
Mean	Record’s average duration
Stddev	Record’s Standard Deviation
Sum	Record’s Total Duration
Min	Record’s Minimum duration
Max	Record’s Maximum duration
Spkts	Packet count: Source
Dpkts	Packet count: Destination
Sbytes	Byte count: Source

Dbytes	Byte count: Destination
Rate	Total packets per second in transaction
Srate	Source-to-destination packets per second
Drate	Destination-to-source packets per second
Category	Traffic category
Subcategory	Traffic subcategory

Table 1 Few key features of the Iot dataset

Though these features are way less from 46 originally but there was no confirmation of these features to be still redundant proof so the model proceeds with using PCA known as Principal Component Analysis. With the result it was found post 10 component the dataset would reach variance saturation. The dataset now had 10 components.

3.5 Deep Learning and Machine Learning predictions.

As the datasets were found fit to proceed with training. The datasets were then used for training and testing with different Machine learning and Deep Learning techniques. We proceeded with NB Gaussian, Decision Tree algorithm, Random Forest and CNN. These algorithms would help to determine the best algorithm to detect a DDoS attack on a model trained by just DDoS traffic.

4 Design Specification

The experiment model was performed on a custom desktop with Windows 10 which had Python and visual studio installed.

Desktop Specification:

- Performance-oriented CPU: i9 9900k, stable overclock at 5.1ghz liquid metal
- 32 GB DDR4 ram.
- External GPU AMD Radeon 5700xt with 8GB RAM.
- 2TB SSD storage

To perform the data cleansing operations, we used python's data science library Pandas and NumPy and for the analysis of data, we used the Keras framework. To activate the GPU for our research we used TensorFlow which is also a framework available in Python.

Software used:

- 64-bit Windows 10 operating system is running on the desktop.
- Python 3.10.5 64-Bit
- Microsoft Visual Studio Code v1.60.2

Model's Graphical Design:

There are three key components which make up the SDN IoT network architecture. The first layer is comprised of network applications and services that require protection, while the middle layer is comprised of the SDN controller, which filters malicious traffic using Machine learning or Deep Learning Algorithms.

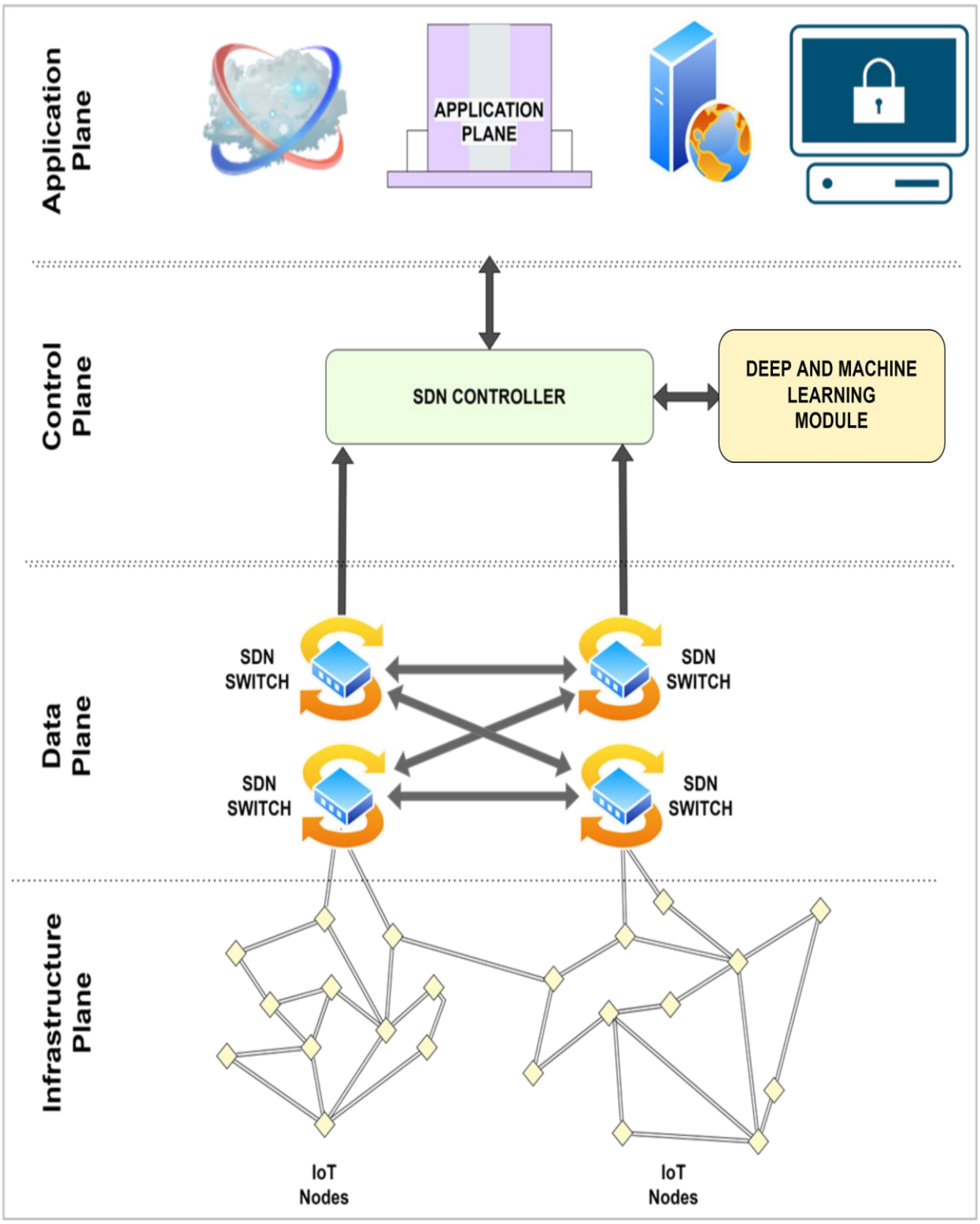


Figure 2 Graphical representation of Model's Architecture

The Module can fetch these results only via SDN controller i.e., the traffic data SDN switches collects from numerous IoT nodes connected in the "Infrastructure Plane". These nodes are a web of IoT devices which are collecting data as well must present smart results which are fetched by the data they send.

5 Implementation

The model used tools and libraries to process the dataset, perform feature selection, and train and test the model. We would be discussing these tools and libraries in this section and what role did they play in helping the model to progress.

Dataset Preparation: The Bot-Iot dataset was already set by the college for training and testing. On further studying the dataset it was found the that 3 of the CSV out of 4 were only having DDoS traffic and the last one had a combination of all traffic. As the goal was to train the model with just DDoS traffic. We choose to go with two datasets i.e., one with just DDoS traffic whereas for testing we used the CSV with all the traffic's combined.

It was found that the DDoS dataset we selected had forty-six columns and 3000000 rows and the testing dataset had 668522 rows and the same 46 columns. All the NaN values were changed to 0 and, Infinity and blank spaces were dropped for the model's smooth processing. All the 64-bit data type were converted to 32-bit to save the memory consumption of the model. It was later found that the model had categorical features which were converted to numerical values. Both the datasets had labelling we resorted to change the DoS and DDoS to single variable i.e., DDoS and whereas the rest of the traffic had nothing to do with DDoS and were converted to Normal. The labels of the dataset were dropped and few features which were redundant or were of no use.

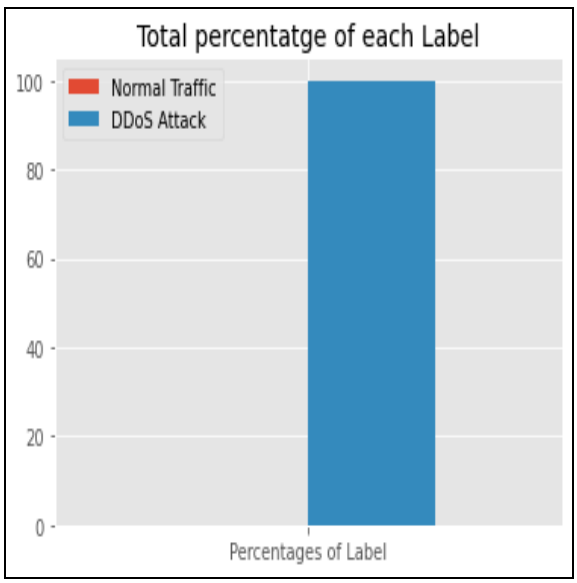


Figure 3 DDoS Traffic

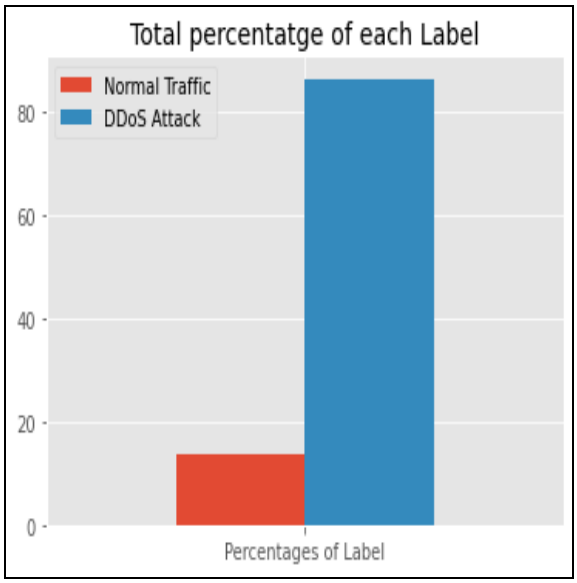


Figure 4 DDoS and Normal Traffic

Feature Selection: Since the dataset still had 41 features. This could mean that there will be many features of these datasets which wouldn't be of need. So, model proceeded by choosing two feature selection algorithms i.e., Chi2(Chi-Square) and Tree-based feature selection.

These features together selected total of 20 features, which were combined, and any repetition of features were dropped. Both the datasets went through the same process. Post which it was found even 20 features were a lot and manually selecting these features could lead to human error.

Figure 5 Value distribution of features.

The model then chooses to proceed with Principal Component Analysis, so any redundant feature in both the dataset would be combined and thus the model will have less components to process.

As the redundant features would not help the model to distinguish between a normal traffic and ddos traffic. Since we don't know the features which would help us identify a ddos attack we can't delete features even if we consider it redundant, as we don't know which feature could help us identify the attack. Here is where PCA comes into picture. PCA doesn't delete these features but reduce the redundancy and noise in the dataset. This helps in highlighting the important characteristics of a dataset. The results were satisfactory as the 90% variance was achieved in the first 5 components for both the datasets

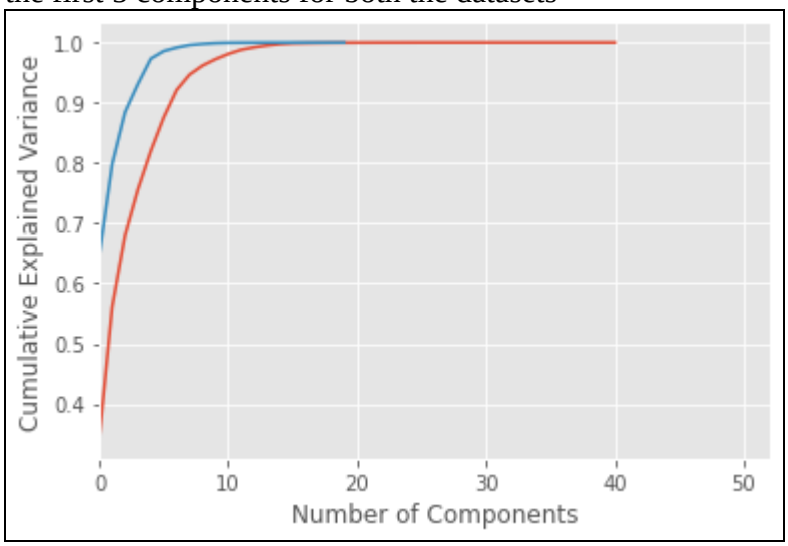
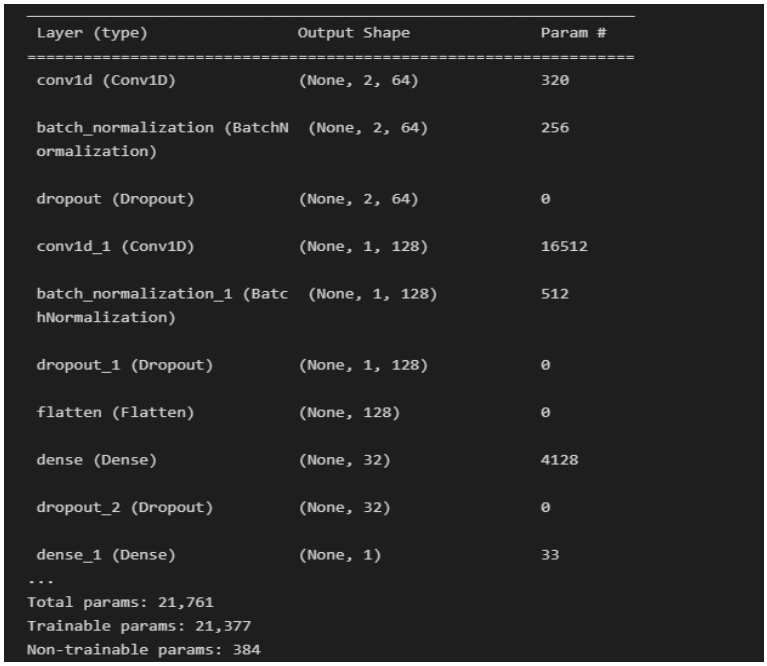


Figure 6 PCA result for both the dataset.

As the redundant features would not help the model to distinguish between a normal traffic and ddos traffic. Since we don't know the features which would help us identify a ddos attack we can't delete features even if we consider it redundant, as we don't know which feature could help us identify the attack. Here is where PCA comes into picture. PCA doesn't delete these features but reduce the redundancy and noise in the dataset. This helps in highlighting the important characteristics of a dataset. The results were satisfactory as the 90% variance was achieved in the first 5 components for both the datasets

Dataset training and testing: As the model's aim is to train it with only DDoS traffic and then test with a combination of both DDoS and Normal traffic, the model then proceeds by classifying the dataset with variables as X_train, y_train for DDoS dataset and X_test and y_test for the combined traffic dataset. Here the X stands for the dataset without label and y is the label which defined the traffic type.

Machine Learning and Deep Learning Model: The model proceeded with checking the accuracy for few of the machine learning algorithms. Random Forest, Gaussian NB and Decision tree were the machine learning algorithm which were being tested with the two datasets. These algorithms were then assessed with the accuracy, precision and recall obtained by metrics derived from the prediction model made using the test dataset.



Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 2, 64)	320
batch_normalization (BatchNormalization)	(None, 2, 64)	256
dropout (Dropout)	(None, 2, 64)	0
conv1d_1 (Conv1D)	(None, 1, 128)	16512
batch_normalization_1 (BatchNormalization)	(None, 1, 128)	512
dropout_1 (Dropout)	(None, 1, 128)	0
flatten (Flatten)	(None, 128)	0
dense (Dense)	(None, 32)	4128
dropout_2 (Dropout)	(None, 32)	0
dense_1 (Dense)	(None, 1)	33

...
Total params: 21,761
Trainable params: 21,377
Non-trainable params: 384

Figure 7 Layers of CNN

Further the model went with Deep learning Algorithms. CNN (Convolutional Neural Network) is used to reduce the information parameters using the feature extraction characteristic in each layer. Decreasing the association of features between the layer would improve the training time and would help in extending the use case of the network.

6 Evaluation

The intent of the research is to determine the result of evaluating the dataset with Random Forest, Decision Tree, NB Gaussian, and the CNN algorithm. With the aid of the dataset we utilized, we could evaluate each machine learning and deep learning approach individually on the data. The accuracy, false positives, true negatives, and positives would all be assessed in order to determine the optimal technique for safeguarding IoT devices.

- **False Positive (FP)** - Denotes the model's inaccuracy when predicting a positive attack when the observed attack is benign.
- **False Negatives (FN)** - Are specifically defined as malicious attacks that the model mistakenly forecasted as benign.
- **True Positives (TP)** - Estimates the frequency of accurately recognized real attacks by the model.
- **True Negative (TN)** - This metric measures the accuracy with which benign traffics are recognized.

Accuracy: To assess the accuracy of the proposed model, divide the total amount of correct predictions by the grand total of predictions. These performance metrics are regarded as being the simplest. These measurements, meanwhile, are not the only factor when assessing the effectiveness of any model as independent metrics.

Accuracy = TP + TN / TP + TN + FN + FP.

Precision: These parameters allow us to assess how well our model predicts successful outcomes. In order to achieve this, we divide the total number of True positives by the total number of positive predictions the model produced.

The following formula might be used to gauge how precise the model is

Precision = TP / TP + FP.

Recall: By dividing the total number of True positives discovered by the model by the actual number of positives present, these metrics helps in providing us metrics that measure the ability of the model to detect positive samples.

Recall= TP / TP + FN.

6.1 Case Study 1: Random Forest

The training dataset i.e., DDoS traffic dataset was used to train the model using Random Forest Algorithm and then the dataset with both DDoS traffic as well as Normal traffic was used to test the Random Forest Algorithm model. The projected result wasn't the best but the fact that there were no false negative found was a partial win for the model. Model gave back an accuracy of 82.292%. Precision and Recall were not possible considering the value of metrics.

```
Accuracy score= 0.86292


confusion matrix


Count of True Positive =  0
Count of False Positive =  91638
Count of False Negative =  0
Count of True Negative =  576884


Precision, Recall, F1


              precision    recall  f1-score   support

      Normal       0.00      0.00      0.00     91638
        DDoS       0.86      1.00      0.93    576884

    accuracy                           0.86    668522
   macro avg       0.43      0.50      0.46    668522
weighted avg       0.74      0.86      0.80    668522
```

Figure 8 Metrics

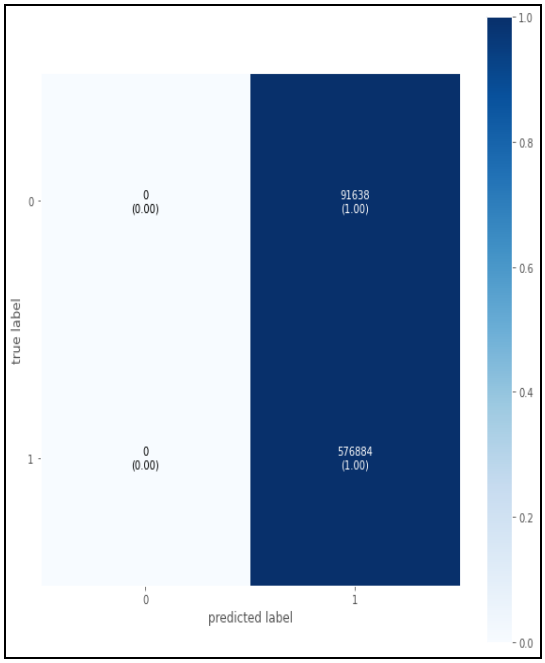


Figure 9 Confusion matrix

6.2 Case Study 2: Gaussian NB

The training dataset i.e., DDoS traffic dataset was used to train the model using Gaussian NB and then the dataset with both DDoS traffic as well as Normal traffic was used to test the Gaussian NB Algorithm model. The projected result wasn't the best but the fact that there were no false negative found was a partial win for the model. Model gave back an accuracy of 86.304%. Precision and Recall were not possible considering the value of metrics.

```
Accuracy score= 0.86304


confusion matrix


Count of True Positive =  0
Count of False Positive =  91559
Count of False Negative =  0
Count of True Negative =  576963


Precision, Recall, F1


              precision    recall  f1-score   support

      Normal       0.00      0.00      0.00     91559
        DDoS       0.86      1.00      0.93    576963

    accuracy                           0.86    668522
   macro avg       0.43      0.50      0.46    668522
weighted avg       0.74      0.86      0.80    668522
```

Figure 10 Metrics

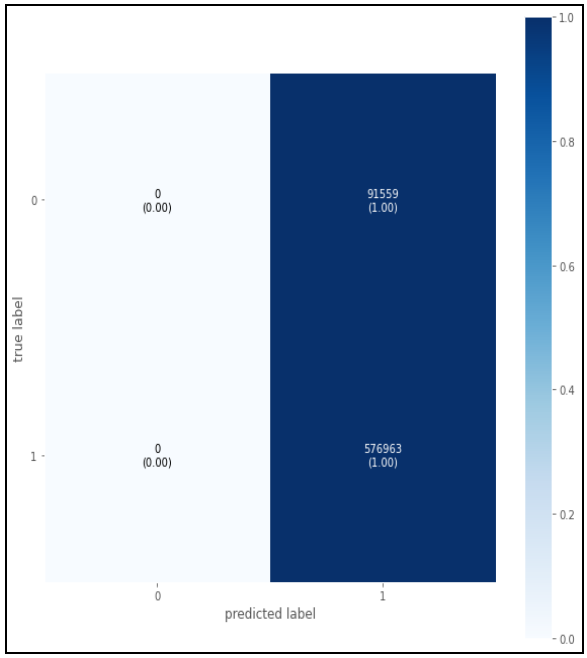


Figure 11 Confusion Matrix

6.3 Case Study 3: Decision Tree

The training dataset i.e., DDoS traffic dataset was used to train the model using Decision Tree and then the dataset with both DDoS traffic as well as Normal traffic was used to test the Decision Tree model. The projected result wasn't the best but the fact that there were no false negative found was a partial win for the model. Model gave back an accuracy of 86.359%. Precision and Recall were not possible considering the value of metrics.

```
Accuracy score= 0.86359


confusion matrix


Count of True Positive =  0
Count of False Positive =  91193
Count of False Negative =  0
Count of True Negative =  577329


Precision, Recall, F1


              precision    recall  f1-score   support

      Normal       0.00      0.00      0.00     91193
        DDoS       0.86      1.00      0.93    577329

    accuracy                           0.86    668522
   macro avg       0.43      0.50      0.46    668522
weighted avg       0.75      0.86      0.80    668522
```

Figure 12 Metrics

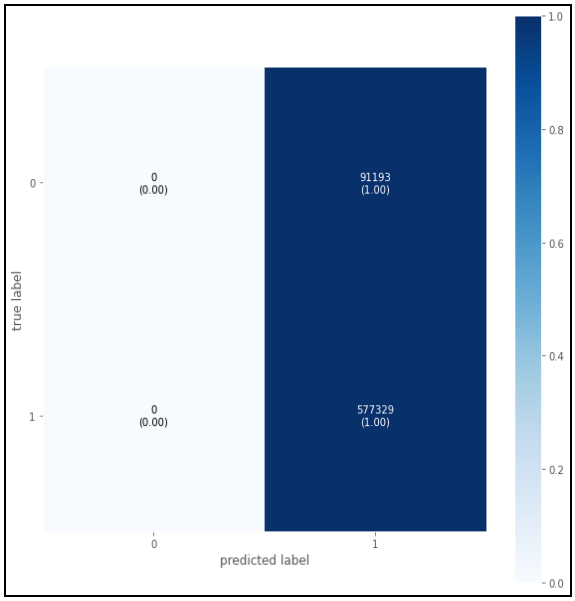


Figure 13 Confusion Matrix

6.4 Case Study 4: Convolutional Neural Network

The training dataset i.e., DDoS traffic dataset was used to train the model using Convolutional neural network and then the dataset with both DDoS traffic as well as Normal traffic was used to test the Convolutional neural network model. The projected result wasn't the best but the fact that there were no false negative found was a partial win for the model. Model gave back an accuracy of 86.304%. Precision and Recall were not possible considering the value of metrics.

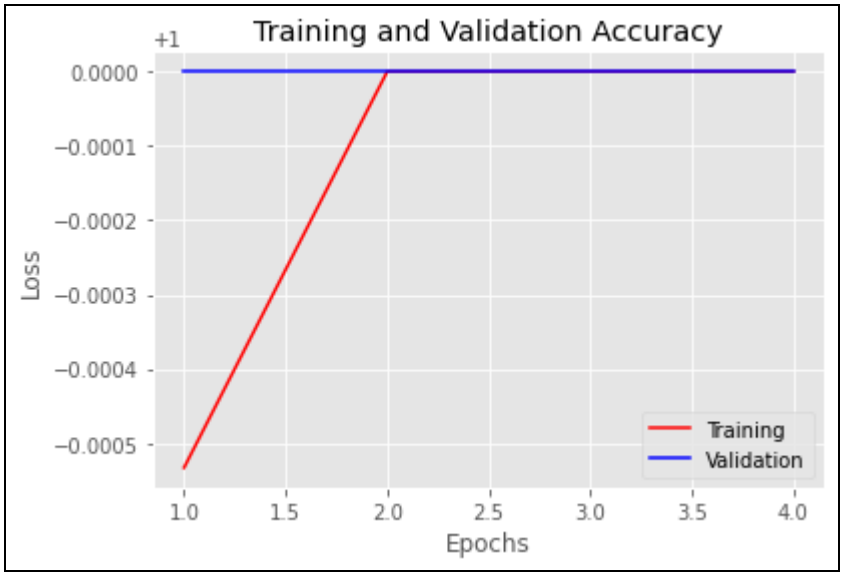


Figure 14 CNN Training and Validation

```
20892/20892 [==============================] - 12s 564us/step
Test set
   Accuracy: 0.86304


confusion matrix


Count of True Positive =  0
Count of False Positive =  91559
Count of False Negative =  0
Count of True Negative =  576963


Precision, Recall, F1


              precision    recall  f1-score   support

      Normal       0.00      0.00      0.00     91559
        DDoS       0.86      1.00      0.93    576963

    accuracy                           0.86    668522
   macro avg       0.43      0.50      0.46    668522
```

Figure 15 Metrics

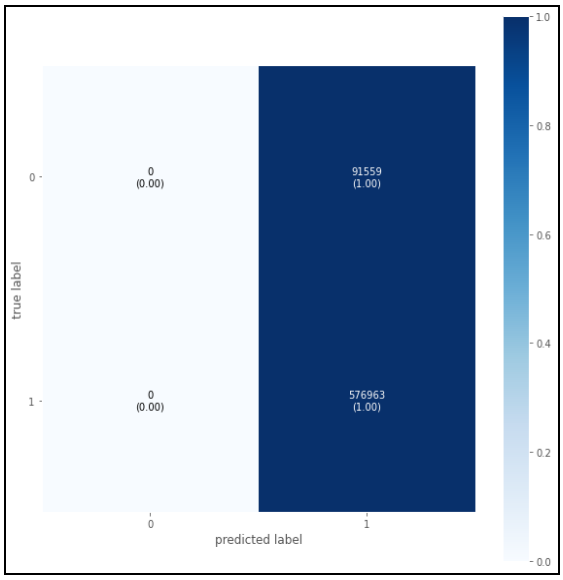


Figure 16 Confusion Matrix

6.5 Discussion

Experiments were performed in this research to demonstrate that the model may be utilized to decrease or have negligible False Negative while still maintaining detection accuracy for DDoS in SDN-IoT network, which is essential for Cybersecurity research. We can observe through an analysis of all algorithms that the suggested model can have a high level of

security against DDoS attack as the chance of getting not detecting is low as the model is only trained with DDoS traffic so though there would be False positive i.e., detecting a normal traffic as DDoS. This prediction might cause hindrance but getting a False Positive would be better than getting a False Negative.

False Positive could be present for many reasons in a network traffic when detecting for DDoS attack.

- When there is a backup is going on, the sudden increase in bandwidth usage could trigger a false DDoS warning.
- When there is a loop in a switch device, the loop creates a sudden surge in traffic when working at layer 2 devices. The surge could be falsely labelled as DDoS attack.(Support, 2019)
- When an organisation or group of people become active at business hours i.e., When people try to access an organization server at the same time to perform heavy data work it could be falsely labelled as DDoS attack.
- When a network utilizes redundant ISP lines for different type of application and if one of them goes down, the load on shared network increase as the traffic is directed from a single ISP line.(Tech, 2019)

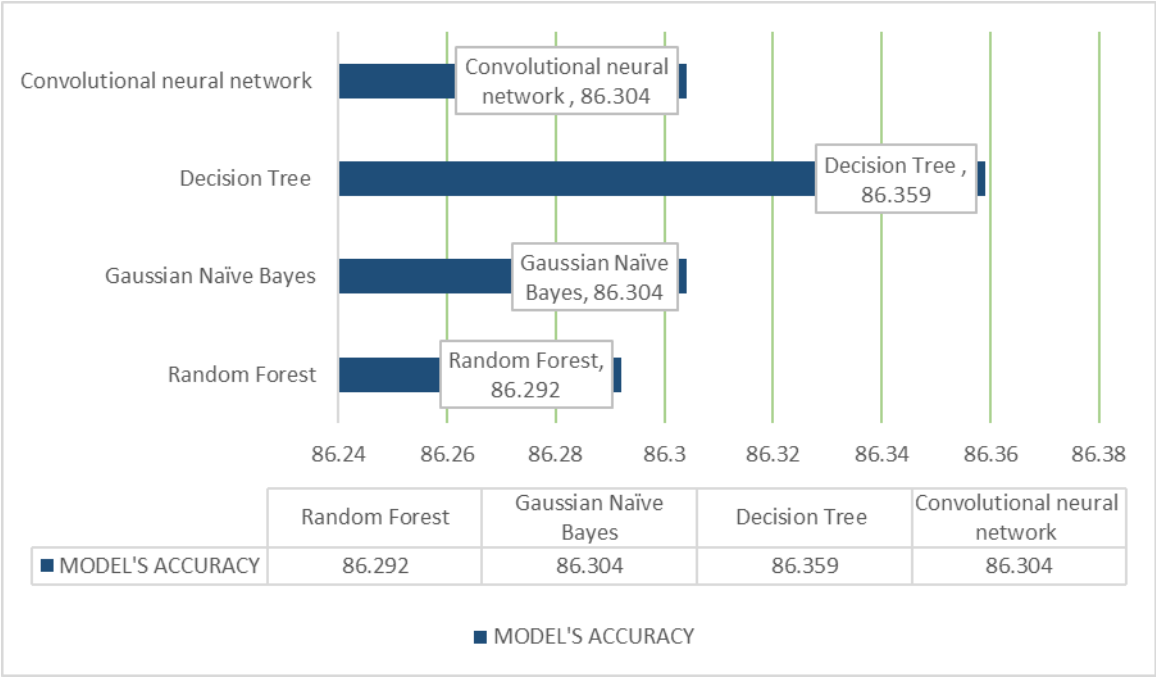


Figure 17 Accuracy of Machine learning and Deep Learning models

7 Conclusion and Future Work

The paper focuses on making an IoT network secure using SDN-coupled with machine learning and deep learning. With selection of 20 different features of the IoT network traffic from a total of 46 features of the original dataset the model then proceeds with converting them into 5 components using PCA. These components will help the model to work more proficiently as it will have important features that would help the model to make precise prediction. The proposed model is trained with only DDoS traffic which makes the model

focused more on detecting DDoS than differentiating between an Attack or normal traffic. This change in model type might have false positives but the chance of false negative is almost negligible, and a false negative is considered more disastrous than false positive making the model more convenient for different applications in different sectors.

For future the proposed model could focus on having a parallel model being run where one of the models would have been trained by DDoS traffic where the other model will be trained by all types of traffic. As the current proposed model was found to have false positive when compared with labelled dataset i.e., test dataset but for live monitoring distinguishing a false positive from true negative will be hard. To overcome the issue the results from both the models will then be clubbed together. This would introduce us with an overlap of attacks which could be treated as confirmed attacks whereas those traffics which weren't detected by the second model but was flagged by the first model or vice-versa would need to be addressed separately creating a 2-dimension results. Post processing the false positive and true negative would be null or negligible as the primary model and secondary model results will help them to be distinguished by an admin monitoring these results. Also post logs of these traffic has been analysed the model increases the accuracy by reviewing the results.

.

References

Alam, M.S., Vuong, S.T., 2013. Random Forest Classification for Detecting Android Malware | IEEE Conference Publication | IEEE Xplore [WWW Document]. URL https://ieeexplore.ieee.org/document/6682136 (accessed 7.29.22).

Ammar, M., Russello, G., Crispo, B., 2018. Internet of Things: A survey on the security of IoT frameworks. J. Inf. Secur. Appl. 38, 8–27. https://doi.org/10.1016/j.jisa.2017.11.002

Bhunia, S.S., Gurusamy, M., 2017. Dynamic attack detection and mitigation in IoT using SDN, in: 2017 27th International Telecommunication Networks and Applications Conference (ITNAC). pp. 1–6. https://doi.org/10.1109/ATNAC.2017.8215418

Chakrabarty, S., Engels, D.W., 2016. A secure IoT architecture for Smart Cities, in: 2016 13th IEEE Annual Consumer Communications & Networking Conference (CCNC). pp. 812–813. https://doi.org/10.1109/CCNC.2016.7444889

Chatterjee, B., Das, D., Maity, S., Sen, S., 2018. RF-PUF: Enhancing IoT Security through Authentication of Wireless Nodes using In-situ Machine Learning. https://doi.org/10.48550/arXiv.1805.01374

CloudStor - CloudStor is powered by AARNet [WWW Document], n.d. . CloudStor. URL https://cloudstor.aarnet.edu.au/plus/s/umT99TnxvbpkkoE (accessed 8.14.22).

Das, R., Gadre, A., Zhang, S., Kumar, S., Moura, J.M.F., 2018. A Deep Learning Approach to IoT Authentication, in: 2018 IEEE International Conference on Communications (ICC). pp. 1–6. https://doi.org/10.1109/ICC.2018.8422832

Dawoud, A., Shahristani, S., Raun, C., 2018. Deep learning and software-defined networks: Towards secure IoT architecture. Internet Things 3–4, 82–89. https://doi.org/10.1016/j.iot.2018.09.003

Diro, A.A., Chilamkurti, N., 2018. Distributed attack detection scheme using deep learning approach for Internet of Things. Future Gener. Comput. Syst. 82, 761–768. https://doi.org/10.1016/j.future.2017.08.043

Fiore, U., Palmieri, F., Castiglione, A., De Santis, A., 2013. Network anomaly detection with the restricted Boltzmann machine. Neurocomputing 122, 13–23. https://doi.org/10.1016/j.neucom.2012.11.050

Ham, H.-S., Kim, H.-H., Kim, M.-S., Choi, M.-J., 2014. Linear SVM-Based Android Malware Detection for Reliable IoT Services. J. Appl. Math. 2014, e594501. https://doi.org/10.1155/2014/594501

Kaspersky, L., 2020. DENIAL OF SERVICE: HOW BUSINESSES EVALUATE THE THREAT OF DDOS ATTACKS.

Lima Filho, F.S. de, Silveira, F.A.F., de Medeiros Brito Junior, A., Vargas-Solar, G., Silveira, L.F., 2019. Smart Detection: An Online Approach for DoS/DDoS Attack Detection Using Machine Learning. Secur. Commun. Netw. 2019, e1574749. https://doi.org/10.1155/2019/1574749

Liu, J., Xiao, Y., Chen, C., 2012. Authentication and Access Control in the Internet of Things. pp. 588–592. https://doi.org/10.1109/ICDCSW.2012.23

Mosenia, A., Jha, N.K., 2017. A Comprehensive Study of Security of Internet-of-Things. IEEE Trans. Emerg. Top. Comput. 5, 586–602. https://doi.org/10.1109/TETC.2016.2606384

Ozay, M., Esnaola, I., Yarman Vural, F., Kulkarni, S., Poor, H.V., 2015. Machine Learning Methods for Attack Detection in the Smart Grid. IEEE Trans. Neural Netw. Learn. Syst. 27. https://doi.org/10.1109/TNNLS.2015.2404803

Patnaik, Ranjit, Raju, S., Sivakrishna, K., Patnaik, R, 2020. Internet of Things-Based Security Model and Solutions for Educational Systems. pp. 171–205. https://doi.org/10.1007/978-981-15-7965-3_11

Rathore, S., Park, J.H., 2018. Semi-supervised learning based distributed attack detection framework for IoT. Appl. Soft Comput. 72, 79–89. https://doi.org/10.1016/j.asoc.2018.05.049

Ravi, V., Kp, S., Poornachandran, P., 2017. Evaluation of Recurrent Neural Network and its Variants for Intrusion Detection System (IDS). Int. J. Inf. Syst. Model. Des. 8, 43–63. https://doi.org/10.4018/IJISMD.2017070103

Saharkhizan, M., Azmoodeh, A., Dehghantanha, A., Choo, K.-K.R., Parizi, R., 2020. An Ensemble of Deep Recurrent Neural Networks for Detecting IoT Cyber Attacks Using Network Traffic. IEEE Internet Things J. PP, 1–1. https://doi.org/10.1109/JIOT.2020.2996425

Servida, F., Casey, E., 2019. IoT forensic challenges and opportunities for digital traces. Digit. Investig. 28, S22–S29. https://doi.org/10.1016/j.diin.2019.01.012

Stellios, I., Kotzanikolaou, P., Psarakis, M., Alcaraz, C., Lopez, J., 2018. A Survey of IoT-Enabled Cyberattacks: Assessing Attack Paths to Critical Infrastructures and Services. IEEE Commun. Surv. Tutor. 20, 3453–3495. https://doi.org/10.1109/COMST.2018.2855563

Support, N., 2019. What is a network loop? [WWW Document]. NETGEAR KB. URL https://kb.netgear.com/000060475/What-is-a-network-loop (accessed 8.9.22).

Tech, A., 2019. 5 Reasons You Might See a Surge in Bandwidth Usage > Access Tech. Access Tech. URL https://www.accesstech.net/2019/06/10/5-reasons-you-might-see-a-surge-in-bandwidth-usage/ (accessed 8.9.22).

The Bot-IoT Dataset | UNSW Research [WWW Document], n.d. URL https://research.unsw.edu.au/projects/bot-iot-dataset (accessed 8.14.22).

Torres, P., Catania, C., Garcia, S., Garino, C.G., 2016. An analysis of Recurrent Neural Networks for Botnet detection behavior, in: 2016 IEEE Biennial Congress of Argentina (ARGENCON). pp. 1–6. https://doi.org/10.1109/ARGENCON.2016.7585247

Zhou, W., Yu, B., 2018. A cloud-assisted malware detection and suppression framework for wireless multimedia system in IoT based on dynamic differential game. China Commun. 15, 209–223. https://doi.org/10.1109/CC.2018.8300282