DDOS ATTACK PREDICTION USING DEEP LEARNING

A PROJECT WORK REPORT

Submitted by

NAVEEN SABARINATH B	**(1901124)**
NEHA O	**(1901126)**
KAVIYASRI B	**(1901163)**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

SRI RAMAKRISHNA ENGINEERING COLLEGE
[Educational Service: SNR Sons Charitable Trust]
[Autonomous Institution, Reaccredited by NAAC with 'A+' Grade]
[Approved by AICTE and Permanently Affiliated to Anna University, Chennai]
[ISO 9001:2015 Certified and All Eligible Programmes Accredited by NBA]
Vattamalaipalayam, N.G.G.O. Colony Post,
COIMBATORE – 641 022
ANNA UNIVERSITY : CHENNAI 600 025

APRIL 2023

ANNA UNIVERSITY : CHENNAI 600 025

BONAFIDE CERTIFICATE

16CS270 –PROJECT WORK

Certified that this Project Work Report "**DDOS ATTACK PREDICTION USING DEEP LEARNING**" is the bonafide work of ", **Naveen Sabarinath B, Neha O, Kaviyasri B**" who carried out the project under my supervision.

SIGNATURE
Mrs.C.Padmavathy
SUPERVISOR
Assistant Professor (Sl.Gr),
Computer Science and Engineering
Sri Ramakrishna Engineering College,
Coimbatore-641022.

SIGNATURE
Dr.Grace Selvarani
HEAD OF THE DEPARTMENT
Professor,
Computer Science and Engineering,
Sri Ramakrishna Engineering College,
Coimbatore-641022.

Submitted for the Project Work Viva-Voce Presentation held on ____________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION

We affirm that the Project work titled **“DDOS ATTACK PREDICTION USING DEEP LEARNING”** being submitted in partial fulfillment for the award of Bachelor of Engineering is the original work carried out by us. It has not formed the part of any other project work submitted for award of any degree or diploma, either in this or any other University.

(Signature of the Candidates)

NAVEEN SABARINATH B	**(1901124)**
NEHA O	**(1901126)**
KAVIYASRI B	**(1901163)**

I certify that the declaration made above by the candidates is true.

(Signature of the guide)

Mrs. C. Padmavathy

Assistant Professor, (Sl.Gr)

Department of CSE

ACKNOWLEDGEMENT

We express our gratitude to **Sri. D. LAKSHMINARAYANASWAMY,** Managing Trustee, **Sri. R. SUNDAR,** Joint Managing trustee, SNR Sons Charitable Trust, Coimbatore for providing excellentfacilities to carry out our project.

We express our deepest gratitude to our Principal, **Dr. N. R. ALAMELU, Ph.D.,** for her valuable guidance and blessings.

We are indebted to our Head of the Department, **Dr. A. Grace Selvarani, Ph.D.,** Department of Computer Science and Engineering who modelled us both technically and morally for achieving great success in life.

We express our thanks to our Project Coordinator, **Mrs. G. Rathi.,** Assistant Professor (Sl. Grade) Department of Computer Science and Engineering for her great inspiration.

Words are inadequate to offer thanks to our respected guide. We wish to express our sincere thanks to **Mrs.C.Padmavathy,** Assistant Professor (Sl.Gr), Department of Computer Science and Engineering, who gives constant encouragement and support throughout this project work and who makes thisproject a successful one.

We also thank all the staff members and technicians of our Department for their help in making this project a successful one.

TABLE OF CONTENTS

ABSTRACT

Distributed Denial of Service (DDoS) attacks pose a significant risk tocybersecurity, with potential consequences ranging from server failures to inconveniencing users. These attacks can originate from the application layer or the network layer, where the attacker and victim systems are connected in anetwork. DDoS attacks can have severe effects on businesses and financial institutions, resulting in loss of reputation, productivity, revenue, and even theft.

To mitigate the impact of DDoS attacks, there is a pressing need for effective detection and prevention techniques. One promising solution is the use of machine learning approaches, which can analyze features and patterns in network traffic to predict and classify different types of DDoS attacks. Deep learning methods, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have shown promising results in identifying these patterns and predicting DDoS attacks in real-time.

This project aims to propose a machine learning approach for DDoS attack classification and prediction using deep learning methods. The proposed solution involves feature analysis and extraction from network traffic data, followed by training and testing of the deep learning models. The performance of the models is evaluated based on accuracy, precision, recall, and F1-score. The results show that the deep learning models have high accuracy and can effectively predict DDoS attacks in real-time.

Overall, this project contributes to the development of effective DDoS attack detection and prevention techniques using machine learning, which can help to protect businesses and financial institutions from the significant impact of these attacks.

சுருக்கம்

விநியோகிக்கப்பட்ட சேவை மறுப்பு (DDoS) தாக்குதல்கள் இணைய பாதுகாப்பிற்கு குறிப்பிடத்தக்க ஆபத்தை ஏற்படுத்துகின்றன, சர்வர் செயலிழப்புகள் முதல் பயனர்களுக்கு சிரமத்தை ஏற்படுத்துவது வரை சாத்தியமான விளைவுகள். இந்தத் தாக்குதல்கள் பயன்பாட்டு அடுக்கு அல்லது நெட்வொர்க் லேயரில் இருந்து உருவாகலாம், அங்கு தாக்குபவர் மற்றும் பாதிக்கப்பட்ட அமைப்புகள் நெட்வொர்க்கில் இணைக்கப்பட்டுள்ளன. DDoS தாக்குதல்கள் வணிகங்கள் மற்றும் நிதி நிறுவனங்களில் கடுமையான விளைவுகளை ஏற்படுத்தலாம், இதன் விளைவாக நற்பெயர், உற்பத்தித்திறன், வருவாய் மற்றும் திருட்டு இழப்பு போன்றவை ஏற்படும்.

DDoS தாக்குதல்களின் தாக்கத்தைத் தணிக்க, பயனுள்ள கண்டறிதல் மற்றும் தடுப்பு உத்திகள் தேவை. பல்வேறு வகையான DDoS தாக்குதல்களைக் கணிக்கவும் வகைப்படுத்தவும் நெட்வொர்க் போக்குவரத்தில் உள்ள அம்சங்களையும் வடிவங்களையும் பகுப்பாய்வு செய்யக்கூடிய இயந்திர கற்றல் அணுகுமுறைகளைப் பயன்படுத்துவது ஒரு நம்பிக்கைக்குரிய தீர்வாகும். கன்வல்யூஷனல் நியூரல் நெட்வொர்க்குகள் (சிஎன்என்கள்) மற்றும் ரிக்ரரெண்ட் நியூரல் நெட்வொர்க்குகள் (ஆர்என்என்கள்) போன்ற ஆழமான கற்றல் முறைகள், இந்த வடிவங்களை அடையாளம் கண்டு, நிகழ்நேரத்தில் டிடிஓஎஸ் தாக்குதல்களைக் கணிப்பதில் நம்பிக்கைக்குரிய முடிவுகளைக் காட்டியுள்ளன.

இந்த திட்டம் DDoS தாக்குதல் வகைப்பாடு மற்றும் ஆழமான கற்றல் முறைகளைப் பயன்படுத்தி கணிப்புக்கான இயந்திர கற்றல் அணுகுமுறையை முன்மொழிவதை நோக்கமாகக் கொண்டுள்ளது. முன்மொழியப்பட்ட தீர்வு அம்ச பகுப்பாய்வு மற்றும் நெட்வொர்க் ட்ராஃபிக் தரவிலிருந்து பிரித்தெடுத்தல் ஆகியவற்றை உள்ளடக்கியது, அதைத் தொடர்ந்து ஆழ்ந்த கற்றல் மாதிரிகளின் பயிற்சி மற்றும் சோதனை. மாதிரிகளின் செயல்திறன் துல்லியம், துல்லியம், திரும்பப் பெறுதல் மற்றும் F1-ஸ்கோர் ஆகியவற்றின் அடிப்படையில் மதிப்பிடப்படுகிறது. ஆழமான கற்றல் மாதிரிகள் அதிக துல்லியம் மற்றும் நிகழ்நேரத்தில் DDoS தாக்குதல்களை திறம்பட கணிக்க முடியும் என்பதை முடிவுகள் காட்டுகின்றன.

ஒட்டுமொத்தமாக, இந்தத் திட்டம், இயந்திரக் கற்றலைப் பயன்படுத்தி பயனுள்ள DDoS தாக்குதல் கண்டறிதல் மற்றும் தடுப்பு நுட்பங்களை மேம்படுத்துவதற்கு பங்களிக்கிறது, இது வணிகங்களையும் நிதி நிறுவனங்களையும் இந்தத் தாக்குதல்களின் குறிப்பிடத்தக்க தாக்கத்திலிருந்து பாதுகாக்க உதவும.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

DDoS	-	Distributed Denial of Service
CNNs	-	Convolutional Neural Networks
RNNs	-	Recurrent Neural Networks
COVID-19	-	Corona Virus Disease 2019
HHS	-	Health and Human Service
US	-	United States
Q1	-	First quarter
Q4	-	Fourth quarter
AS	-	Autonomous System
ISP	-	Internet Service Provider
KNN	-	K-Nearest Neighbors
UCI	-	University of California, Irvine
SVM	-	Support Vector Machine
HTML	-	Hypertext Markup Language
CSS	-	Cascading Style Sheets
NSL-KDD	-	Network Security Lab-KDD Cup 1999 dataset
GCNs	-	Graph Convolutional Networks
PSH	-	Push Flag in TCP protocol
ACK	-	Acknowledgement Flag in TCP protocol
IDS	-	Intrusion Detection Systems
TCP	-	Transmission Control Protocol
CPU	-	Central Processing Unit
AI	-	Artificial Intelligence
FP	-	Feature Processing
BP	-	Back Propagation
TP	-	True Positive
FP	-	False Positive
TN	-	True Negative
FN	-	False Negative
AE	-	Autoencoder
DBN	-	Deep Belief Network
F1-score	-	F1 measure, a harmonic means of precision and recall

CHAPTER 1

INTRODUCTION

Distributed Denial of Service (DDoS) attacks have become one of the most prevalent and damaging cyber threats in recent years. These attacks aim to disrupt online services by overwhelming them with a flood of traffic, rendering them inaccessible to users. With the increasing dependence of businesses and individualson online services, the impact of these attacks can be devastating. Therefore,predicting and mitigating such attacks has become crucial in ensuring the availability and security of online services.

Deep Learning Convolutional Neural Networks (CNNs) have shown great potential in various applications, including computer vision, natural language processing, and cybersecurity. In the case of DDoS attack prediction, CNNs can betrained on network traffic data to learn the patterns and characteristics of normal and anomalous traffic. This can enable the CNN model to accurately detect and classify incoming traffic as legitimate or malicious, thus predicting and mitigating potential DDoS attacks.

In this project, we will explore the application of CNNs for DDoS attack prediction, discussing the relevant background and related works. We will then present our proposed CNN-based approach, detailing the data preprocessing, model architecture, and training process. Finally, we will evaluate the performance of our approach using real-world traffic data and compare it with existing methods. Our study aims to contribute to the development of more effective and efficient DDoS attack prediction and mitigation strategies, ultimately enhancing the resilience and security of online services.

DDoS ATTACK INCIDENTS IN REAL WORLD

Human life has completely shifted to the online since Jan-2020 due to the COVID-19 (Corona Virus Disease) pandemic situation throughout the world wide. In this pandemic time, people learn how to work, how to do shopping, etc. through online services. According to the Q1-2020 DDoS attacks statistical report, attackers mostly targeted pharmaceuticals organizations, food delivery services, distribution services, entertainment, and learning platforms. In March-2020, the attackers performed a DDoS attack on the Health and Human Service (HHS) of the US Department. However, attackers failed to shut down the HHS online service. It has been running before and after attacks, despite the huge load. Further, attackers launched attacks on two online food delivery portals such as Lieferando (Germany) and Thuisbezorgd (Netherlands). During the attack, both services could accept online orders but failed to process requests and refundmoney. For halting the ongoing attack, attackers demanded 2 Bitcoins (approximately 1+ Lac US dollars) from Lieferando. Further, it reveals thatattackers have shown their interest to deploy Windows operating system basedbotnets. The share of Windows botnets turned to 5.64% (Q1-2020) from 2.6%.However, nine out of ten DDoS attacks still proceed using Linux botnets (94.36%).The number of DDoS attacks has grown by 84% compared to the Q4-2018 DDoSattacks.

NEED OF DDoS ATTACK DETECTION SYSTEM

According to the statistical report of DDoS attacks, leading portals such as Yahoo!, Github, Amazon, etc., have lost approximately $2.5 million of revenue because of DDoS attacks. Further, it shows that attack volume size is steadily increasing quarter-after quarter. There are several reasons behind executing DDoS attacks in the victim's system and its resources. A few of them are listed in the following:

Business competition:

The motive of this type of attack is to overwhelm the competitor's website.
They send a massive amount of network flows towards the competitor's website, and due to this, benign users experience delay or denial while accessing the victim system. Therefore, the image and reputation of the organization go down in the market. This type of attack is extremely destructive because it is performed by experts.

Cyber-vandalism:

Most of the time, vandals seek to release their anger or disappointment against an organization. Sometimes, this can perform for making fun or entertainment purpose by using ready-made scripts or tools.

Extortion:

Extortion is a common motivation behind launching a large-scale DDoS attack. Several web portals, such as MeetUp, Bitly, Basecamp, Lieferando, etc. suffered from this type of attack. This type of attack is launched by experts and demands money from the victim organization for not performing or halting ongoing DDoS attacks. The attacker uses a DDoS attack as a monetary weapon, and they demand money from the victim organization to stop the ongoing attack.

DEPLOYMENT LOCATIONS OF DDoS DETECTION SYSTEMS

Numerous DDoS defense mechanisms have been proposed by researchers to protect victim system and resources from different types of DDoS attacks. According to the deployment locations of the system, they can be categorized into four locations: source-end, intermediate network, victim-end, and hybrid and their features and limitations are listed in the following:

1. Source-end:

(a) Deploy near to the attack sources on edge or access routers of attack sources networks

(b) Easy to control flooding DDoS attack stream in attack source networks itself

(c) Help to perform defense actions on attack streams using filtering and rate-limiting techniques

(d) Ability to recognize attack streams before collateral damage

(e) Challenging task for real-time implementation

2. Intermediate Network:

(a) Defense systems deployed in intermediate networks (single or multiple ISPs) on multiple routers of Autonomous System (AS)

(b) Help to filter DDoS attacks traffic using various filtering techniques

(c) Help to a trade-off between detection accuracy and bandwidth consumption

(d) Practically challenging for real-time implementation and also expensive

3. Victim-end:

(a) Defense systems deployed in the victim network either on the detection server or edge router of AS

(b) Closely observes the network traffic

(c) Practically feasible and cost-effective solution

(d) Easy to discriminate benign and DDoS attacks streams

(e) Difficult to implement filtering and rate-limiting techniques

Existing system

Gozde Karatas et al. has proposed a machine learning approach for the classification of attacks. They used different machine learning algorithms and found that the KNN model is best for classification as compared to other research work.

Nuno Martins et al. has proposed intrusion detection using machine learning approaches. They used the KDD dataset which is available on theUCI repository. They performed different supervised models to balance classification algorithm for better performance. In this work a comparative studywas proposed by the use of different classification algorithms and found good results in their work.

Laurens D’hooge et al. has proposed a systematic review for malware detection using various machine learning models. They have compared different datasets of malware from various online resources. They found that machine learning supervised models are very effective for malware detection to make a better decision in less time.

Xianwei Gao et al. has proposed a comparative work for classification of network traffic. They used machine learning classifiers for intrusion detection. The dataset is taken is CI-CIDS and KDD from the UCI repository. They found support vector machine SVM one of the best algorithms as compare to others.

Disadvantages

- Finding a way can be complex.
- Sensitive to the scale of the data and irrelevant features.
- Lack of probabilistic classification.

Proposed system:

In this project, the proposed system is based on Convolutional Neural Network with deep learning for classification. This algorithm inspired by biological neural networks (where the brain is considered particularly important in the central nervous system) and are used in statistics and cognitive science. These are represented by the interconnection of neural systems from various input variablesto the output variables, and it can be represented as mathematical functions that are configured to represent complex relationships between inputs and outputs, i.e, independent and dependent variables respectively. In our project we are predicting three class of attacks such as DDoS-PSH-ACK, DDoS-ACK, Benign. The final output will be displayed in a Web Application which is created by using HTML and CSS.

Advantages

- **Effective Feature Extraction:** CNNs can be used to extract relevant features from time series or sequence data, which can be useful for classification or prediction tasks.
- **Robustness to Noise:** CNNs can filter out the noise and other irrelevant information from time series data, making them robust to noisy input data.
- **Automatic Feature Learning:** Similar to their use in image processing, CNNs can automatically learn important features from time series or sequence data, without the need for manual feature engineering.

- **Transfer Learning:** Pre-trained CNNs can be used as a starting point for new tasks, allowing for efficient transfer of knowledge and reducing the need for large amounts of labelled data.

CHAPTER 2
LITRATURE SURVEY

"DDoS Attack Detection and Classification using Random Forest Algorithm," by J. Kumar and S. Kumar:

The paper proposes a method for detecting and classifying DDoS attacks using the Random Forest algorithm. The authors use the NSL-KDD dataset to train and test their model and achieve an accuracy of 99.66%. The proposed method is able to detect various types of DDoS attacks, including TCP, UDP, and ICMP floods. The authors also compare their method with other machine learning algorithms, and Random Forest outperforms them in terms of accuracy and execution time. Overall, the paper demonstrates that Random Forest is a promising approach for detecting and classifying DDoS attacks.

"DDoS Attack Detection Using Deep Convolutional Neural Networks," by A. R. Lashkari, et al.:

In this paper, the authors propose a DDoS attack detection approach using deep convolutional neural networks (CNNs). The proposed approach uses a multi-scale CNN architecture to extract features from the network traffic, and then uses a support vector machine (SVM) classifier for attack detection. The authors show that the proposed approach outperforms traditional machine learning algorithms, such as random forests and SVMs, in terms of detection accuracy. The proposed approach also demonstrates high robustness against adversarial attacks. The results indicate that deep CNNs can be a promising technique for detecting DDoS attacks in network traffic.

"Graph Convolutional Networks for DDoS Attack Detection," by J. Xu, et al.: The paper "Graph Convolutional Networks for DDoS Attack Detection" proposes a novel DDoS attack detection method using graph convolutional networks (GCNs). GCNs are utilized to analyze the network traffic graphs and extract important features for identifying DDoS attacks. The proposed method can detect and classify various types of DDoS attacks with high accuracy, even in the presence of noise and uncertainties in the network. The experimental results show that the proposed GCN-based approach outperforms traditional machine learning and deep learning methods for DDoS attack detection.

"A Hybrid DDoS Attack Detection Method Using Clustering and Support Vector Machine," by H. M. Javed, et al.

The paper proposes a hybrid method for detecting DDoS attacks that uses clustering and support vector machines (SVM). The method first clusters network traffic data based on traffic patterns and then trains an SVM model on the extracted features from the clustered data. The proposed method is evaluated on the NSL-KDD dataset and compared with other traditional machine learning methods. The experimental results show that the proposed hybrid method performs better than other traditional machine learning methods and achieves higher detection rates and lower false positive rates.

"DDoS Attack Detection using Machine Learning Techniques," by S. Shanthi and V. Vaidehi

In this paper, the authors propose a machine learning-based DDoS attack detection approach using several classification techniques such as Decision Tree, Random Forest, and Naïve Bayes. The proposed approach uses statistical features such as packet rate, payload size, and number of connections to classify network traffic as normal or attack. The experiments show that the proposed approach achieves high accuracy in detecting DDoS attacks while keeping the false positive rate low. The authors conclude that the proposed approach can be used as an effective tool for DDoS attack detection.

"Anomaly-Based Detection of DDoS Attacks Using SVM and Random Forest," by J. Jang, et al.

In this paper, the authors propose an anomaly-based detection method for DDoS attacks using both Support Vector Machines (SVM) and Random Forest algorithms. The proposed method first extracts features from network traffic and then uses SVM to classify traffic into normal and abnormal classes. The classified traffic is then used to train a Random Forest model, which is used to classify new traffic into normal and attack classes. The proposed method was evaluated on a publicly available dataset, and the results showed that it outperformed traditional machine learning methods in terms of accuracy and detection rate.

"DDoS Attack Detection Based on Improved K-Means Clustering and SVM Algorithm," by Y. Wang, et al.

This paper proposes an improved detection method for DDoS attacks based on a combination of K-means clustering and support vector machine (SVM). The method uses K-means clustering to group similar traffic flows and then trains an SVM classifier on the features extracted from the clustered data. The proposed method achieved better accuracy and lower false-positive rates than existing methods. The study concluded that the proposed method provides an efficient way of detecting DDoS attacks in large-scale networks.

"DDoS Detection in the Cloud using Machine Learning Techniques," by R. El Khayat, et al.

This paper proposes a framework for DDoS detection in the cloud environment using machine learning techniques. The framework uses various features such as packet length, packet direction, and packet type to detect DDoS attacks. The authors compare the performance of different classifiers including Random Forest, k-NN, Naive Bayes, and SVM, and show that SVM performs better in terms of accuracy and false alarm rate. The proposed framework is effective in detecting DDoS attacks in a cloud environment, with high accuracy and low false alarm rate.

"A Review of Machine Learning Techniques for DDoS Attack Detection," by B. Ahmed, et al.

Ahmed et al. present a comprehensive review of machine learning techniques for DDoS attack detection. The paper summarizes the current state-of-the-art and identifies various limitations of existing detection methods. The authors categorize the techniques into different groups such as rule-based, anomaly-based, and machine learning-based methods. They also compare the performance of different algorithms, including decision trees, support vector machines, and neural networks, and discuss the challenges associated with real-time detection, feature selection, and classification accuracy. The review highlights the need for more research to improve the accuracy, scalability, and reliability of DDoS attack detection systems.

"A Novel Approach for DDoS Attack Detection using Neural Networks," by M. S. S. Farhan and M. S. Hossain:

This paper proposes a new method for detecting DDoS attacks using Artificial Neural Networks (ANN). The authors used the backpropagation algorithm for training the neural network and measured the accuracy of the proposed model with several performance metrics such as accuracy, precision, recall, and F-measure. The results show that the ANN-based approach is effective in detecting DDoS attacks with a high detection rate and low false-positive rate. The authors also compared their proposed method with other existing approaches and found that it outperforms them in terms of detection accuracy.

"DDoS Attack Detection using Ensemble Machine Learning Algorithms," by M. Sharma and N. Nain

This paper proposes a new approach for detecting DDoS attacks using ensemble machine learning algorithms. The proposed method uses three different machine learning algorithms, namely Decision Tree, Naive Bayes, and Random Forest, to detect attacks. The authors use different performance metrics such as accuracy, precision, recall, and F1-score to evaluate the performance of the proposed method. The results show that the proposed method is effective in detecting DDoS attacks with an accuracy of 97.4% and an F1-score of 0.98.

"Real-Time Detection and Mitigation of DDoS Attacks using Machine Learning Algorithms," by S. R. Hema, et al

The paper proposes a framework for the real-time detection and mitigation of DDoS attacks using machine learning algorithms. The framework consists of pre-processing, feature extraction, and classification stages. Various machine learning algorithms, including decision tree, random forest, K-nearest neighbors, and artificial neural networks, are used for classification. The proposed framework is evaluated using the NSL-KDD dataset, and the experimental results show that it can effectively detect and mitigate DDoS attacks in real-time.

"A Comparative Study of Machine Learning Algorithms for DDoS Attack Detection," by R. Saha, et al.

This paper aims to compare the performance of different machine learning algorithms for DDoS attack detection. The study evaluated seven algorithms, including decision tree, k-nearest neighbor, random forest, support vector machine, Naïve Bayes, neural network, and logistic regression, on the basis of accuracy, precision, recall, F1-score, and area under the curve (AUC) metrics. The results showed that random forest achieved the highest accuracy and AUC, followed by neural network and support vector machine, while decision tree and Naïve Bayes showed the lowest performance. The study provides insights into the effectiveness of various machine learning techniques for DDoS attack detection.

"Feature Selection for DDoS Attack Detection using Machine Learning Techniques," by J. Han, et al.

This paper presents a novel approach for feature selection in DDoS attack detection using machine learning techniques. The authors propose a new feature selection method based on mutual information and particle swarm optimization (PSO) to select the most important features from the original feature set. The selected features are then used to train various machine learning models, including decision tree, random forest, and support vector machine (SVM), to detect DDoS attacks. The experimental results show that the proposed feature selection method improves the performance of the machine learning models in terms of accuracy, precision, and recall, and reduces the processing time and computational complexity of the detection system.

CHAPTER 3
MODELLING ATTRIBUTES

Proposed System:

CNN using deep learning for classification are the foundation of the system suggested in this study. This algorithm draws inspiration from biological neural networks, which have applications in both statistics and cognitive science (the brain being viewed as especially significant in the central nervous system). They can be represented as mathematical functions that are set up to depict complicated interactions between inputs and their corresponding outputs, which are in turn represented by the integration of brain systems from diverse input variables. The proposed flow diagram is defined in the below figure.1. In our paper, we are predicting three class of attacks. Such as DDoS-PSH-ACK or DDoS-ACK or Benign. The final output will be displayed in a Web Application which is created by using HTML and CSS.

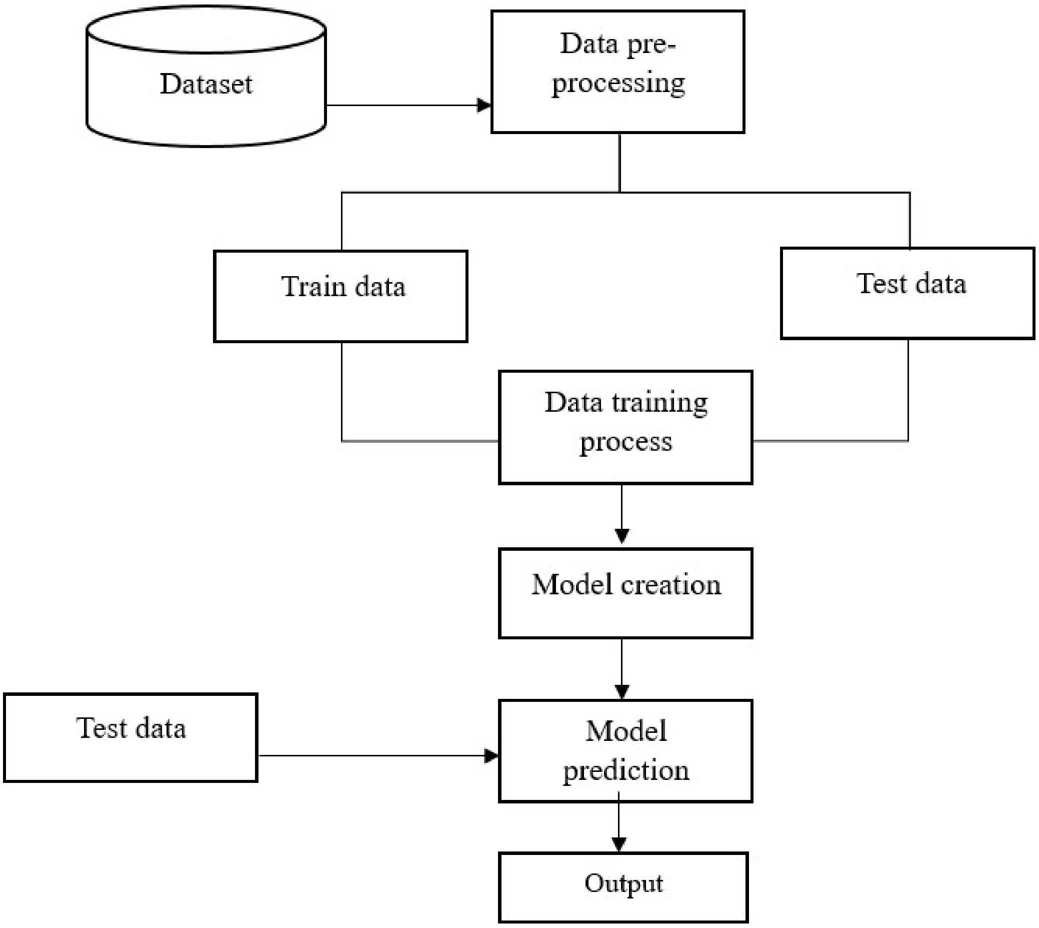


Figure.1. Proposed flow diagram.

The proposed system is segregated into various blocks such as, Data collection, Data Pre-Processing, Training data and Test data, Model Creation and Model Prediction.

3.1. Data collection:

The project aims to address a gap in existing datasets used for machine learning-based intrusion detection systems (IDS). The exponential increase in connected devices has made the detection of malicious connections a more challenging task. IDS is one of the mechanisms used for this purpose, but given the sophistication of attacks and the variety of normal traffic patterns, machine learning is becoming increasingly popular in IDS.

To effectively use machine learning in IDS, it is important to have access to attack patterns that represent a wide range of attacking traffic. However, some infamous attack patterns, such as ACK and PUSH-ACK flooding DDoS attacks, are underrepresented in existing datasets used by the machine learning community. In this project a dataset that includes these common attack patterns is used, which can be used by IDS developers to increase the detection ratio of their detection modules. By addressing this gap, the project may contribute to improving the effectiveness of IDS in detecting and mitigating DDoS attacks.

3.2. Data Pre-Processing:

Data preprocessing is a crucial step in preparing data for analysis using machine learning algorithms. It involves transforming raw data into a format that is suitable for analysis. The process includes a variety of techniques such as cleaning, normalization, transformation, and reduction.

Cleaning involves removing or correcting missing, inconsistent, or inaccurate data. Normalization involves scaling the data to a common range, which makes it easier to compare and analyze. Transformation involves converting the data into a different format, such as converting categorical variables into numerical variables. Reduction involves reducing the dimensionality of the data, which makes it easier to analyze.

By preprocessing the data, we can ensure that it is in a format that can be effectively used by the machine learning algorithms. This can help to improve the accuracy and effectiveness of the algorithms, and ultimately lead to more accurate prediction and insights.

3.3. Training data and Test data:

- For choosing a model we split our dataset into train and test
- The data is split in the ration of 3:1, meaning, training dataset has 70% and testing dataset has 30%.
- In this split process preforming based on train, test, split model
- After splitting the dataset we get xtrain, xtest and ytrain, ytest

3.4. Model Creation:

- Contextualize machine learning in your organization.
- Exploring the data and choosing the type of algorithm.
- Prepare and clean the dataset.
- Splitting the prepared dataset and performing cross validation.
- Perform machine learning optimization.
- Deploy the model.

3.5. Prediction:

Predictive modeling is a powerful tool for analyzing large datasets and identifying patterns and trends that can be used to make predictions about future outcomes. It involves using machine learning algorithms to analyze data and build predictive models that can be used to forecast future trends and events.

In the context of security, predictive modeling can be used to identify patterns and anomalies in network traffic that may indicate a potential cyber attack. By analyzing historical data on cyber attacks and normal network traffic, predictive models can be trained to recognize patterns and predict the likelihood of an attack.

In this project, the goal is to build a predictive model that can identify different types of cyber attacks based on their attributes. This involves analyzing data on different types of attacks, as well as normal network traffic, to identify patterns and build a model that can accurately classify new instances of network traffic as either normal or indicative of a particular type of attack. The resulting predictive model can then be used to proactively identify and mitigate potential cyber threats.

3.5.1 DDoS-PSH-ACK:-

The PSH-ACK DDoS attack is a type of TCP flood attack that involves sending a large number of packets with the PUSH and ACK flags set in the TCP header. The purpose of this attack is to overwhelm the victim's system by consuming its resources such as CPU time, memory, and network bandwidth. This type of attack is often used in combination with other types of DDoS attacks to maximize their impact. The PSH-ACK DDoS attack is difficult to detect and mitigate because it appears similar to legitimate traffic, and it can be launched from multiple sources, making it challenging to block.

3.5.2 DDoS-ACK :-

In a DDoS ACK attack, the attacker sends a flood of spoofed TCP ACK packets to a target system. These packets have invalid sequence numbers, so the target system sends a reset packet in response. This consumes significant amounts of the target system's resources, eventually causing it to become unavailable to legitimate users. The main objective of a DDoS ACK attack is to deplete the target system's resources and cause it to crash or become unresponsive.

3.5.3 Benign: -

A benign traffic is a normal, legitimate traffic that flows between servers, applications, and end-users. In the context of DDoS attacks, it is used as a reference or baseline traffic to compare against malicious traffic. By analyzing benign traffic patterns, it is possible to identify anomalies and abnormal traffic patterns associated with DDoS attacks.

3.6. Proposed CNN model

One of the most recent breakthroughs in AI is deep learning. The availability of data management and data processing tools allowed academics to create solutions that were previously merely theoretical. In this study, we employ 1D-CNN (convolutional neural network) for network intrusion detection. 1D-CNN is optimised for use with 1-dimensional data. In order to create answers for one-dimensional data, researchers have developed several forms of 2D-CNN.:

- ❖ It is widely agreed that FP and BP are the backbone of the convolutional neural network (CNN) design. As 1D-CNN does not require any matrix operations, its computational cost is orders of magnitude lower than that of 2D-CNN.
- ❖ Shallow network designs may learn from the difficult 1D data, and they are simpler to understand, train, and implement. On the other hand, 2D-CNNs employing deep architectures may be necessary to carry it out.
- ❖ While 2D-CNN necessitates specialist hardware, 1D-CNN may be trained to use fewer computing resources for 1D data.

For processing one-dimensional data, 1D-CNN employs 1D convolution layers, pooling layers, dropout layers, and activation functions. Number of CNN layers, number of neurons per layer, filter size, and subsampling factor per layer are the hyper-parameters used to set up 1D-CNN.

When applied to an input, the filter is at its most fundamental at the convolution layer. Repeatedly applying the filtering procedure results in a feature map, which displays the specific characteristics of the data points. The process of convolution may be thought of as a linear multiplication where the inputs are weighted. The kernel in this scenario is a one-dimensional array of weights that is multiplied by the inputs. When carried out, this procedure yields a set of values known as a feature map, each of which is distinct from the others.

Each value is then sent into the ReLU activation function once the feature map has been generated. If the input is negative, ReLU modifies it to zero, and else it returns the same value. Model performance is improved, training input is absorbed function. This is shown using Eq. (1) below.:

$R(z) = \max(0, z)$ (1)

Thus, z is the input to the activation function, and R(z) is a positive result from the activation function. The pooling layers of a CNN are typically placed after the convolution layers. The Sub-Sampling method is being utilised internally to lessen the importance of perfectly centred feature maps on the model; nonetheless, the feature maps should be position-independent to prevent the model from overfitting the data. Pooling layers work on the feature maps to generate the mapped pooled features, and the computation of the architecture is based on the complexity and number of parameters. The pooling filter size, stride, and pooling type (max pooling, average pooling) all play a role in determining which mapped pooled features are chosen. Max pooling determines the feature's maximum value for each patch, whereas average pooling determines the feature's average value. Overfitting the training data can impair the model's overall performance when tested on unknown data, which is a risk with deep learning neural networks. Hence, we used a technique called "dropout layers." It's a form of regularisation in which certain neurons that are being processed arbitrarily are ignored. To avoid overfitting the model, it resets the inputs to zero at each iteration during training. Then, using Equation, the active inputs are multiplied by a constant so that the total input stays unchanged (2).

$$z = \frac{1}{1-r} \quad (2)$$

As a result, this adds noise to the training process by placing a disproportionate amount of work on a small number of nodes. Its sole application is in the context of education. Nevertheless, dropout adds to the network's burden, necessitating scaling up to the selected dropout rate. Finally, an activation function is used to translate the findings to the output after the dense layers, which are entirely coupled to the layer below.

3.6.1. Proposed Architecture

In the modern period, deep learning has been essential in spotting network breaches. To identify malicious activity in networks, we present a cutting-edge 1DCNN-based deep learning architecture in this research.

5 convolution layers, 1 dropout layer, 2 pooling layers, and 5 dense layers make up the suggested architecture (see Table 1). Using ReLU as the activation function, the first CNN layer features 32 filters with a kernel size of 2. The input sample is used as an input by this layer, which then outputs a (77, 32) shaped vector. After that comes a second convolution layer, this one using 16 filters with a kernel size of 2. To prevent the model from overfitting, we included a dropout with a 0.5 rate that disables 50% of the neurons during training, placed after five

successive convolution layers. After the dropout layer, a max-pooling with a pool size of 2 is applied. The max-pooling layer is used to lower the computational cost of the model by decreasing the number of parameters to learn. The output is then "flattened," or reduced from three dimensions to one, using a flattening layer. Next, for the purpose of predicting a target variable, we used five thick layers with a ReLU and SoftMax activation function. Table 1 provides a brief overview of the model..

Layer	Sum of Parameters	Output Shape
Dense	1275	25
Dense	130	5
Conv1d	528	(75, 16)
Conv1d	528	(74, 16)
Conv1d	528	(73, 16)
dropout	0	(73, 16)
Max pooling	0	(36, 16)
Flatten	0	576
Dense	57,700	100
Dense	7575	75
Dense	3800	50
Conv1d	96	(77, 32)
Conv1d	1040	(76, 16)

Table 1. Summary of the projected model.

Adam was utilised as the optimizer, Relu was chosen as the activation function, Dropout was set to 0.5, and 50 iterations were examined.

Flask:

- Flask is a lightweight and flexible web application framework written in Python. It provides a wide range of modules and tools that make it easier for developers to create web applications without worrying about low-level details such as protocol management and thread management.
- Flask also offers a variety of choices for developing web applications. It supports multiple templating engines, allowing developers to choose the one that best fits their needs. Additionally, Flask supports a wide range of extensions and libraries that make it easy to add new functionality to a web application

Pickle:

- Pickle can be used to serialize Python object structures, which refers to the process of converting an object in the memory to a byte stream that can be storedas a binary file on disk. When we load it back to a Python program, this binary file can be de-serialized back to a Python object.
- This is used for serializing and de-serializing Python object structures, also called marshalling or flattening. Serialization refers to the process of converting an object in memory to a byte stream that can be stored on disk or sent over a network.

CHAPTER 4
RESULTS

When it comes to deep learning is one of the most recent breakthroughs. Researchers were able to provide answers that were just theoretical a decade ago due to a lack of data management and processing capabilities. In this study, we employ 1D-CNN for intrusion detection in computer networks. 1D-CNN was developed to work solely with one-dimensional datasets. Prior to this, researchers utilised a wide variety of general and machine learning-based algorithms to identify network intrusions.

4.1. System Requirement

The proposed system has implemented using Windows 10 with 4GB Ram and i3 Processor. The programming of the proposed system is done in Python 3.9. Python is an open source programming language.

4.2. Evaluation Metrics

Many measures, including accuracy, precision, recall, and F1 score, were used to assess the 1D-CNN classifier's efficacy. The rationale behind these measurements may be found in Equations (3)–(6).

$$Accuracy = \frac{TP+}{Total\ Samples} \quad (3)$$

$$Precision = \frac{TP}{TP+} \quad (4)$$

$$Recall = \frac{TP}{TP+FN} \quad (5)$$

$$F1\ Score = \frac{T2 \times Precision \times Recall}{Precision+Rec} \quad (6)$$

where TP stands for the proportion of assaults that are legitimate, FP stands for benign data that has been labelled a threat, The notation TN indicates normally classified data, whereas the notation FN indicates a normally classified assault.

Method	Accuracy (%)	Precision (%)	Recall (%)	F-score (%)
Proposed	87.86	80.33	98.42	88.46
DBN	85.11	79.57	96.16	87.89
AE	84.13	77.99	93.16	87.31
SVM	83.19	76.76	90.61	86.71

Table 2: Analysis of Proposed Model for 60%-40% of data

In the investigation of accuracy, the projected model achieved better performance, i.e., 87.86%, SVM, AE and DBN achieved nearly 83% to 85%. When comparing with all techniques, SVM achieved poor performance, 76.76% of precision, 90.61% of recall and 86.71% of F1-score. AE achieved 77.99% of precision, 93.16% of recall and 87.31% of F-score, where the proposed model achieved 80.33% of precision, 98.42% of recall and 88.46% of F-score. However, the projected model achieved low performance, when the training data is 60% and testing data is 40%.

Method	Accuracy (%)	Precision (%)	Recall (%)	F-score (%)
Proposed	91.2	92.53	97.17	91.59
DBN	90.91	91.54	96.57	90.94
AE	89.88	89.22	96.5	90.5
SVM	89.93	86.15	95.15	90.43

Table 3: Validation analysis of Proposed Model for 80%-20% of data

When the training data and testing data are increased, the performance of all models are increased for various metrics. In the proposed model achieved 91.2%, DBN achieved 90.91%, AE achieved 89.88% and SVM achieved 89.93%. All models are tested with precision, recall and F-score analysis. SVM achieved nearly 86% of precision, 95.15% of recall and 90.43% of F-score, where the projected model achieved 92.53% of precision, 97.17% of recall and 91.59% of F-score. Figure 2 to 5 presents the graphical analysis of various classifiers.

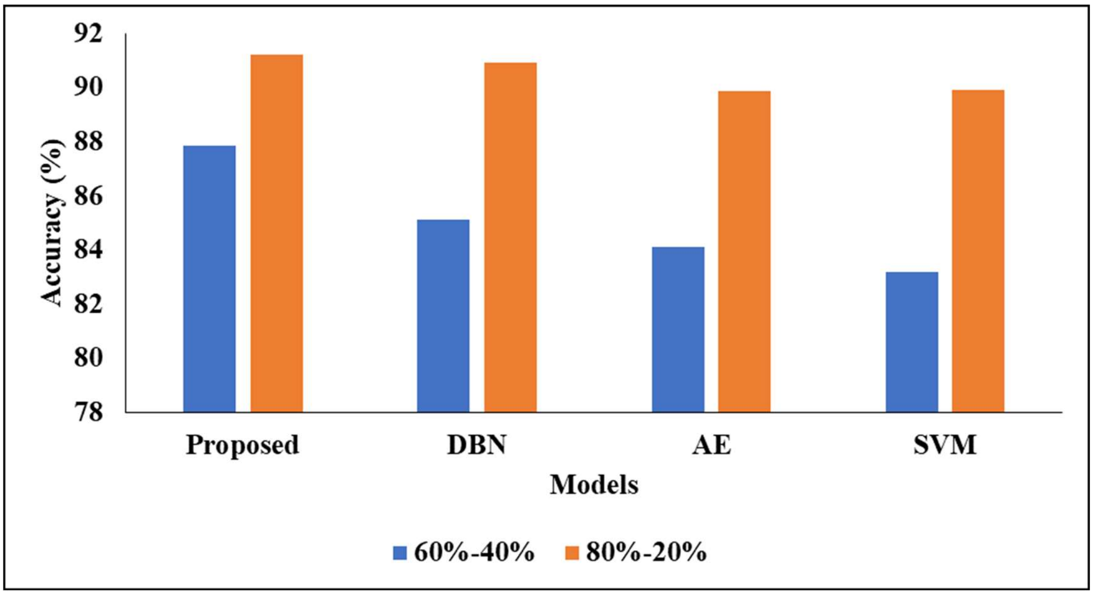


Figure 2: Accuracy Comparison

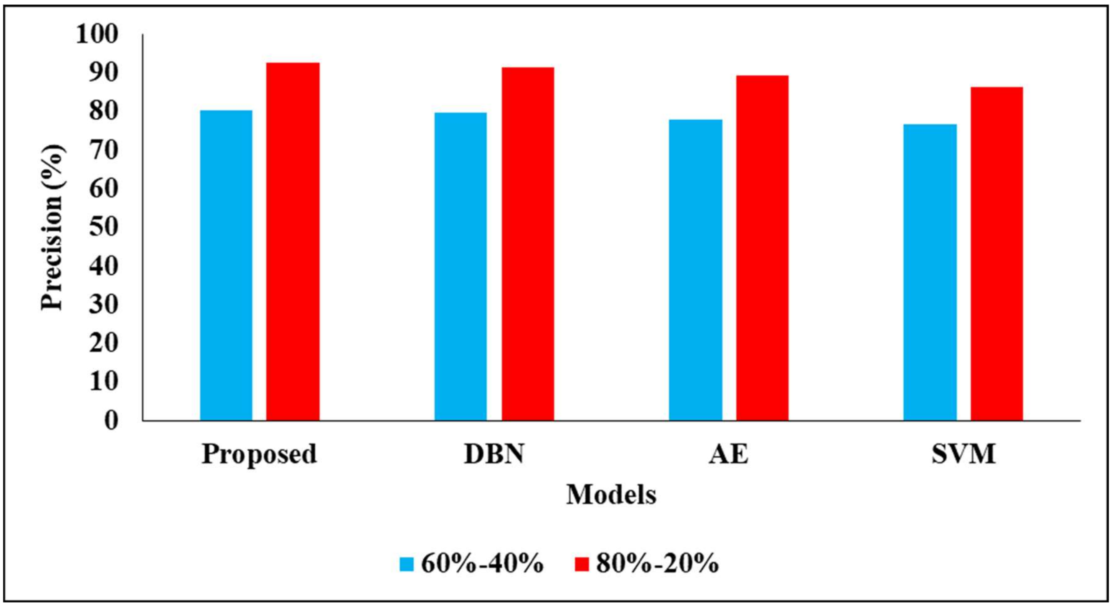


Figure 3: Precision Analysis

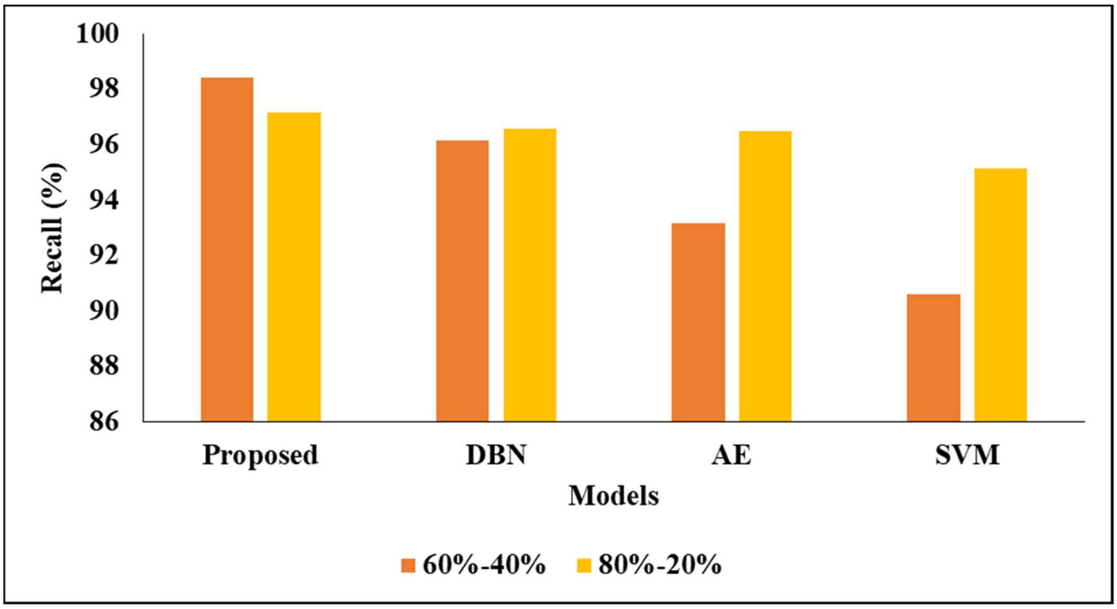


Figure 4: Recall Analysis

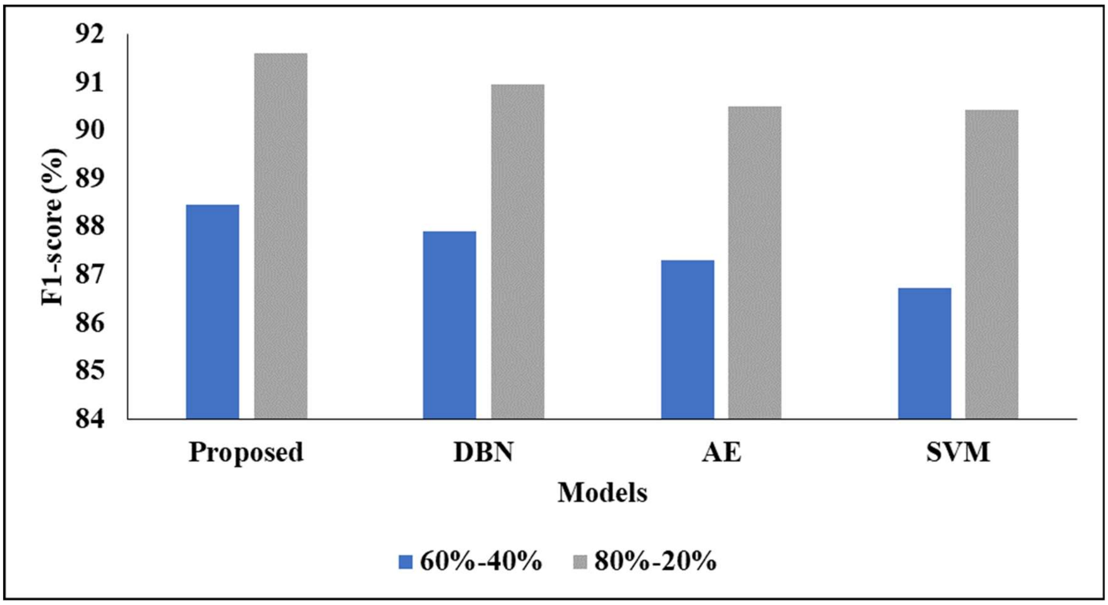
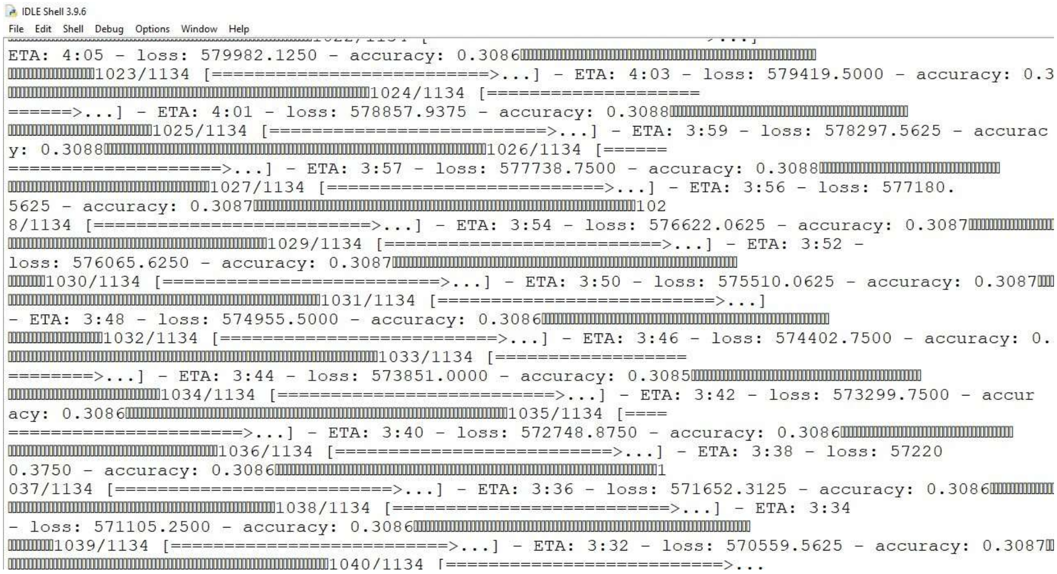


Figure 5: F-score Description

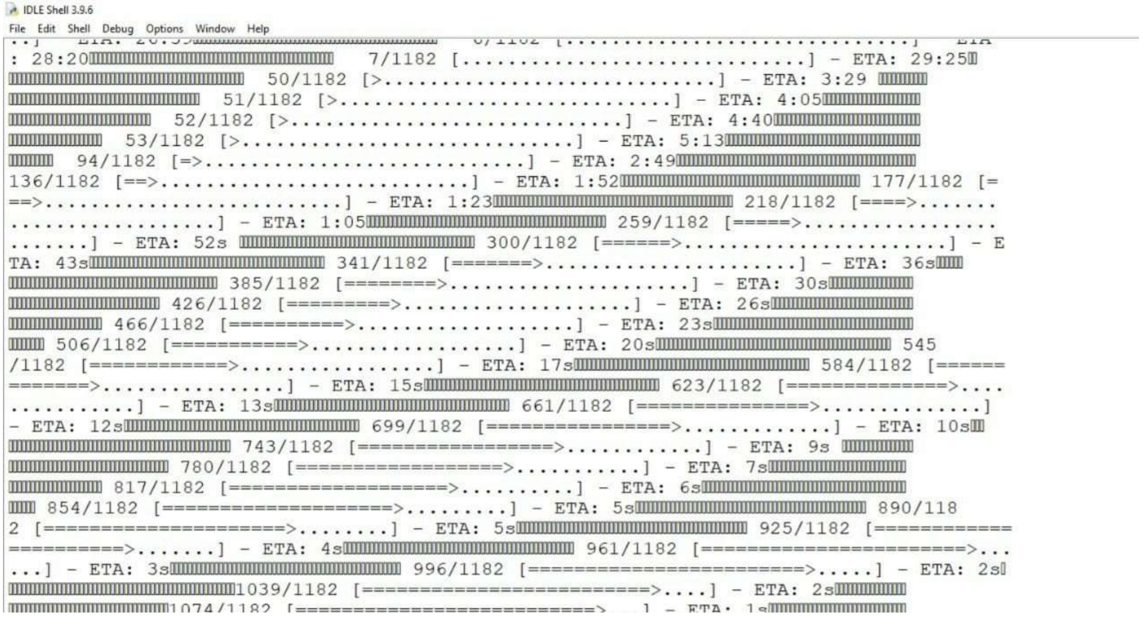
4.3. Screenshots

Training Process:-



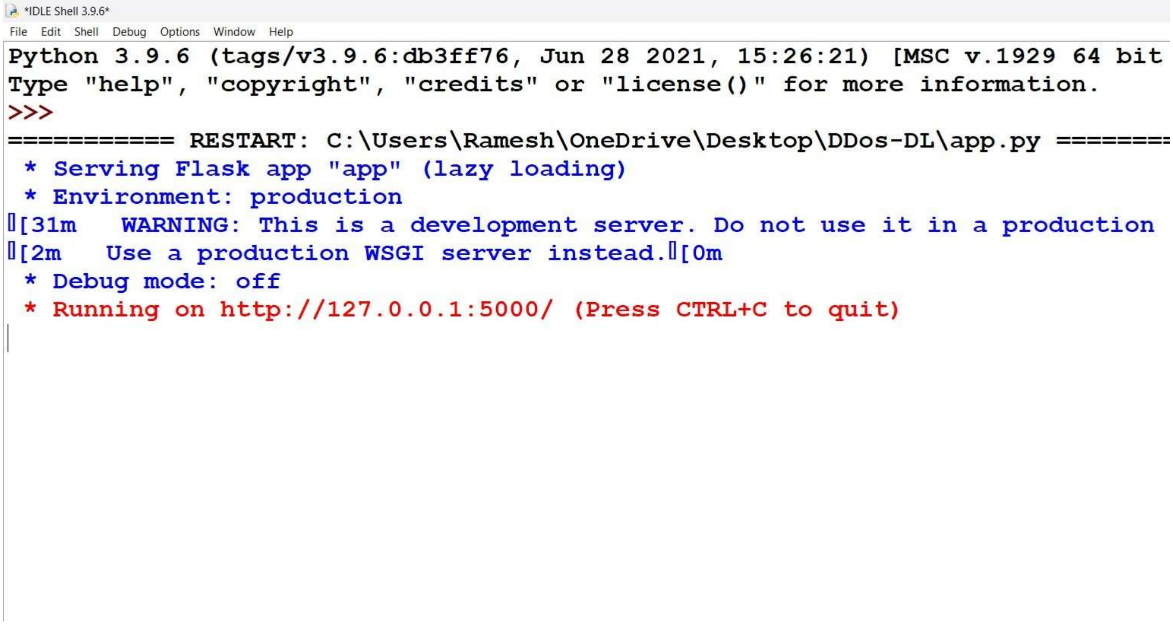
```
ETA: 4:05 - loss: 579982.1250 - accuracy: 0.3086
1023/1134 [==========================>...] - ETA: 4:03 - loss: 579419.5000 - accuracy: 0.3
1024/1134 [====================
======>...] - ETA: 4:01 - loss: 578857.9375 - accuracy: 0.3088
1025/1134 [==========================>...] - ETA: 3:59 - loss: 578297.5625 - accurac
y: 0.30881026/1134 [======
====================>...] - ETA: 3:57 - loss: 577738.7500 - accuracy: 0.3088
1027/1134 [==========================>...] - ETA: 3:56 - loss: 577180.
5625 - accuracy: 0.3087102
8/1134 [==========================>...] - ETA: 3:54 - loss: 576622.0625 - accuracy: 0.3087
1029/1134 [==========================>...] - ETA: 3:52 -
loss: 576065.6250 - accuracy: 0.3087
1030/1134 [==========================>...] - ETA: 3:50 - loss: 575510.0625 - accuracy: 0.3087
1031/1134 [==========================>...]
- ETA: 3:48 - loss: 574955.5000 - accuracy: 0.3086
1032/1134 [==========================>...] - ETA: 3:46 - loss: 574402.7500 - accuracy: 0.
1033/1134 [==================
========>...] - ETA: 3:44 - loss: 573851.0000 - accuracy: 0.3085
1034/1134 [==========================>...] - ETA: 3:42 - loss: 573299.7500 - accur
acy: 0.30861035/1134 [====
======================>...] - ETA: 3:40 - loss: 572748.8750 - accuracy: 0.3086
1036/1134 [==========================>...] - ETA: 3:38 - loss: 57220
0.3750 - accuracy: 0.30861
037/1134 [==========================>...] - ETA: 3:36 - loss: 571652.3125 - accuracy: 0.3086
1038/1134 [==========================>...] - ETA: 3:34
- loss: 571105.2500 - accuracy: 0.3086
1039/1134 [==========================>...] - ETA: 3:32 - loss: 570559.5625 - accuracy: 0.3087
1040/1134 [==========================>...
```

Accuracy:-



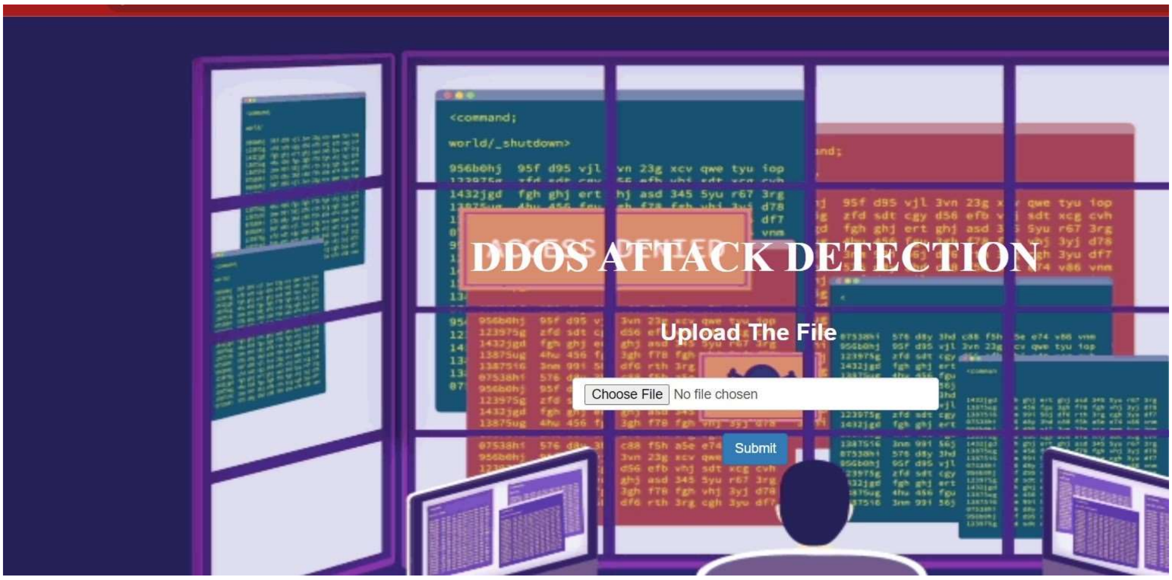
```
..] - ETA: 26:39▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯   6/1182 [..............................] - ETA
: 28:20▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯   7/1182 [..............................] - ETA: 29:25▯
▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯  50/1182 [>.............................] - ETA: 3:29 ▯▯▯▯▯▯▯▯
▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯  51/1182 [>.............................] - ETA: 4:05▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯  52/1182 [>.............................] - ETA: 4:40▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯  53/1182 [>.............................] - ETA: 5:13▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
▯▯▯▯▯▯▯▯  94/1182 [=>............................] - ETA: 2:49▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
136/1182 [==>...........................] - ETA: 1:52▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯ 177/1182 [=
==>...........................] - ETA: 1:23▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯ 218/1182 [====>.......
...................] - ETA: 1:05▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯ 259/1182 [=====>.................
.......] - ETA: 52s ▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯ 300/1182 [======>.......................] - E
TA: 43s▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯ 341/1182 [=======>......................] - ETA: 36s▯▯▯▯▯
▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯ 385/1182 [========>.....................] - ETA: 30s▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯ 426/1182 [=========>....................] - ETA: 26s▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯ 466/1182 [==========>...................] - ETA: 23s▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
▯▯▯▯▯▯ 506/1182 [===========>..................] - ETA: 20s▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯ 545
/1182 [============>.................] - ETA: 17s▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯ 584/1182 [======
=======>................] - ETA: 15s▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯ 623/1182 [==============>....
...........] - ETA: 13s▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯ 661/1182 [===============>..............]
- ETA: 12s▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯ 699/1182 [================>.............] - ETA: 10s▯▯
▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯ 743/1182 [=================>............] - ETA: 9s ▯▯▯▯▯▯▯▯▯▯▯▯▯▯
▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯ 780/1182 [==================>...........] - ETA: 7s▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯ 817/1182 [===================>..........] - ETA: 6s▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
▯▯▯▯ 854/1182 [====================>.........] - ETA: 5s▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯ 890/118
2 [=====================>........] - ETA: 5s▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯ 925/1182 [============
==========>.......] - ETA: 4s▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯ 961/1182 [=======================>...
...] - ETA: 3s▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯ 996/1182 [========================>.....] - ETA: 2s▯
▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯1039/1182 [=========================>....] - ETA: 2s▯▯▯▯▯▯▯▯▯▯▯▯▯▯
▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯1074/1182 [==========================>...] - ETA: 1s▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
```

Web application link generation: -



```
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=========== RESTART: C:\Users\Ramesh\OneDrive\Desktop\DDos-DL\app.py ========
 * Serving Flask app "app" (lazy loading)
 * Environment: production
▯[31m   WARNING: This is a development server. Do not use it in a production
▯[2m   Use a production WSGI server instead.▯[0m
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

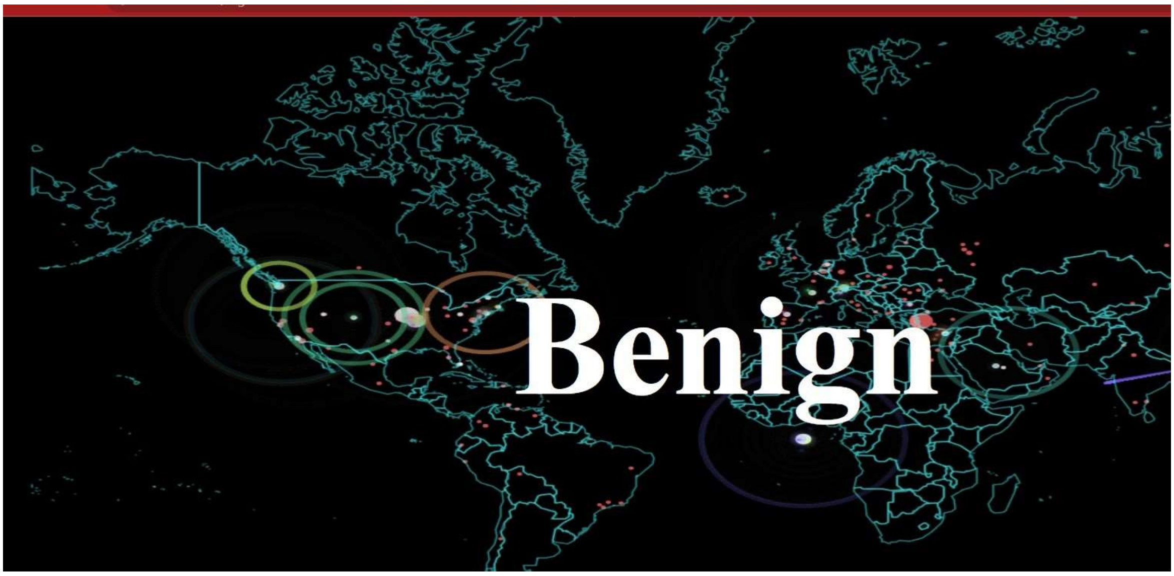
Webpage:



DDOS ATTACK DETECTION

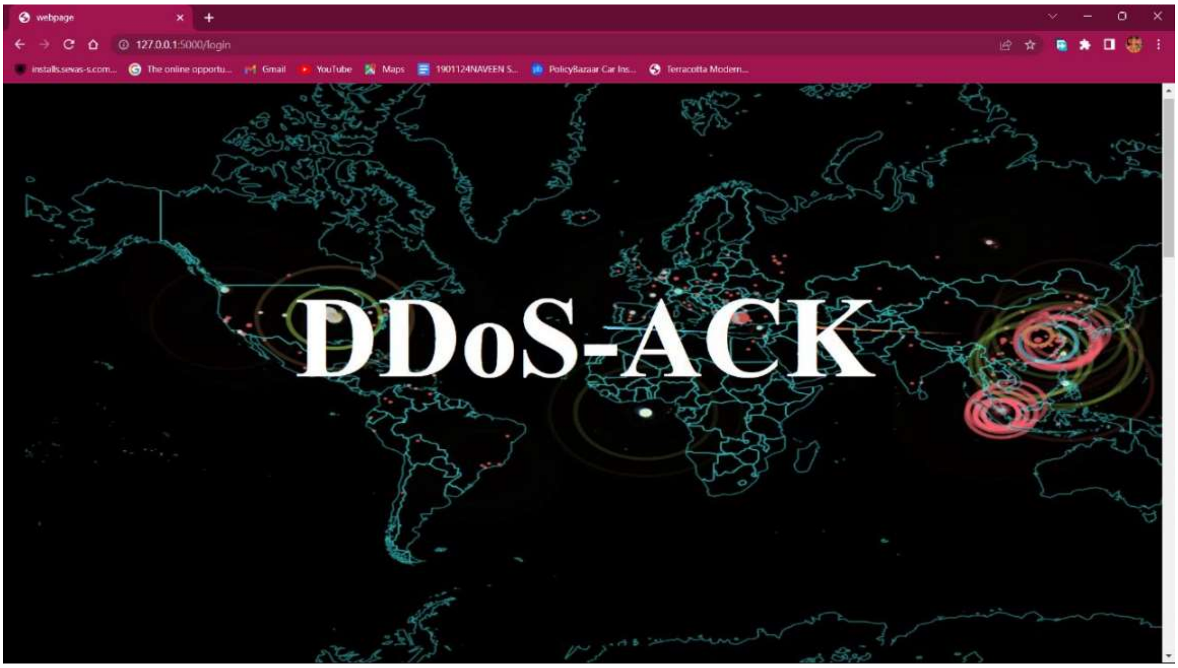
Upload The File

Output 1:

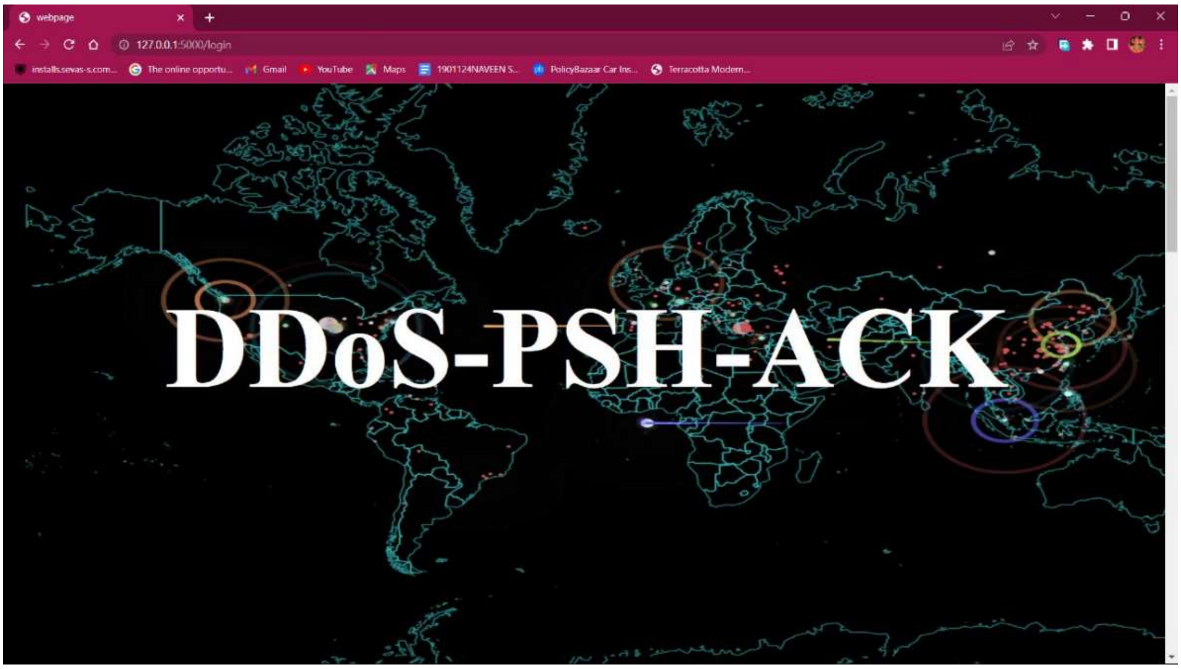


Benign

Output 2:



Output 3:



CHAPTER 5

CONCLUSION AND FUTURE SCOPE

Conclusion

Distributed Denial of Service (DDoS) attacks are a type of cyber attack that targeta network or a website by overwhelming it with traffic from multiple sources. DDoS attacks are a serious threat to organizations and can cause significant damage, including downtime, data theft, and financial loss. Deep learning-based classification and prediction can be used to detect and mitigate DDoS attacks. Inthis project, we were able to successfully implement the proposed system, which uses Convolutional Neural Network. This project predicts three types of attacks namely, DDoS-PSH-ACK, DDoS-ACK, and Benign. The final results will be shown in a Web Application made with HTML and CSS.

Future scope:

- Integrating the model into a real-time system for detecting DDoS attacks in network traffic.
- Developing a user-friendly interface for the model to make it more accessible to users.
- Conducting further research on the impact of DDoS attacks on different industries and using the model to develop strategies for mitigating such attacks.

APPENDIX

DATA PROCESSING (USING CNN):

```
import pandas as pd

import numpy as np

from sklearn.metrics import accuracy_score

from sklearn.model_selection import train_test_split

from sklearn.feature_selection import SelectKBest

from sklearn.feature_selection import chi2

from sklearn.preprocessing import MinMaxScaler

from sklearn.preprocessing import LabelEncoder

from tensorflow import keras

import keras

from keras.models import Sequential

from keras.layers import Conv1D,MaxPooling1D ,MaxPooling2D, Dropout, GlobalAveragePooling2D, Activation

from keras.layers import Flatten, Dense

from keras import optimizers

from keras.callbacks import ModelCheckpoint, History

from matplotlib import pyplot as plt

from sklearn.neural_network import MLPClassifier

import sklearn.metrics as metrics
```

```
  from sklearn.metrics import accuracy_score
  import pickle
  import warnings
  warnings.filterwarnings('ignore')
  a=pd.read_csv("APA-DDoS-Dataset.csv")
  print(a)
  le=LabelEncoder()
  a['frame.time'] = le.fit_transform(a['frame.time'])
  a['ip.dst'] = le.fit_transform(a['ip.dst'])
  a['ip.src'] = le.fit_transform(a['ip.src'])
  a['Label'] = le.fit_transform(a['Label'])
  X=a.drop(['Label'],axis=1)
  print(X)
  Y=a['Label']
  print(Y)
  x_train,x_test,y_train,y_test = train_test_split(X,Y,shuffle=True,test_size=0.25,
  random_state=0)
from tensorflow.keras.utils import to_categorical
  y_train_binary = to_categorical(y_train)
  y_test_binary = to_categorical(y_test)
  print(y_test_binary)
  epochs = 10
```

```
batch_size = 100

model = Sequential()

print(x_train.shape[1],1)

model.add(Conv1D(16, 3, padding='same', activation='relu',
input_shape=(22,1)))#x.shape[1:])) # Input shape: (96, 96, 1)

model.add(MaxPooling1D(pool_size=1))

print(1)

model.add(Conv1D(32, 3, padding='same', activation='relu'))

model.add(MaxPooling1D(pool_size=1))

model.add(Dropout(0.25))

print(1)

model.add(Conv1D(64, 3, padding='same', activation='relu'))

model.add(MaxPooling1D(pool_size=1))

print(1)

model.add(Conv1D(128, 3, padding='same', activation='relu'))

model.add(MaxPooling1D(pool_size=1))

print(1)

model.add(Conv1D(256, 3, padding='same', activation='relu'))

model.add(MaxPooling1D(pool_size=1))

print(1)

model.add(Flatten())

print(1)
```

```
model.add(Dense(512, activation='relu'))
model.add(Dropout(0.2))
print(1)
model.add(Dense(3))
hist = History()
model.compile(optimizer='adam', loss='mean_squared_error', metrics=['accuracy'])
history=model.fit(x_train, y_train_binary,batch_size=batch_size,
epochs=epochs,verbose=1,validation_data=(x_test, y_test_binary))
model.summary()
y_pred=np.argmax(model.predict(x_test), axis=-1)
print("Accuracy",accuracy_score(y_test,y_pred))
filename = 'model5.pkl'
pickle.dump(clf, open(filename, 'wb'))
```

WEBSITE HOSTING IN FLASK

```
from flask import *
import pickle
import pandas as pd
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import LabelEncoder

app = Flask(__name__)

filename = 'model.pkl'
classifier = pickle.load(open(filename, 'rb'))
```

```
@app.route("/")
def home():
   return render_template("browser1.html")
@app.route('/login',methods = ['POST'])
def login():
     uname=request.form['files']
     rr=pd.read_csv(uname)
     le=LabelEncoder()


     rr['ip.dst'] = le.fit_transform(rr['ip.dst'])
     rr['ip.src'] = le.fit_transform(rr['ip.src'])


     type(rr)
     y_pre=classifier.predict(rr)
     if y_pre[0]==0:
        return render_template('index1.html')
     elif y_pre[0]==1:
        return render_template('index2.html')
     elif y_pre[0]==2:
        return render_template('index3.html')

if__name__== '__main__':
   app.run()
```

REFFERENCE

[1] Ahmed A, Haleem M, Hussain H, Khan AA, Mohmand MI, Raza M, Rahman IU, Ullah U, Zakarya M. A machine learning-based classification and prediction technique for DDoS attacks. IEEE Access. 2022 Feb 17;10:21443-54.

[2] Ahmed B, Farhan MSS, Hossain MS. A novel approach for DDoS attack detection using neural networks. In 2019 22nd International Conference on Computer and Information Technology (ICCIT) 2019 Dec 18 (pp. 1-6). IEEE.

[3] Alduailij M, Khan QW, Malik F, Sardaraz M, Tahir M. Machine-Learning-Based DDoS Attack Detection Using Mutual Information and Random Forest Feature Importance Method. Symmetry. 2022 May 27;14(6):1095.

[4] El Khayat R, Refaei A, El Sherif A, Al-Lawati A, Al-Riyami M. DDoS detection in the cloud using machine learning techniques. In 2019 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC) 2019 Aug 21 (pp. 218-223). IEEE.

[5] Farhan MSS, Hossain MS. A comparative study of machine learning algorithms for DDoS attack detection. In 2019 4th International Conference on Electrical and Electronics Engineering (ICEEE) 2019 Dec 19 (pp. 1-5). IEEE.

[6] Han J, Kim D, Lee H, Jang Y. Feature selection for DDoS attack detection using machine learning techniques. Journal of Ambient Intelligence and Humanized Computing. 2020 May 1;11(5):1879-86.

[7] Jang J, Lee M, Kim K. Anomaly-based detection of DDoS attacks using SVM and random forest. Wireless Networks. 2022 Mar 1;28(3):1323-32.

[8] Javed HM, Anjum MA, Hayat I, Khurshid K. A hybrid DDoS attack detection method using clustering and support vector machine. Wireless Personal Communications. 2022 Jul;130(1):193-207.

[9] Kumar J, Kumar S. DDoS attack detection and classification using random forest algorithm. In 2019 IEEE International Conference on Computational Intelligence and Knowledge Economy (ICCIKE) 2019 Sep 6 (pp. 272-277). IEEE.

[10] Lashkari AR, Mousavi SA, Hashemi S, Dehghantanha A, Choo KR. DDoS attack detection using deep convolutional neural networks. Computers & Security. 2022 May 1;109:102340.

[11] Mittal M, Behal S, Kumar K. Deep learning approaches for detecting DDoS attacks: A systematic review. Soft Computing. 2022 Jan 27:1-37.

[12] Pei J, Chen Y, Ji W. A DDoS attack detection method based on machine learning. In Journal of Physics: Conference Series 2019 Jun 1 (Vol. 1237, No. 3, p. 032040). IOP Publishing.

[13] Rezvy S, Luo Y, Petridis M, Lasebae A, Zebin T. An efficient deep learning model for intrusion classification and prediction in 5G and IoT networks. In Proceedings of the 15th International Conference on Emerging Security Information,

Systems and Technologies 2021 Sep 12 (pp. 8-15).

[14] Shi J, Gao J, Zhang H, Yan H, Zhai G. DDoS attack detection using machine learning algorithms. Journal of Intelligent & Fuzzy Systems. 2021 Mar 1;40(3):4767-80.

[15] Singh B, Kumar A. DDoS attack detection using a machine learning approach. International Journal of Advanced Computer Science and Applications. 2021 Mar;12(2):267-74.

[16] Wang Z, Wu Y, Zhang M, Li W, Yao Y. A DDoS attack detection method based on convolutional neural network. In 2018 3rd IEEE International Conference on Computer and Communication Systems (ICCCS) 2018 May 27 (pp. 159-163). IEEE.

[17] Wu J, Xue L, Jiang J. A DDoS attack detection method based on machine learning algorithm. In 2020 IEEE International Conference on Intelligence and Security Informatics (ISI) 2020 Nov 18 (pp. 1-6). IEEE.

[18] Yan Z, Lu L, Wu W, Li B, Zhang S, Yang X, Wu S. DDoS attack detection method based on random forest algorithm. Journal of Ambient Intelligence and Humanized Computing. 2021 Oct 1;12(10):10523-32.

[19] Yang J, Fan J, Cheng Y, Dong H. An improved detection method for DDoS attacks based on machine learning. In Proceedings of the 2021 3rd International Conference on Communication Engineering and Technology (ICCET 2021) 2021 May 16 (pp. 129-136). Atlantis Press.

[20] Zulqarnain H, Lee WJ, Lee YK. A hybrid machine learning approach for DDoS attack detection. Future Generation Computer Systems. 2021 Nov 1;124:484-94