VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Detection of DDoS Attacks Using Deep Learning Algorithms

A PROJECT REPORT
For

CSE3502 Information Security Management

In

B. Tech (IT)

By

Aman Singh (20BIT0124)
Chavan Mukul Manish (20BIT0238)
Shriyansh Sinha (20BIT0353)
Shah Jatin Manoj (20BIT0387)
Rohil Saxena (20BIT0404)

Winter Semester – 2022~23

Under the Guidance of
Dr. SUMAIYA THASEEN

School of Information Technology and Engineering

1. Abstract

An attack known as a DDoS, or Distributed Denial of Service, is an assault in which the attacker floods a server with internet traffic in an effort to prevent the server from accessing any requests made by its customers. Because each company requires its own server to maintain the confidentiality of its data in the modern day, the frequency of distributed denial of service assaults has significantly increased. Since the organisations are in the process of developing their servers, the attacker is able to target their server, which may have relatively low request restrictions. As a consequence, the adversaries will be able to bring down the server in just a few minutes' time.

Therefore, in order to protect the servers from DDoS assaults, we will be utilising two cutting-edge technologies that are seeing explosive growth at the present time: machine learning (ML) and deep neural networks (DNN). Therefore, by utilising the logs of the server, we can train a machine learning model to refuse a request coming from an IP address that is attempting to make a DDoS assault on the server. This will protect the server from being subjected to a DDoS attack. In order to classify them, we will use techniques such as Naive Bayes, Random Forest, MLP, and Decision tree, amongst others.

2. Keywords

Distributed Denial of Service, attacker, server, internet traffic, customers, confidentiality, data, organisations, request restrictions, adversaries, cutting-edge technologies, machine learning, ML, deep neural networks, Naive Bayes, Random Forest, MLP, Decision tree

3. Introduction

Distributed Denial of Service (DDoS) attacks are a major threat to internet security, as they can disrupt the operations of servers and websites, causing significant financial losses. Traditional methods of DDoS detection involve monitoring network traffic and identifying patterns that indicate an attack is occurring. However, these methods are often ineffective against sophisticated attacks that can mimic legitimate traffic.

In recent years, deep learning algorithms have emerged as a promising approach to detecting DDoS attacks. These algorithms can learn complex patterns in network traffic data and identify anomalies that may indicate an attack. By using deep learning techniques, it is possible to develop models that can detect even previously unseen types of DDoS attacks, making them a valuable addition to traditional detection methods.

This paper explores the use of deep learning algorithms for detecting DDoS attacks. Specifically, we investigate the effectiveness of various deep learning architectures, including Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), in identifying different types of DDoS attacks. We also examine the impact of different training strategies and input features on the performance of these models.

The results of our study demonstrate the potential of deep learning algorithms for detecting DDoS attacks. We show that these algorithms can achieve high levels of accuracy in identifying both known and unknown types of attacks, and that their performance can be improved through careful selection of training strategies and input features. We conclude that deep learning represents a promising approach to DDoS detection, and that it is likely to become an increasingly important tool in the fight against cybercrime.

Objectives:

- The literature Review help us understand already existing similar ideas and what new approaches we can use ourselves.
- Our methodology is to make sure we find the best Deep Learning Algorithm For DDOS detection

D. Literature Review

1.) Prevention Techniques against Distributed Denial of Service Attacks in Heterogeneous Networks: A Systematic Review

Ammarah Cheema, Moeenuddin Tariq, Adnan Hafiz, Muhammad Murad Khan,Fahad Ahmad and Muhammad Anwar

One of the most crucial concerns in network security is the Distributed Denial of Service (DDoS) attack. By deploying a large number of infected computers to overload the bandwidth or network service, these types of assaults significantly increase the risk that genuine users won't be able to access network services. Genuine users are prevented from accessing the service because the targeted servers are overloaded with malicious packets or connection requests, which slows down or even crashes the server operations.

In this research, it is classified DDoS assaults precisely and outlined the causes, effects, and motives of the attackers. DDoS attacks against Internet of Things (IoT) devices are further expanded based on application and network levels. Modern defence strategies to thwart such attacks have been the subject of an extensive literature analysis. To identify the best answers, each mechanism has undergone a thorough investigation. They have objectively assessed the available DDoS attack protection strategies and summarised our significant results in comparison tables. This publication also offers suggestions for further research for novice researchers.

Conclusion -

(I)The victim will suffer financial loss since they won't be able to use services while the assault is happening.

(ii) Negative effect on the company's future: Customers may lose trust in the target if it appears to have security weaknesses.

(iii) A legal possibility would exist if user information had been compromised or the target had not met service-level agreements as a result of the attack.

In this survey report, they have looked at state-of-the-art defence strategies that are being used right now to swiftly fight off DDoS assaults and limit the harm done to the targeted system and its legitimate users. To determine which protection method was the most effective, they conducted a thorough examination of the available methods for preventing DDoS attacks. Additionally, the specialised protection methods for IoT and SDN devices are thoroughly explained. Future scholars will find this paper useful in learning more about the many DDoS attack types and effective security strategies for identifying, reducing, and preventing them.

Link- https://www.hindawi.com/journals/scn/2022/8379532/

2.) Distributed Denial of Service Attacks on Cloud Computing Environment: A Comprehensive Review
By Israa T. Aziz, Ihsan H. Abdulqadder, Thakwan A. Jawad

Recent years have seen a fast advancement in cloud computing, which is changing how the information technology (IT) sector operates. Cloud computing environments (CCE) are vulnerable to a variety of predatory attacks, with distributed denial of service (DDoS) being a major one. DDoS attacks, like those against Cloudflare and Spanhaus, are alarmingly and frequently used to exploit sample network management protocols because of their apparent transparency, large volumes of stored data, and comparable ease of operability (SNMP). Flooding, spoofing, user-to-root, port scanning, large XML, forceful parsing, reflection assaults, and other important DDoS attacks are only a few examples.
Cloud computing has rapidly advanced in recent years, transforming how the information technology (IT) industry functions. Distributed denial of service (DDoS) is one of the main predatory assaults that cloud computing environments (CCE) are susceptible to. Because of their apparent openness, massive amounts of stored data, and relative simplicity of operation, DDoS operations, like those against Cloudflare and Spanhaus, are disturbingly and routinely used to target sample network management technologies (SNMP). Examples of significant DDoS attacks include flooding, spoofing, user-to-root, port scanning, huge XML, forced parsing, reflection attacks, and others.
In this paper, discussion of various studies carried out concerning CCEs is presented with more emphasis is given on the studies about DDoS is done.

Conclusion –

Typically, DDoS attacks overload the victim resources by flooding servers, systems, or networks with traffic, making it impossible for authorised users to utilise the resources. Furthermore, the attacks are scattered throughout a number of attack systems, making it difficult to detect them and defend against them. IRC botnets are a serious hazard since they have developed further, increasing the risk as they amass armies of bots for massive attacks. The main obstacles to successful DDoS mitigation methods are twofold:
(1) To initiate DDoS flooding assaults, a huge number of zombies are used, and
(2) A zombie's IP address is often faked under the attacker's control.
The goal of this effort is to develop novel optimization methods to counter DDoS assaults in the future by utilising a software-defined network (SDN) and networking function virtualization (NFV) in a cloud context.

Link - https://journals.cihanuniversity.edu.iq/index.php/cuesj/article/view/535

3.) Man-in-the-middle and denial of service attacks detection using machine learning algorithms
By Sura Abdulmunem Mohammed Al-Juboori, Firas Hazzaa, Zinah Sattar Jabbar, Sinan Salih, Hassan Muwafaq Gheni
Man-in-the-middle (MTM) and denial of service (DoS) attacks on networks enable many attackers to access and steal crucial data from physically linked devices in any network. By gathering relevant datasets from the Kaggle website for MTM and DoS attacks, the research is applied numerous machine learning algorithms to thwart these attacks and safeguard the devices. Due to the large number of null values in the dataset, preprocessing approaches including filling in the missing values were used in this study. Then, they employed four machine learning algorithms—random forest (RF), eXtreme gradient boosting (XGBoost), gradient boosting (GB), and decision tree—to identify these assaults (DT).
Precision, accuracy, recall, and f1-score are just a few of the classification measures that are used to rate algorithm performance. The study's findings in both datasets are as follows: Both the MTM and DoS attacks can be detected with the same performance from all algorithms, which is greater than 99% in all metrics for the MTM attack and greater than 97% in all metrics for the DoS assault. Results demonstrated that these algorithms are quite successful at spotting MTM and DoS assaults, which has prompted them to put them to use in defending devices against these threats.
Conclusion-
By getting relevant datasets from the Kaggle website, they created four machine learning algorithms in this study to identify two well-known assaults that target the linked devices in any network.
i)A DoS attack can be detected by all algorithms with a performance of at least 97% in all metrics and
ii) An MTM assault can be detected by all algorithms with a performance of at least 99% in all measurements.
Therefore, we can use these four methods to identify MTM and DoS assaults effectively for both datasets. As a result, we can utilise our devices to be protected from these attacks. They also intend to gather datasets pertaining to more assaults and employ different machine learning methods in further research. Additionally, they will use all cutting-edge models, pre-trained models, and deep learning algorithms on future datasets.

Link - https://www.beei.org/index.php/EEI/article/view/4555

4) Detection of DDoS attacks with feed forward based deep neural network model
By: Cil, A.E.; Yildiz, K.; Buldu, A.
2021
As a result of the increase in the services provided over the internet, it is seen that the network infrastructure is more exposed to cyber attacks. The most widely used of these attacks are Distributed Denial of Service (DDoS) attacks that easily disrupt services. The most important

factor in the fight against DDoS attacks is the early detection and separation of network traffic. In this study, it is suggested to use the deep neural network (DNN) as a deep learning model that detects DDoS attacks on the sample of packets captured from network traffic. DNN model can work quickly and with high accuracy even in small samples because it contains feature extraction and classification processes in its structure and has layers that update itself as it is trained. As a result of the experiments carried out on the CICDDoS2019 dataset containing the current DDoS attack types created in 2019, it was observed that the attacks on network traffic were detected with 99.99% success and the attack types were classified with an accuracy rate of 94.57%. The high accuracy values obtained show that the deep learning model can be used effectively in combating DDoS attacks.
Link: https://www.sciencedirect.com/science/article/pii/S0957417420311647

5)Detection of DDoS Attack and Classification Using a Hybrid Approach
Suman Nandi; Santanu Phadikar; Koushik Majumder
2020
In the area of cloud security, detection of DDoS attack is a challenging task such that legitimate users use the cloud resources properly. So in this paper, detection and classification of the attacking packets and normal packets are done by using various machine learning classifiers. We have selected the most relevant features from NSL KDD dataset using five (Information gain, gain ratio, chi-squared, ReliefF, and symmetrical uncertainty) commonly used feature selection methods. Now from the entire selected feature set, the most important features are selected by applying our hybrid feature selection method. Since all the anomalous instances of the dataset do not belong to DDoS category so we have separated only the DDoS packets from the dataset using the selected features. Finally, the dataset has been prepared and named as KDD DDoS dataset by considering the selected DDoS packets and normal packets. This KDD DDoS dataset has been discretized using discretize tool in weka for getting better performance. Finally, this discretize dataset has been applied on some commonly used (Naive Bayes, Bayes Net, Decision Table, J48 and Random Forest) classifiers for determining the detection rate of the classifiers. 10 fold cross validation has been used here for measuring the robustness of the system. To measure the efficiency of our hybrid feature selection method, we have also applied the same set of classifiers on the NSL KDD dataset, where it gives the best anomaly detection rate of 99.72% and average detection rate 98.47% similarly, we have applied the same set of classifiers on NSL DDoS dataset and obtain the average DDoS detection of 99.01% and the best DDoS detection rate of 99.86%. In order to compare the performance of our proposed hybrid method, we have also applied the existing feature selection methods and measured the detection rate using the same set of classifiers. Finally, we have seen that our hybrid approach for detecting the DDoS attack gives the best detection rate compared to some existing methods.
Link: https://www.researchgate.net/profile/Suman-Nandi-2/publication/340971033_Detection_of_DDoS_Attack_and_Classification_Using_a_Hybrid_Approach/links/624158da8068956f3c539d96/Detection-of-DDoS-Attack-and-Classification-Using-a-Hybrid-Approach.pdf

6) Efficient Detection of DDoS Attacks Using a Hybrid Deep Learning Model with Improved Feature Selection
by Daniyal Alghazzawi ,Omaimah Bamasag ,Hayat Ullah and Muhammad Zubair Asghar
2021
DDoS (Distributed Denial of Service) attacks have now become a serious risk to the integrity and confidentiality of computer networks and systems, which are essential assets in today's world. Detecting DDoS attacks is a difficult task that must be accomplished before any mitigation strategies can be used. The identification of DDoS attacks has already been successfully implemented using machine learning/deep learning (ML/DL). However, due to an inherent limitation of ML/DL frameworks—so-called optimal feature selection—complete accomplishment is likewise out of reach. This is a case in which a machine learning/deep learning-based system does not produce promising results for identifying DDoS attacks. At the moment, existing research on forecasting DDoS attacks has yielded a variety of unexpected predictions utilising machine learning (ML) classifiers and conventional approaches for feature encoding. These previous efforts also made use of deep neural networks to extract features without having to maintain the track of the sequence information. The current work suggests predicting DDoS attacks using a hybrid deep learning (DL) model, namely a CNN with BiLSTM (bidirectional long/short-term memory), in order to effectively anticipate DDoS attacks using benchmark data. By ranking and choosing features that scored the highest in the provided data set, only the most pertinent features were picked. Experiment findings demonstrate that the proposed CNN-BI-LSTM attained an accuracy of up to 94.52 percent using the data set CIC-DDoS2019 during training, testing, and validation.
Link: https://www.mdpi.com/2076-3417/11/24/11634/pdf

7) Detection of DDoS Attack using Machine Learning Algorithms
By C M NalayiniI, Dr. Jeevaa Katiravan
(2022)

DDoS attack is one of the most dangerous and growing attack with monstrous growth of Internet. Botnet is a network of bots works together to target the victim abundantly. It is a complex problem because it involves bots in distributed environment on the Internet and affecting numerous networks. It is a big threat to all cloud related platforms say
like Amazon Cloud servers. In February a very largest DDoS attacks was launched with 1.3 of Terabits per second traffic transfer. The most severe attacks are Traffic attack, Application attack and Volume attack or bandwidth attack. Traffic based attacks targets the victim by sending large volume of TCP and UDP packets via botnet to degrade the performance

of the server to make the targeted network down. In volume-based attack, first the attacker identifies the total bandwidth of the targeted network and sends bulk volume of unwanted data to occupy maximum bandwidth to deny the services of the legitimate user. Another type of attack is mainly targeting specific applications to make application access difficult and unavailable to legitimate users. In traditional Systems we generally use filtering and Threshold based techniques for detection and trace route mechanism for mitigation of DDoS. But identifying unknown attacks is a big task in the traditional methods. Most of the attackers' targets application layer to damage the access severely. Its focus is to affect the normal behavior of the system. Since security is the major challenge in Networking Field, it's very essential to efficient tools and methodology to defend against DDoS attack. Now a day machine leaning is one efficient methodology help us to detect known and unknown attacks accurately with the help many available machine learning algorithms. DDoS attack comes under the classification problem. Since the attack traffic size is relatively larger its crucial to find the best algorithm to detect it accurately, there is need to find a good algorithm. In machine learning, dataset is fed as the input to various models to train and test it efficiently for efficient detection and prediction. DDoS attack detection and prediction can be done by comparing various parameters such as accuracy, precision, recall and false alarm rate. Therefore it is necessary to find best machine learning model to detect DDoS at an earlier stage to avoid major security issues in the network.

Conclusion - Random Forest algorithm is found to be the best among the eight algorithms to detect DDoS attack in the CIC IDS 2017 dataset by applying k-fold cross validation. It gives high accuracy, precision and recall/TPR at the same time gives low FAR (False Alarm Rate). The FAR of the Random Forest which is very low say like 0.05 is comparatively low than other algorithms. It produces more accuracy by reducing the over fitting in all its decision trees.

Link:
https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4173187#:~:text=A%20set%20of%20eight%20supervised,for%20training%20and%20testing%20purpose.

8) Performance evaluation of Botnet DDoS attack detection using machine learning
By Tong Anh Tuan, Hoang Viet Long, Le Hoang Son, Raghvendra Kumar, Ishaani Priyadarshini, Nguyen Thi Kim Son
(2020)

Botnet is regarded as one of the most sophisticated vulnerability threats nowadays. A large portion of network traffic is dominated by Botnets. Botnets are conglomeration of trade PCs (Bots) which are remotely controlled by their originator (BotMaster) under a Command and-Control (C&C) foundation. They are the keys to several Internet assaults like spams, Distributed Denial of Service Attacks (DDoS), rebate distortions, malwares, and phishing. To over the problem of DDoS attack, various machine learning methods typically Support Vector Machine

(SVM), Artificial Neural Network (ANN), Naïve Bayes (NB), Decision Tree (DT), and Unsupervised Learning (USML) (K-means, X-means etc.) were proposed. With the increasing popularity of Machine Learning in the field of Computer Security, it will be a remarkable accomplishment to carry out performance assessment of the machine learning methods given a common platform. This could assist developers in choosing a suitable method for their case studies and assist them in further research. This paper performed an experimental analysis of the machine learning methods for Botnet DDoS attack detection. The evaluation is done on the UNBS-NB 15 and KDD99 which are well-known publicity datasets for Botnet DDoS attack detection. Machine learning methods typically Support Vector Machine (SVM), Artificial Neural Network (ANN), Naïve Bayes (NB), Decision Tree (DT), and Unsupervised Learning (USML) are investigated for Accuracy, False Alarm Rate (FAR), Sensitivity, Specificity, False positive rate (FPR), AUC, and Matthew's correlation coefficient (MCC) of datasets. Performance of KDD99 dataset has been experimentally shown to be better as compared to the UNBS-NB 15 dataset. This validation is significant in computer security and other related fields.

Conclusion - In this paper, the authors have analyzed machine learning algorithms for Botnet DDoS attack detection. The tested algorithms are SVM, ANN, NB, DT, and USML (K-means, X-means, etc.). The evaluation was done on the UNBS-NB 15 and KDD99 datasets, which are well-known publicity for Botnet DDoS attack detection. It has been shown that USML (unsupervised learning) is the best at differentiating between Botnet and normal network traffic in term of Accuracy, False Alarm Rate (FAR), Sensitivity, Specificity, False positive rate (FPR), AUC and MCC. This validation is significant in computer security and other related fields.

Link: https://link.springer.com/article/10.1007/s12065-019-00310-w

9) Detection of Unknown DDoS Attacks with Deep Learning and Gaussian Mixture Model
By Chin-Shiuh Shieh, Wan-Wei Lin, Thanh-Tuan Nguyen, Chi-Hong Chen, Mong-Fong Horng and Denis Miu
(2021)

DDoS (Distributed Denial of Service) attacks have become a pressing threat to the security and integrity of computer networks and information systems, which are indispensable infrastructures of modern times. The detection of DDoS attacks is a challenging issue before any mitigation measures can be taken. ML/DL (Machine Learning/Deep Learning) has been applied to the detection of DDoS attacks with satisfactory achievement. However, full-scale success is still beyond reach due to an inherent problem with ML/DL-based systems—the so-called Open Set Recognition (OSR) problem. This is a problem where an ML/DL-based system fails to deal with new instances not drawn from the distribution model of the training data. This problem is particularly profound in detecting DDoS attacks since DDoS attacks' technology keeps evolving and has changing traffic characteristics. This study investigates the impact of the OSR problem on the detection of DDoS attacks. In response to this problem, the paper proposes a new DDoS detection framework featuring Bi-Directional Long ShortTerm Memory (BI-LSTM), a Gaussian Mixture Model (GMM), and incremental learning. Unknown traffic captured by the GMM are subject to discrimination and labeling by traffic engineers, and then fed back to the framework as additional training samples. Using the data sets CIC-IDS2017 and CIC-DDoS2019 for training, testing, and evaluation, experiment results show that the proposed BI-LSTM-GMM can achieve recall, precision, and accuracy up to 94%. Experiments reveal that the proposed framework can be a promising solution to the detection of unknown DDoS attacks.

Conclusion - This study is a proof of concept for the detection of DDoS attacks with Deep Learning (DL) and a Gaussian Mixture Model (GMM). As a solution to the Open Set Recognition (OSR) problem in DDoS detection, the proposed framework consists of Bi-Directional Long Short-Term Memory (BI-LSTM), GMM, and incremental learning. The Bi-LSTM has demonstrated itself as a practical approach for the discrimination of malicious and legitimate traffic sampled from the distribution of the training data set. The GMM has shown to be an effective measure for the differentiation of novel instances and trained samples. Unknown traffic can be captured by the GMM and labeled by data engineers, and then fed back to the BI-LSTM and the GMM for incremental learning. Both the new traffic and the old traffic can be handled by the updated model correctly and gracefully. The feasibility and effectiveness of the proposed framework has been validated by a series of experiments on data sets CIC-IDS2017 and CIC-DDoS2019.
BI-LSTM is fully capable of performing what it has been trained to do, such as detecting known DDoS attacks. However, when confronted with novel attacks, the system performance degrades severely. The recall drops from 99.8% to 41.2% for Dataset CICIDS2017/Wednesday and Friday. Unknown traffic can be captured by GMM and then labeled by traffic engineers.

Link: https://www.mdpi.com/2076-3417/11/11/5213

10. Application layer DDos Attack Detection using Cuckoo Search Algorithm-Trained Radial Basis Function

H. Beitollahi, D. M. Sharif and M. Fazeli
2022

In order to detect App-DDoS traffic, the objective was to offer a Machine Learning (ML) method that combines the Radial Basis Function (RBF) neural network with the cuckoo search algorithm.

They start with gathering and cleaning training data, then normalising the data, and using the Genetic Algorithm (GA) to choose the best collection of features. The next step is to train an RBF neural network using the cuckoo search optimizer algorithm and the best subset of features. In the end, they evaluated their suggested strategy against the tried-and-true k-nearest neighbor (k-NN), Bootstrap Aggregation (Bagging), Support Vector Machine (SVM), Multi-layer Perceptron (MLP), and (Recurrent Neural Network) RNN approaches.

The conclusion is that it performed better than well-known ML techniques since it had the lowest error rate. The metrics are accuracy =96.9 %, MSE =0.134 , RMSE =0.366 , and MAE =0.067.

Link: https://ieeexplore.ieee.org/document/9795027

11. Machine Learning based DDoS Detection
By S. S. Priya, M. Sivaram, D. Yuvaraj and A. Jayanthiladevi
2020

The objective of this research paper was to build an automated DDoS detection tool which could run on any commodity hardware.
Two features, delta time and packet size, are used to identify DDoS packets from regular packets using three classification algorithms: KNN, Random Forest, and Naïve Bayes. The majority of DDoS attacks, including ICMP floods, TCP floods, UDP floods, and others, can be found by this detector. Particular systems may need a lot of features to detect DDoS, and older systems only identify some forms of DDoS attacks. Some systems might only function with specific protocols. These limitations were overcome in this paper.
The resulting model using only two features resulted in an accuracy of 98.5 percent. It can be easily trained for any type of DDoS attacks. It also does not need a large dataset to detect.

The limitations of this model are that the famous DDoS tool Hping3 was to train the model and it may not detect attacks created by other tools.

Link: https://ieeexplore.ieee.org/abstract/document/9167642

12. Evaluating ML based DDoS Detection with Grid Search Hyperparameter

O. R. Sanchez, M. Repetto, A. Carrega and R. Bolla
2021

In this study, the objective is to use the traffic flow data to evaluate whether a particular flow is connected to a DDoS attack.

A DDoS detector was created using conventional Machine Learning (ML) techniques, and their detection efficiency was enhanced through the use of a thorough hyperparameter search. For resource-constrained situations like the Internet of Things, using lightweight techniques is appropriate to reduce computation overhead. The algorithms of Naïve Bayes, Logistic Regression, KNN, Decision trees, Support Vector Machines, Random Forest etc were used.

The analysis reveals that the majority of algorithms produce acceptable results, with Random Forests obtaining up to 99% detection accuracy, which is comparable to the effectiveness of the most recent deep learning DDoS detection solutions, a requirement for a particular protocol that makes sparse use of features.

For future work, the expansion for the detection of multiclass in order to recognize the attack's nature and use it in a real-world setting.

Link: https://ieeexplore.ieee.org/document/9492633

13) A Machine Learning Approach for DDoS (Distributed Denial of Service) Attack Detection Using Multiple Linear Regression
By - Swathi Sambangi and Lakshmeeswari Gondi
Published: 25 December 2020

The problem of identifying Distributed Denial of Service (DDos) attacks is fundamentally a classification problem in machine learning. In relevance to Cloud Computing, the task of identification of DDoS attacks is a significantly challenging problem because of computational complexity that has to be addressed. Fundamentally, a Denial of Service (DoS) attack is an intentional attack attempted by attackers from single source which has an implicit intention of making an application unavailable to the target stakeholder. For this to be achieved, attackers usually stagger the network bandwidth, halting system resources, thus causing denial of access for legitimate users. Contrary to DoS attacks, in DDoS attacks, the attacker makes use of multiple sources to initiate an attack. DDoS attacks are most common at network, transportation, presentation and application layers of a seven-layer OSI model. In this paper, the research objective is to study the problem of DDoS attack detection in aCloud environment by considering the most popular CICIDS 2017 benchmark dataset and applying multiple regression analysis for building a machine learning model to predict DDoS and Bot attacks through considering a Friday afternoon traffic logfile.

Conclusion-

As detection of DDOS attack has become more common in a distributed environment like Cloud, it is essential to detect the attacks which cause service unavailability of Cloud. To identify such attacks, machine learning models can be used to train and test the attack detection datasets. Alternately, we can use the regression analysis technique by applying one of its important variants known as multiple linear regression analysis. The research objective behind this study is to build a machine learning model that is an ensemble of feature selection using information gain and regression analysis. For experimental study, the dataset considered was in the popularly known CICIDS 2017 dataset. Specifically, the Friday logfile of morning and afternoon are considered which has Benign, Bot and DDoS classes. This paper thus paved a way to show the importance of regression analysis in building an ML model and also shows some of the important visualizations such as residual plots and fit chart which proves the importance of the model and its suitability of considering the model for prediction. In this work, we have limited our analysis for one-day log file and in future, this research may be extended to consider all traffic log files of five days and come out with a consensus-based machine learning model.

Link - https://www.mdpi.com/2504-3900/63/1/51

14) Stability of SDE-LJN System in the Internet to Mitigate Constant-Rate DDoS Attack

By - Kaijiao Huang, Liansheng Tan and Gang Peng

Published: 06 Oct 2021

The Internet is nowadays suffering dramatically serious attacks, with the distributed denial of service (DDoS) attacks being the representative and dominant ones. It is seen that, to stabilize the buffer queue length around a given target under DDoS attacks in the relevant routes is vitally important and helpful to mitigate the attacks and to improve the quality of service (QoS) for normal users. In the current paper, a stochastic queue dynamic model with Levy jump noise, which is affected by the continuous Brownian motion and the discontinuous Poisson process, is worked out to develop a novel and accurate mathematical framework for the stability of a route queue that deals with constant-rate DDoS attacks. This article proposes a security defensive mechanism in the network for solving the network collapse that can possibly be caused by DDoS attacks, otherwise. Particularly, based on the formulation of a stochastic queue dynamic with Levy jump noise, the mechanism that characterizes the behavior of the queue at routers is presented for stabilizing the queue length under constant-rate DDoS attacks. By applying the stochastic control theory into analyzing the performance of queue dynamic under constant-rate DDoS attacks, some explicit conditions are established under which the instantaneous queue length converges to any given target in a route. Simulation results demonstrate the satisfaction of the proposed defense mechanism with sharp contrast to the state-of-the-art active queue management (AQM) schemes.

Future Work-

The new queue model and feedback controller proposed in this paper provide novel ideas and feasible technical solutions for dealing with constant-rate DDoS attacks, thus solving the intractable problems in the field of network security. Unfortunately, the research in this paper only focuses on the type of constant-rate DDoS attack. However, there are many types of DDoS attacks, such as varying-rate. Other types of DDoS attacks can be studied in the future. Developing more interesting researches to deal with other types of DDoS attacks is the scope of our future work. In the future, we will apply the proposed SDE-LJN scheme to the complex environment of reality and further research on the influence of control systems on all kinds of attacked networks' performance. Moreover, since the volume of traffic affects the router CPU utilization, the burst traffic can cause high router CPU utilization. If CPU utilization is consistently very high on the router, it is usually considered to be a problem and needs to be investigated. High router CPU utilization (unavailable) causes the network to crash, which is the result of DDoS attacks.

Therefore, the first extension to our work will monitor the important metric thatis router CPU utilization.

Link - https://www.hindawi.com/journals/scn/2021/4733190/

15) Analysis of Machine Learning Classifiers for Early Detection of DDoS Attacks on IoT Devices

By - Vimal Gaur & Rajneesh Kumar

Published: 08 July 2021

Distributed denial-of-service attacks are still difficult to handle as per current scenarios. The attack aim is a menace to network security and exhausting the target networks with malicious traffic from multiple sites. Although a plethora of conventional methods have been proposed to detect DDoS attacks, so far the rapid diagnosis of these attacks using feature selection algorithms is a daunting challenge. The proposed system uses a hybrid methodology for selecting features by applying feature selection methods on machine learning classifiers. Feature selections methods, namely chi-square, Extra Tree and ANOVA have been applied on four classifiers Random Forest, Decision Tree, k-Nearest Neighbors and XGBoost for early detection of DDoS attacks on IoT devices. We use the CICDDoS2019 dataset containing comprehensive

DDoS attacks to train and assess the proposed methodology in a cloud-based environment (Google Colab). Based on the experimental results, the proposed hybrid methodology provides superior performance with a feature reduction ratio of 82.5% by achieving 98.34% accuracy with ANOVA for XGBoost and helps in early detection of DDoS attacks on IoT devices.

Link - https://link.springer.com/article/10.1007/s13369-021-05947-3

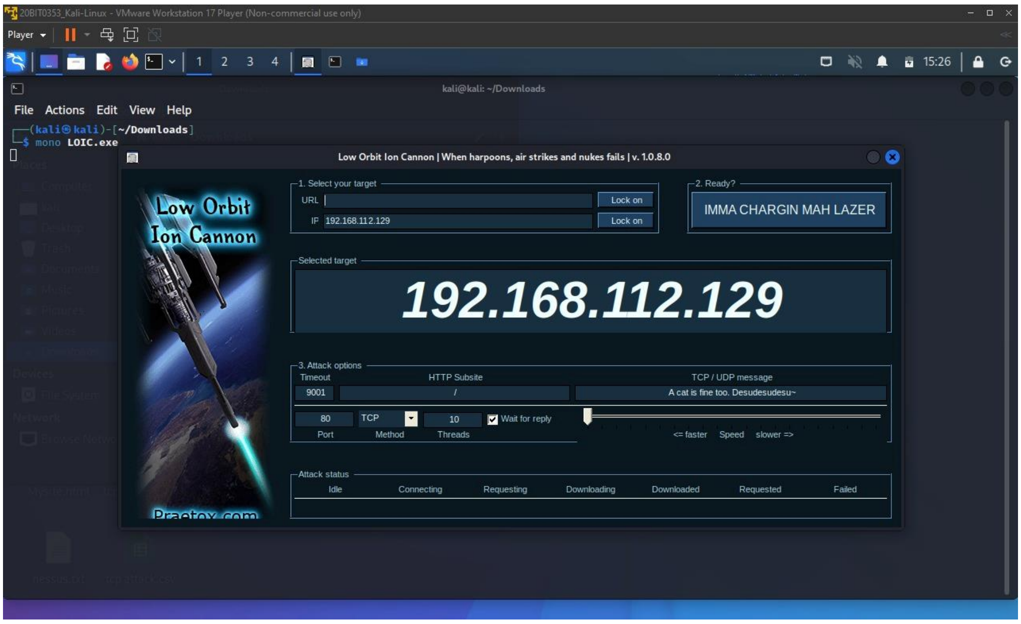
E. Materials and Methods

➢ Code and Dataset

https://drive.google.com/drive/folders/10RkkpANKjm90rXAz3UVHagr7z_V82Xup?usp=sharing

The Dataset is made by LOIC. Low Orbit Ion Cannon (LOIC) is a network stress testing tool that is used to launch Distributed Denial of Service (DDoS) attacks.

LOIC works by sending a large number of HTTP, UDP or TCP packets to a target server, overwhelming its bandwidth and causing it to crash or become unavailable to legitimate users.
The traffic is then captured by Wireshark and saved in a CSV format.


20BIT0353_Kali-Linux - VMware Workstation 17 Player (Non-commercial use only)
Player
1 2 3 4
15:26
kali@kali: ~/Downloads
File Actions Edit View Help
Low Orbit Ion Cannon
Praetox.com
Lock on
Lock on
IMMA CHARGIN MAH LAZER
<= faster Speed slower =>

```
┌──(kali㉿kali)-[~/Downloads]
└─$ mono LOIC.exe
```

Low Orbit Ion Cannon | When harpoons, air strikes and nukes fails | v. 1.0.8.0

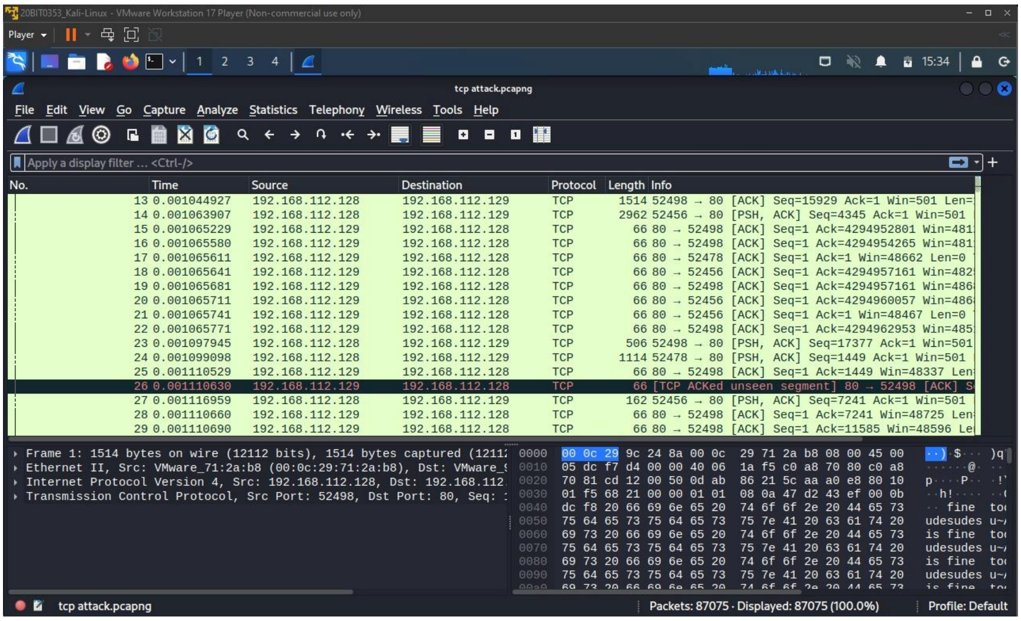
1. Select your target
URL:
IP: 192.168.112.129

2. Ready?

Selected target: 192.168.112.129

3. Attack options
Timeout: 9001
HTTP Subsite: /
TCP / UDP message: A cat is fine too. Desudesudesu~
Port: 80
Method: TCP
Threads: 10
☑ Wait for reply

Attack status
Idle Connecting Requesting Downloading Downloaded Requested Failed


20BIT0353_Kali-Linux - VMware Workstation 17 Player (Non-commercial use only)
Player
1 2 3 4
15:34
tcp attack.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
Profile: Default

```
No.    Time         Source           Destination      Protocol Length Info
    13 0.001044927  192.168.112.128  192.168.112.129  TCP      1514 52498 → 80 [ACK] Seq=15929 Ack=1 Win=501 Len=
    14 0.001063907  192.168.112.128  192.168.112.129  TCP      2962 52456 → 80 [PSH, ACK] Seq=4345 Ack=1 Win=501
    15 0.001065229  192.168.112.129  192.168.112.128  TCP        66 80 → 52498 [ACK] Seq=1 Ack=4294952801 Win=481
    16 0.001065580  192.168.112.129  192.168.112.128  TCP        66 80 → 52498 [ACK] Seq=1 Ack=4294954265 Win=481
    17 0.001065611  192.168.112.129  192.168.112.128  TCP        66 80 → 52478 [ACK] Seq=1 Ack=1 Win=48662 Len=0
    18 0.001065641  192.168.112.129  192.168.112.128  TCP        66 80 → 52456 [ACK] Seq=1 Ack=4294957161 Win=482
    19 0.001065681  192.168.112.129  192.168.112.128  TCP        66 80 → 52498 [ACK] Seq=1 Ack=4294957161 Win=486
    20 0.001065711  192.168.112.129  192.168.112.128  TCP        66 80 → 52456 [ACK] Seq=1 Ack=4294960057 Win=486
    21 0.001065741  192.168.112.129  192.168.112.128  TCP        66 80 → 52456 [ACK] Seq=1 Ack=1 Win=48467 Len=0
    22 0.001065771  192.168.112.129  192.168.112.128  TCP        66 80 → 52498 [ACK] Seq=1 Ack=4294962953 Win=485
    23 0.001097945  192.168.112.128  192.168.112.129  TCP       506 52498 → 80 [PSH, ACK] Seq=17377 Ack=1 Win=501
    24 0.001099098  192.168.112.128  192.168.112.129  TCP      1114 52478 → 80 [PSH, ACK] Seq=1449 Ack=1 Win=501
    25 0.001110529  192.168.112.129  192.168.112.128  TCP        66 80 → 52498 [ACK] Seq=1 Ack=1449 Win=48337 Len
    26 0.001110630  192.168.112.129  192.168.112.128  TCP        66 [TCP ACKed unseen segment] 80 → 52498 [ACK] S
    27 0.001116959  192.168.112.128  192.168.112.129  TCP       162 52456 → 80 [PSH, ACK] Seq=7241 Ack=1 Win=501
    28 0.001110660  192.168.112.129  192.168.112.128  TCP        66 80 → 52498 [ACK] Seq=1 Ack=7241 Win=48725 Len
    29 0.001110690  192.168.112.129  192.168.112.128  TCP        66 80 → 52498 [ACK] Seq=1 Ack=11585 Win=48596 Le
```

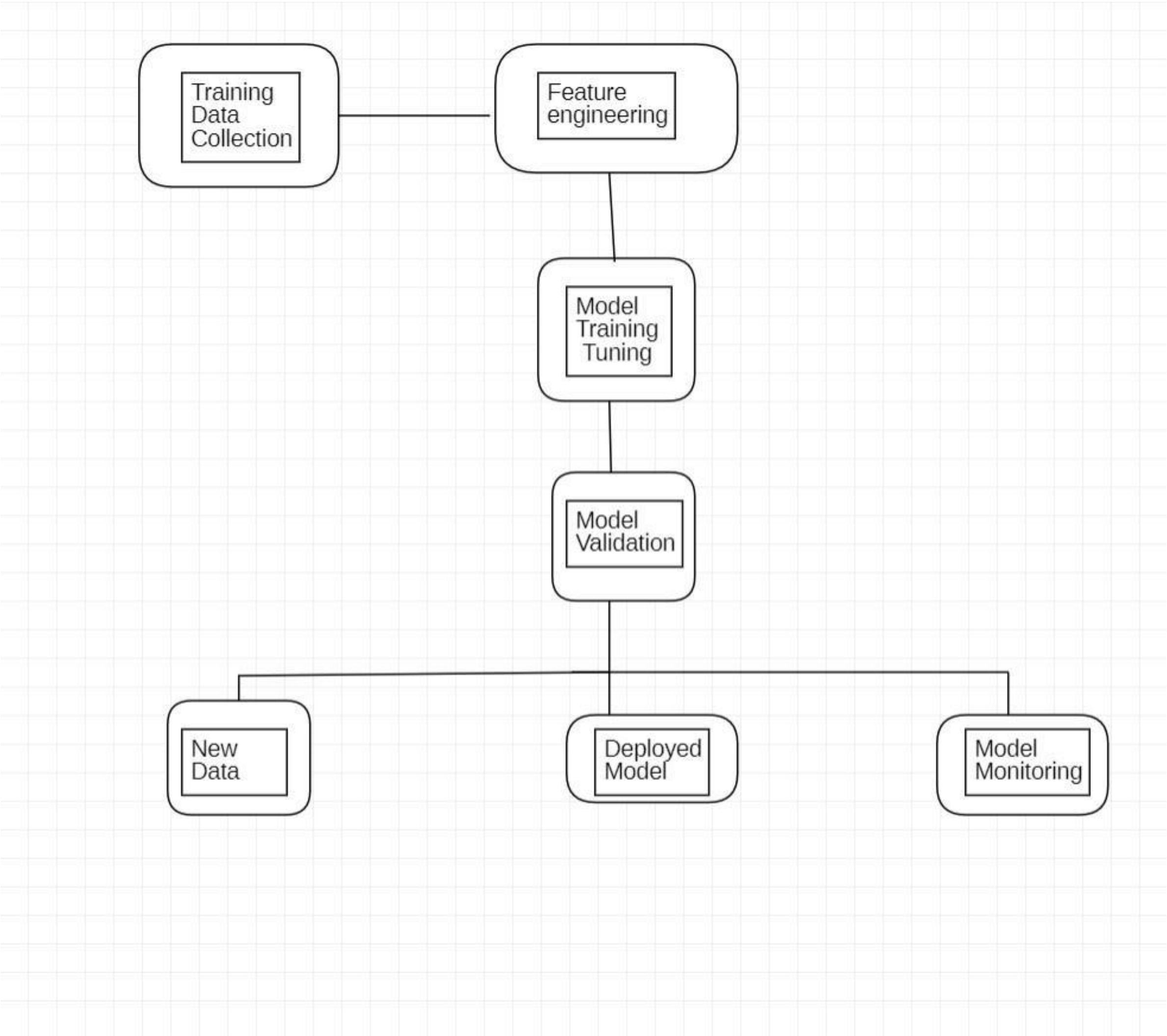
```
› Frame 1: 1514 bytes on wire (12112 bits), 1514 bytes captured (1211
› Ethernet II, Src: VMware_71:2a:b8 (00:0c:29:71:2a:b8), Dst: VMware_
› Internet Protocol Version 4, Src: 192.168.112.128, Dst: 192.168.112
› Transmission Control Protocol, Src Port: 52498, Dst Port: 80, Seq: 
```

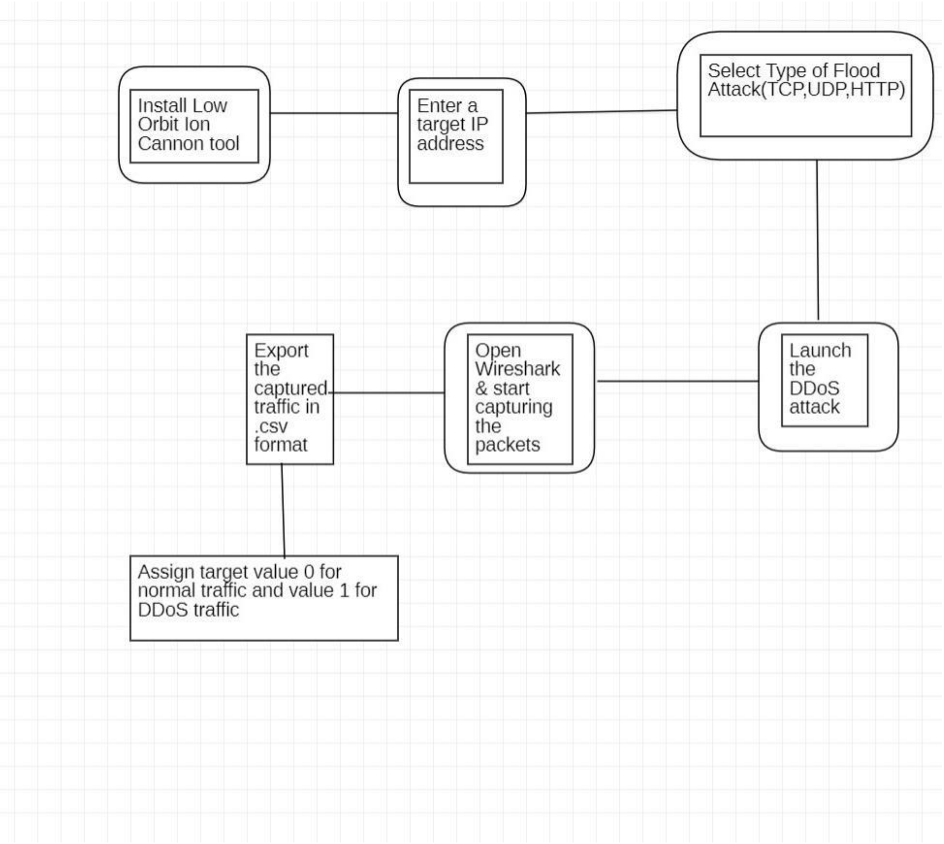
```
0000  00 0c 29 9c 24 8a 00 0c  29 71 2a b8 08 00 45 00   ··)·$···  )q
0010  05 dc f7 d4 00 00 40 06  1a f5 c0 a8 70 80 c0 a8   ······@·  ··
0020  70 81 cd 12 00 50 0d ab  86 21 5c aa a0 e8 80 10   p····P··  ·!
0030  01 f5 68 21 00 00 01 01  08 0a 47 d2 43 ef 00 0b   ··h!····  ··
0040  dc f8 20 66 69 6e 65 20  74 6f 6f 2e 20 44 65 73   ·· fine   to
0050  75 64 65 73 75 64 65 73  75 7e 41 20 63 61 74 20   udesudes  u~
0060  69 73 20 66 69 6e 65 20  74 6f 6f 2e 20 44 65 73   is fine   to
0070  75 64 65 73 75 64 65 73  75 7e 41 20 63 61 74 20   udesudes  u~
0080  69 73 20 66 69 6e 65 20  74 6f 6f 2e 20 44 65 73   is fine   to
0090  75 64 65 73 75 64 65 73  75 7e 41 20 63 61 74 20   udesudes  u~
```

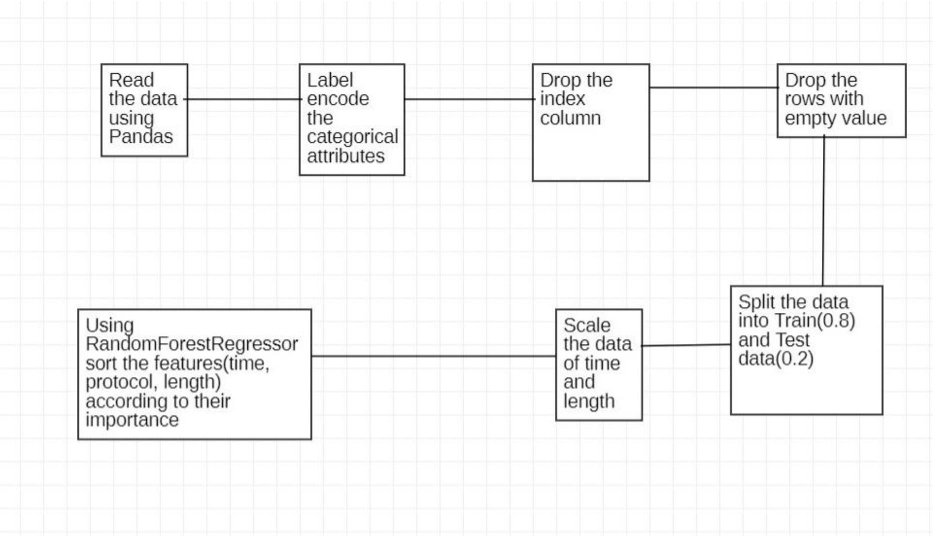
tcp attack.pcapng   Packets: 87075 · Displayed: 87075 (100.0%)

- **High-Level Diagrams & Low-Level Diagrams**

High Level Diagram:

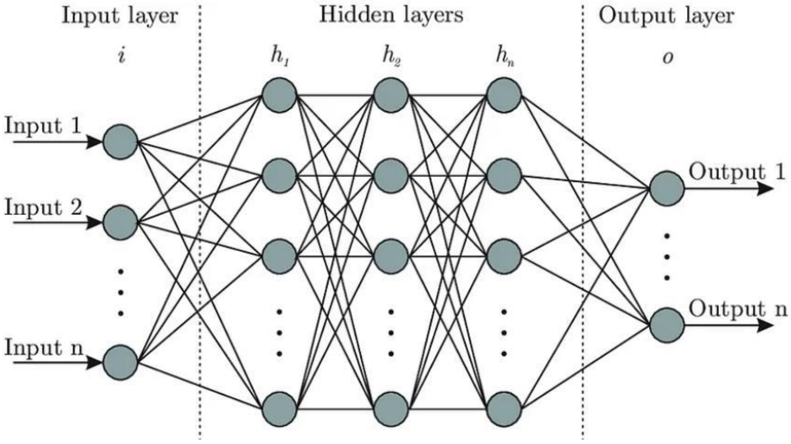



Install Low Orbit Ion Cannon tool
Enter a target IP address
Select Type of Flood Attack(TCP,UDP,HTTP)
Launch the DDoS attack
Open Wireshark & start capturing the packets
Export the captured traffic in .csv format
Assign target value 0 for normal traffic and value 1 for DDoS traffic

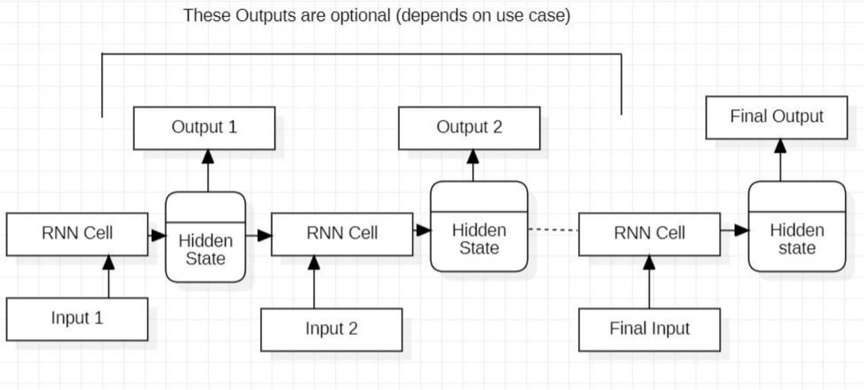

Read the data using Pandas
Label encode the categorical attributes
Drop the index column
Drop the rows with empty value
Split the data into Train(0.8) and Test data(0.2)
Scale the data of time and length
Using RandomForestRegressor sort the features(time, protocol, length) according to their importance

Low Level Diagram:

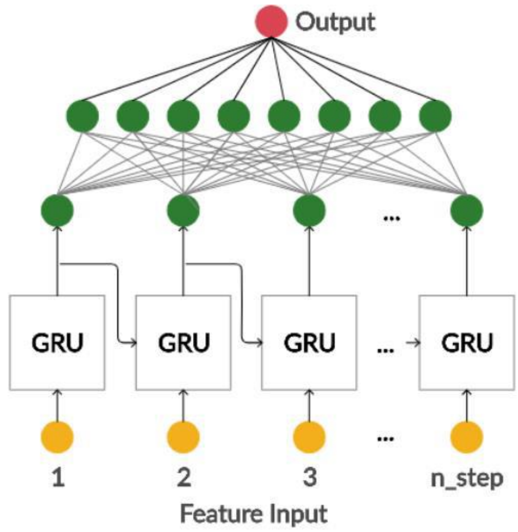
- **ANN**



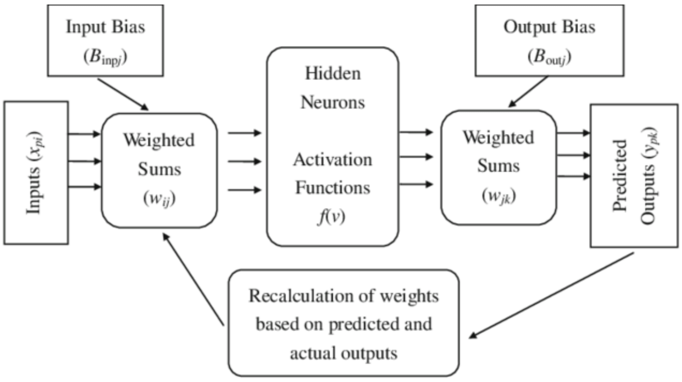
- **RNN**



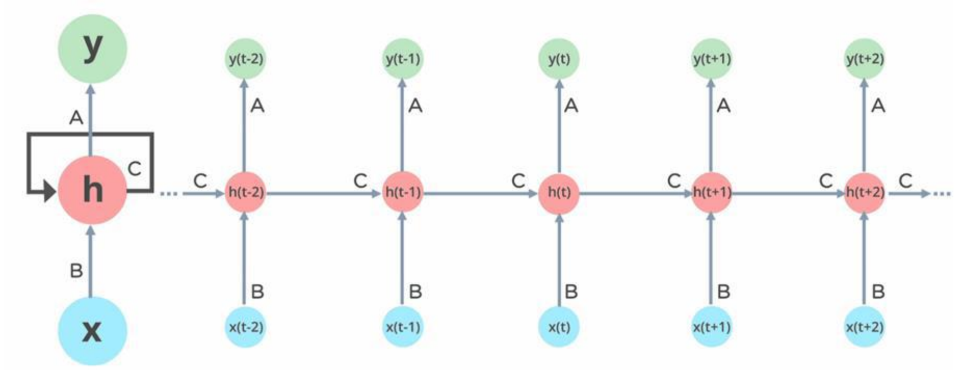
- **GRU**


Output
GRU
GRU
GRU
...
GRU
...
...
1
2
3
n_step
Feature Input

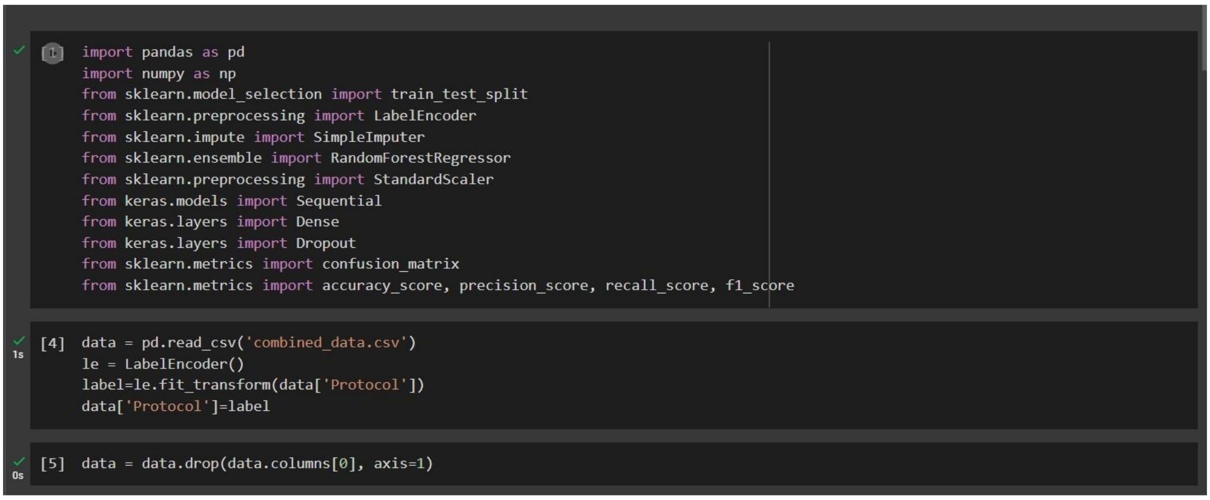
- **MLP**


Input Bias
(B_{inpj})
Inputs (x_{pi})
Weighted
Sums
(w_{ij})
Hidden
Neurons
Activation
Functions
$f(v)$
Output Bias
(B_{outj})
Weighted
Sums
(w_{jk})
Predicted
Outputs (y_{pk})
Recalculation of weights
based on predicted and
actual outputs

- **LSTM**


y
A
h
C
B
x
y(t-2)
y(t-1)
y(t)
y(t+1)
y(t+2)
A
C
h(t-2)
h(t-1)
h(t)
h(t+1)
h(t+2)
B
x(t-2)
x(t-1)
x(t)
x(t+1)
x(t+2)

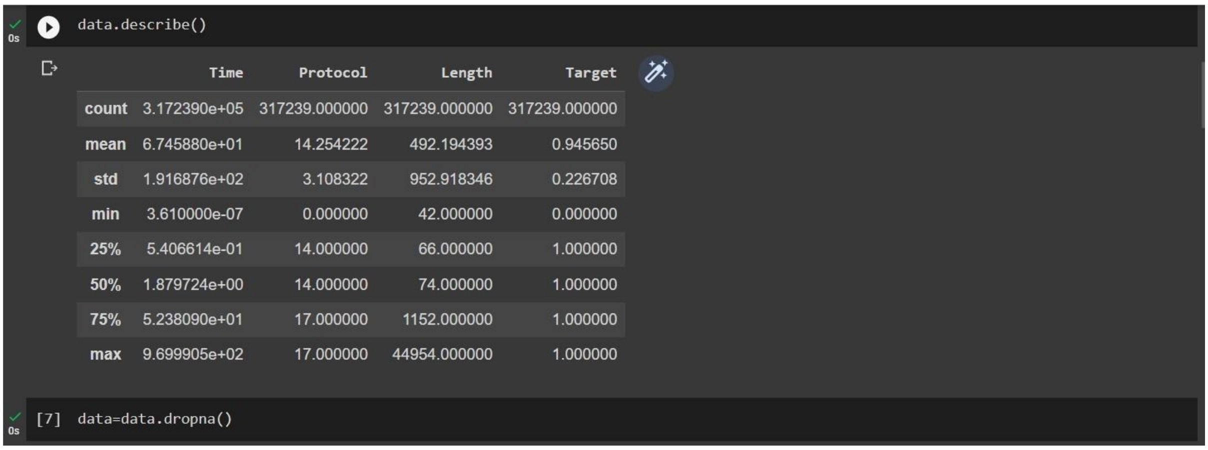
➢ Pre-processing



```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.impute import SimpleImputer
from sklearn.ensemble import RandomForestRegressor
from sklearn.preprocessing import StandardScaler
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Dropout
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
```

```
[4]  data = pd.read_csv('combined_data.csv')
     le = LabelEncoder()
     label=le.fit_transform(data['Protocol'])
     data['Protocol']=label
```

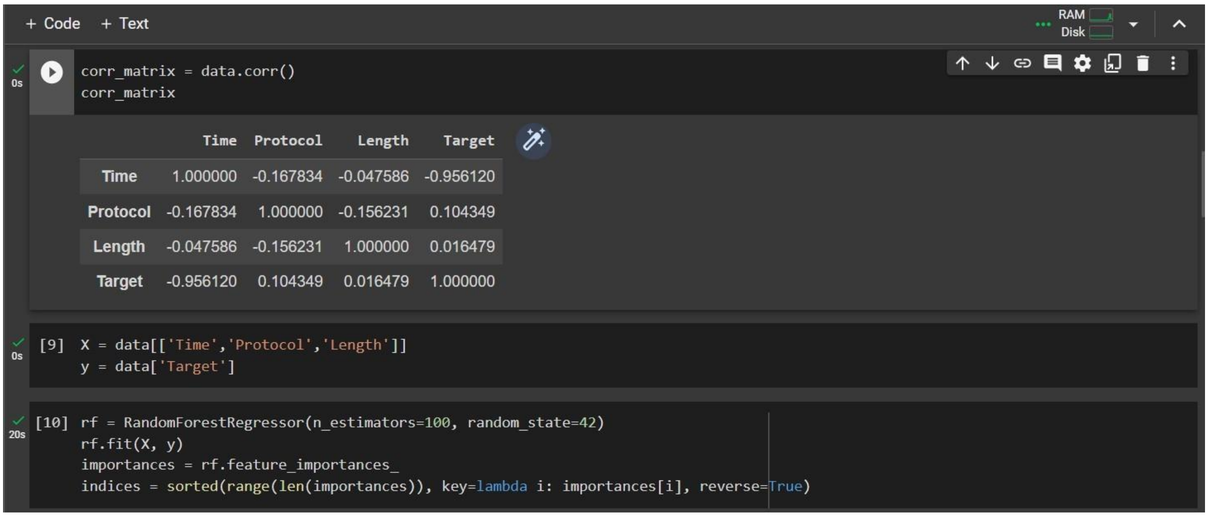
```
[5]  data = data.drop(data.columns[0], axis=1)
```



```
data.describe()
```

	Time	Protocol	Length	Target
count	3.172390e+05	317239.000000	317239.000000	317239.000000
mean	6.745880e+01	14.254222	492.194393	0.945650
std	1.916876e+02	3.108322	952.918346	0.226708
min	3.610000e-07	0.000000	42.000000	0.000000
25%	5.406614e-01	14.000000	66.000000	1.000000
50%	1.879724e+00	14.000000	74.000000	1.000000
75%	5.238090e+01	17.000000	1152.000000	1.000000
max	9.699905e+02	17.000000	44954.000000	1.000000

```
[7]  data=data.dropna()
```

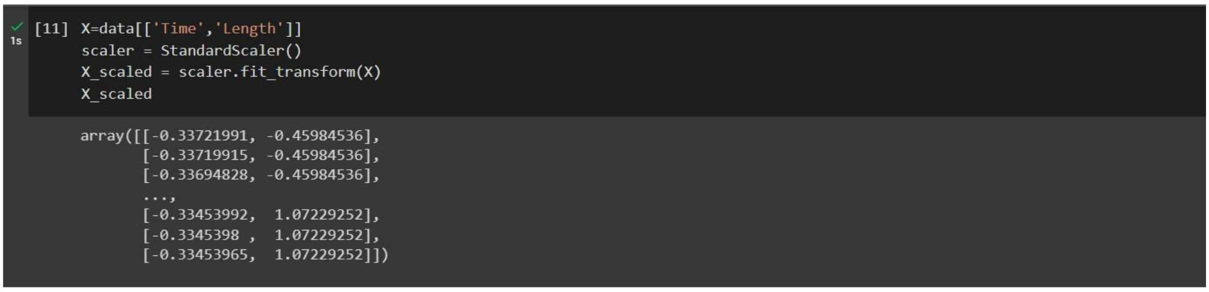


```
corr_matrix = data.corr()
corr_matrix
```

	Time	Protocol	Length	Target
Time	1.000000	-0.167834	-0.047586	-0.956120
Protocol	-0.167834	1.000000	-0.156231	0.104349
Length	-0.047586	-0.156231	1.000000	0.016479
Target	-0.956120	0.104349	0.016479	1.000000

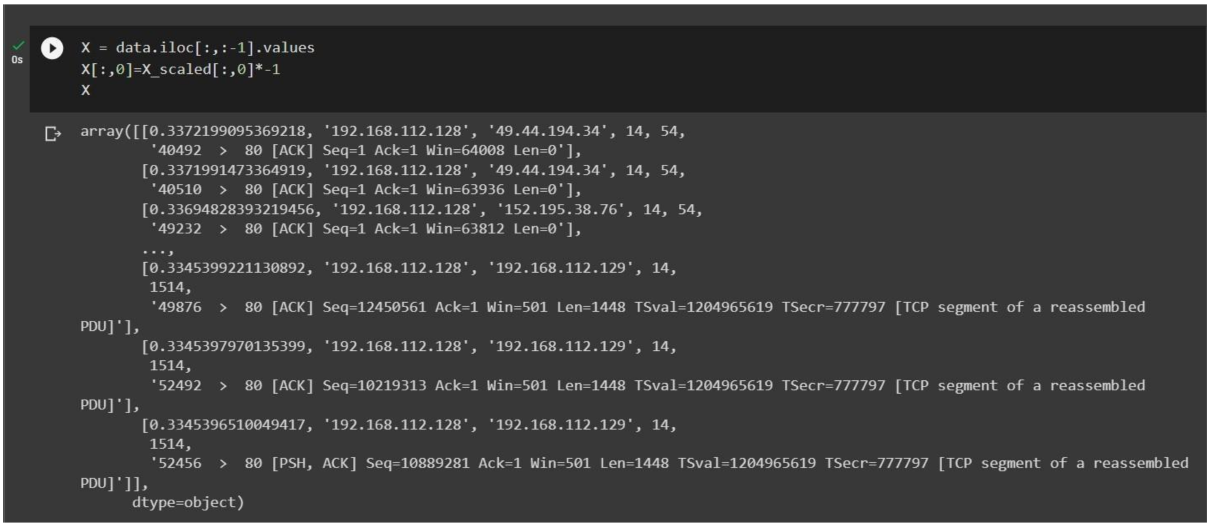
```
[9]  X = data[['Time','Protocol','Length']]
     y = data['Target']
```

```
[10] rf = RandomForestRegressor(n_estimators=100, random_state=42)
     rf.fit(X, y)
     importances = rf.feature_importances_
     indices = sorted(range(len(importances)), key=lambda i: importances[i], reverse=True)
```



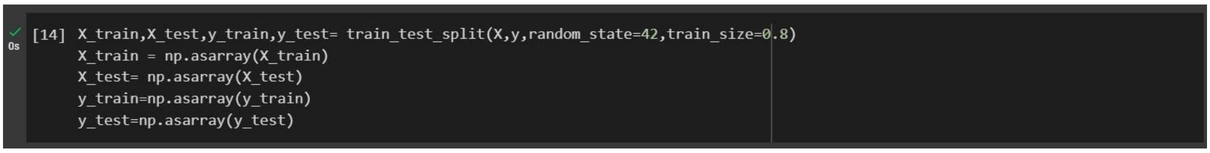
```
[11] X=data[['Time','Length']]
     scaler = StandardScaler()
     X_scaled = scaler.fit_transform(X)
     X_scaled

     array([[-0.33721991, -0.45984536],
            [-0.33719915, -0.45984536],
            [-0.33694828, -0.45984536],
            ...,
            [-0.33453992,  1.07229252],
            [-0.3345398 ,  1.07229252],
            [-0.33453965,  1.07229252]])
```

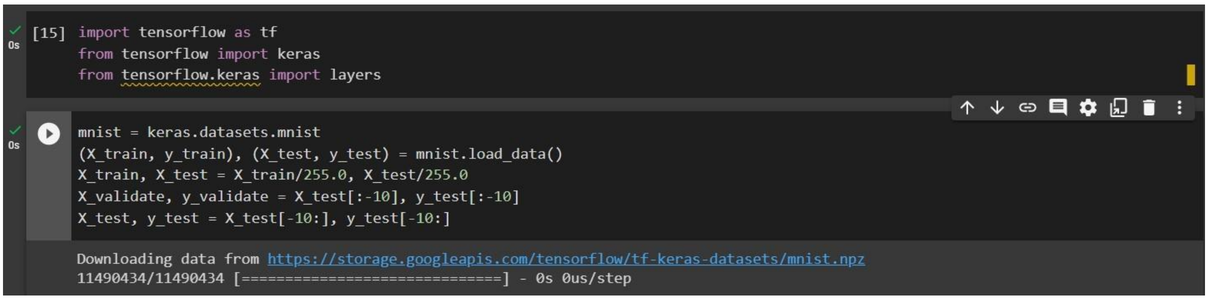


```
X = data.iloc[:,:-1].values
X[:,0]=X_scaled[:,0]*-1
X

array([[0.3372199095369218, '192.168.112.128', '49.44.194.34', 14, 54,
         '40492  >  80 [ACK] Seq=1 Ack=1 Win=64008 Len=0'],
        [0.3371991473364919, '192.168.112.128', '49.44.194.34', 14, 54,
         '40510  >  80 [ACK] Seq=1 Ack=1 Win=63936 Len=0'],
        [0.33694828393219456, '192.168.112.128', '152.195.38.76', 14, 54,
         '49232  >  80 [ACK] Seq=1 Ack=1 Win=63812 Len=0'],
        ...,
        [0.3345399221130892, '192.168.112.128', '192.168.112.129', 14,
         1514,
         '49876  >  80 [ACK] Seq=12450561 Ack=1 Win=501 Len=1448 TSval=1204965619 TSecr=777797 [TCP segment of a reassembled
PDU]'],
        [0.3345397970135399, '192.168.112.128', '192.168.112.129', 14,
         1514,
         '52492  >  80 [ACK] Seq=10219313 Ack=1 Win=501 Len=1448 TSval=1204965619 TSecr=777797 [TCP segment of a reassembled
PDU]'],
        [0.3345396510049417, '192.168.112.128', '192.168.112.129', 14,
         1514,
         '52456  >  80 [PSH, ACK] Seq=10889281 Ack=1 Win=501 Len=1448 TSval=1204965619 TSecr=777797 [TCP segment of a reassembled
PDU']],
       dtype=object)
```



```
[14] X_train,X_test,y_train,y_test= train_test_split(X,y,random_state=42,train_size=0.8)
     X_train = np.asarray(X_train)
     X_test= np.asarray(X_test)
     y_train=np.asarray(y_train)
     y_test=np.asarray(y_test)
```



```
[15] import tensorflow as tf
     from tensorflow import keras
     from tensorflow.keras import layers

     mnist = keras.datasets.mnist
     (X_train, y_train), (X_test, y_test) = mnist.load_data()
     X_train, X_test = X_train/255.0, X_test/255.0
     X_validate, y_validate = X_test[:-10], y_test[:-10]
     X_test, y_test = X_test[-10:], y_test[-10:]

     Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
     11490434/11490434 [==============================] - 0s 0us/step
```

METHODOLOGY

- **GRU Model**

The GRU (Gated Recurrent Unit) model is a type of recurrent neural network that is widely used in natural language processing and other sequence modelling tasks. The model consists of a set of recurrent units, each of which maintains a hidden state vector that represents its internal memory. At each time step, the model takes as input a vector of features (e.g., a word embedding) and updates the hidden state of each unit based on its previous hidden state and the input.

- **ANN Model**

An Artificial Neural Network (ANN) is a computational model inspired by the structure and function of the human brain. Here are the mathematical details of a simple feedforward ANN with a single hidden layer:

First, we define the weights and biases for the ANN. Next, we define the activations of the neurons in the hidden layer. We then apply an activation function f to the net input to obtain the output

Finally, we use an optimization algorithm such as gradient descent to minimize the loss function by adjusting the weights and biases of the ANN. This involves computing the gradients of the loss function with respect to the weights and biases, and then updating the weights and biases in the opposite direction of the gradients. The process is repeated until the loss function converges to a minimum.

- **RNN Model**

Recurrent Neural Networks (RNNs) are a type of neural network that can process sequential data, such as time series or text. They have a "memory" of previous inputs, which allows them to capture temporal dependencies in the data. Here are the mathematical details of a simple RNN with a single hidden layer:

Let's consider a sequence of input data with T time steps and n input features at each time step. First, we define the weights and biases for the RNN. Next, we define the activations of the neurons in the hidden layer. We then apply an activation function g to the output of the neurons in the output layer. Finally, we use an optimization algorithm such as gradient descent to minimize the loss function by adjusting the weights and biases of the RNN. This involves computing the gradients of the loss function with respect to the weights and biases, and then updating the weights and biases in the opposite direction of the gradients. The process is repeated until the loss function converges to a minimum.

- **MLP Model**

Multi-Layer Perceptron (MLP) is a feedforward neural network that can be used for both classification and regression tasks. It consists of an input layer, one or more hidden layers, and an output layer. Here are the mathematical details of an MLP with a single hidden layer:

First, we define the weights and biases for the MLP. Next, we define the activations of the neurons in the hidden layer. We then apply an activation function g to the output of the neurons in the output layer. We define a loss function that measures the difference between the predicted output and the actual output.

- **LSTM Model**

Long Short-Term Memory (LSTM) is a type of recurrent neural network that is designed to handle the vanishing gradient problem in traditional RNNs. It is particularly effective at capturing long-term dependencies in sequential data. Here are the mathematical details of an LSTM cell:

First, we define the inputs and outputs of the LSTM cell. Next, we define the gates of the LSTM cell. There are three gates: the forget gate, the input gate, and the output gate. The forget gate determines how much of the previous cell state should be retained, the input gate determines how much new information should be added to the cell state, and the output gate determines how much of the current cell state should be output as the hidden state. Where σ is the sigmoid function, Next, we define the candidate cell state. Finally, we use the output of the LSTM cell as the prediction. We define a loss function that measures the difference between the predicted output and the actual output.

Finally, we use an optimization algorithm such as gradient descent to minimize the loss function by adjusting the weights and biases of the LSTM cell. This involves computing the gradients of the loss function with respect to the weights and biases, and then updating the weights and biases in the opposite direction of the gradients. The process is repeated until the loss function converges to a minimum.

F. Experimentation & Analysis

1. Experimental setup

The experimental setup for detecting DDoS attacks using deep learning algorithms typically involves the following steps:

Data collection: Network traffic data is collected from the server(s) that are being monitored for DDoS attacks.

Data pre-processing: The collected data is pre-processed to remove any noise, irrelevant information, or duplicates. This step also involves feature selection and extraction, where important features are selected for training the deep learning models.

Data splitting: The pre-processed data is split into training, validation, and test sets. The training set is used to train the deep learning models, the validation set is used to tune hyperparameters and prevent overfitting, and the test set is used to evaluate the performance of the trained models.

Model training: Various deep learning models, such as ANN, RNN, LSTM, GRU, etc., are trained using the training data. The hyperparameters of the models are tuned using the validation set to optimize the performance of the models.

Model evaluation: The trained models are evaluated using the test set to measure their performance in detecting DDoS attacks. Various performance metrics, such as accuracy, precision, recall, F1 score, ROC-AUC, etc., are used to evaluate the models.

Comparison: The performance of different deep learning models is compared to select the best-performing model(s) for detecting DDoS attacks.

Deployment: The selected deep learning model(s) are deployed in a production environment to detect DDoS attacks in real-time. The model(s) may require periodic retraining to adapt to changing network traffic patterns and new attack vectors.

2. Analysis

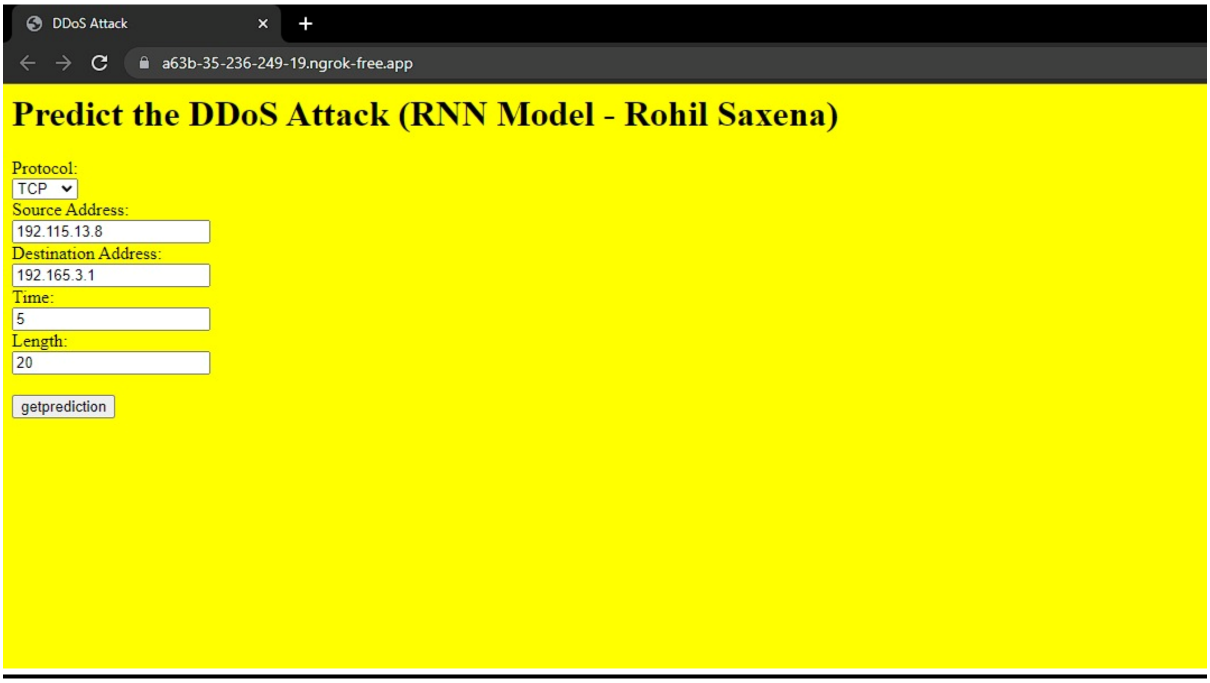
This analysis in based on the output of: https://drive.google.com/drive/folders/10RkkpANKjm90rXAz3UVHagr7z_V82Xup?usp=sharing

Model	Accuracy	f1_score	Recall	Precision
ANN	99..89	99.92	1.0	99.85
GRU	97.30	97.56	1.0	96.98
RNN	99.85	99.01	1.0	98
MLP	99.92	99.69	1.0	99.13
LSTM	22.38		1.0	20

Code For API along with Screen Shots (Review 3)

https://colab.research.google.com/drive/1XBowm8rCPUMix9leziQpQTPrDyALoUwL?usp=sharing

https://colab.research.google.com/drive/1rx8-t14vwVQ836XICzEuSqD3BkpkC1ML?usp=sharing



Predict the DDoS Attack (RNN Model - Rohil Saxena)

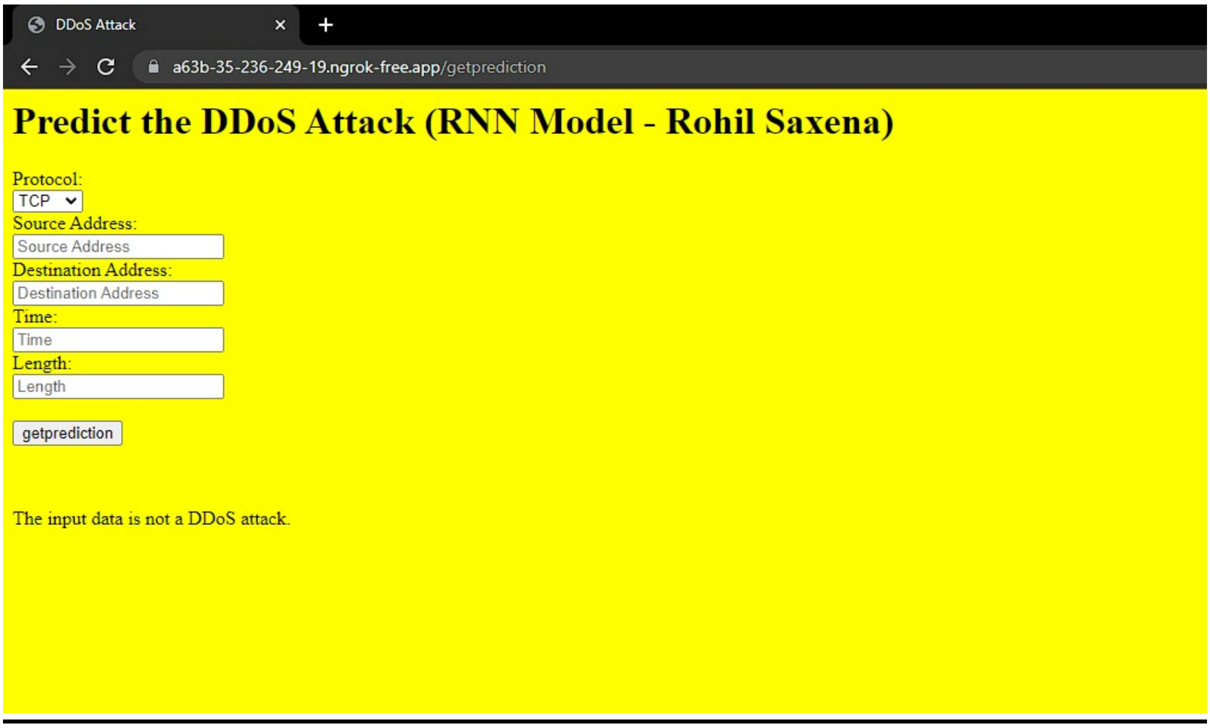
Protocol:
TCP

Source Address:
192.115.13.8

Destination Address:
192.165.3.1

Time:
5

Length:
20



Predict the DDoS Attack (RNN Model - Rohil Saxena)

Protocol:
TCP

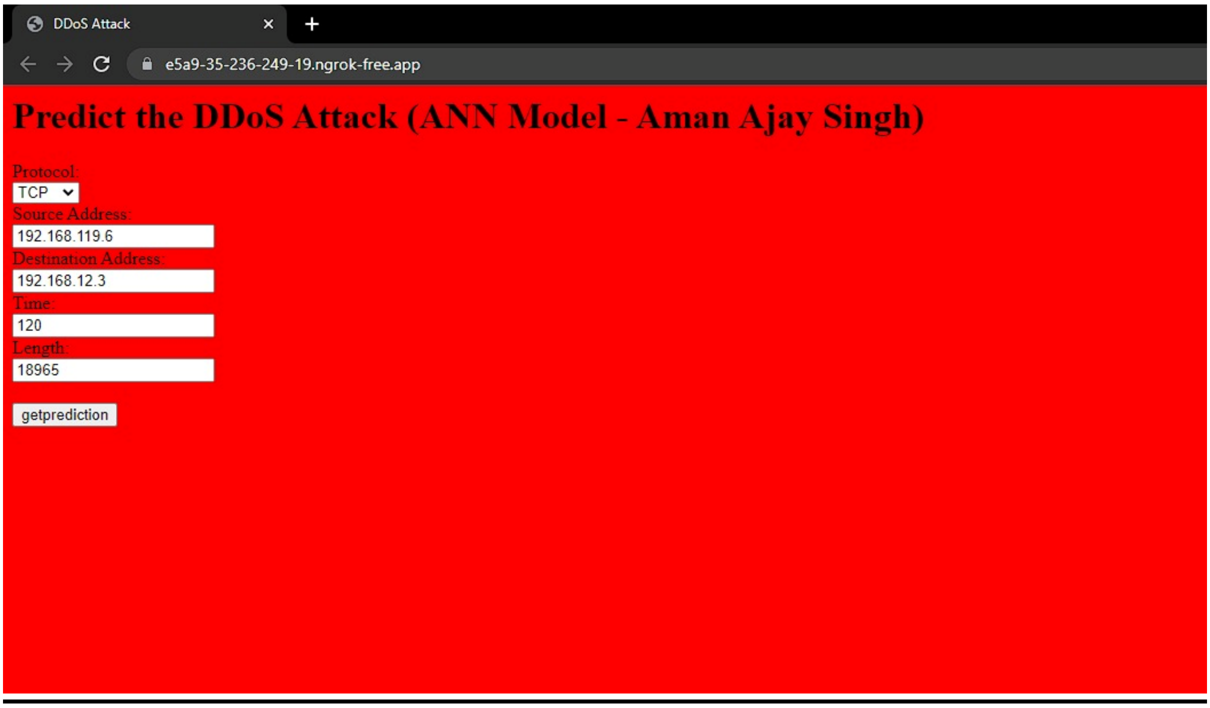
Source Address:
Source Address

Destination Address:
Destination Address

Time:
Time

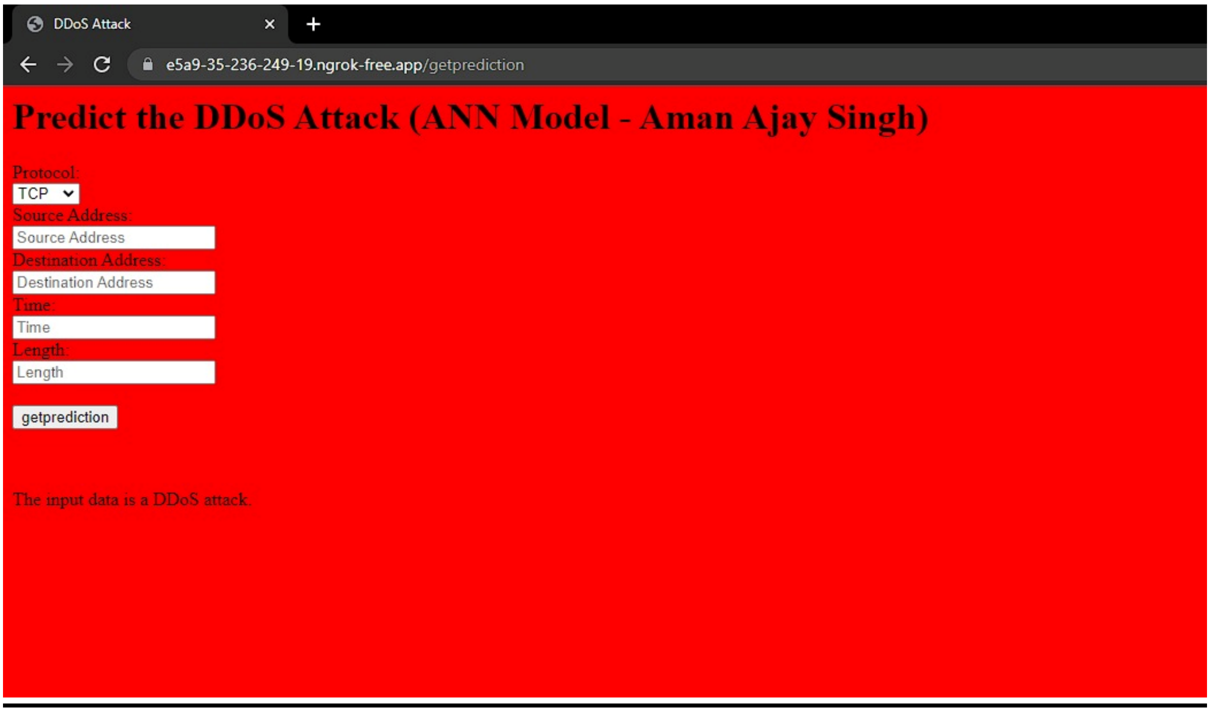
Length:
Length

The input data is not a DDoS attack.



Predict the DDoS Attack (ANN Model - Aman Ajay Singh)

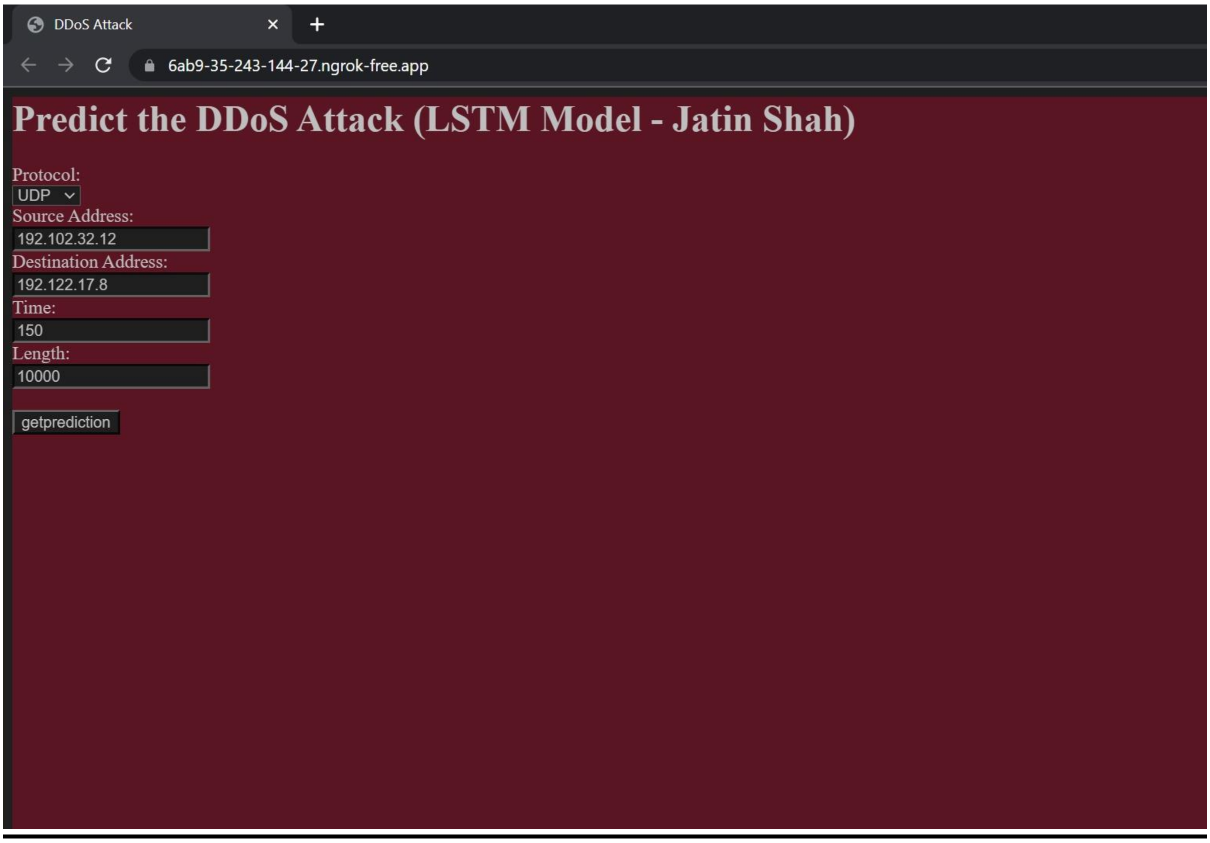
Protocol: TCP
Source Address: 192.168.119.6
Destination Address: 192.168.12.3
Time: 120
Length: 18965



Predict the DDoS Attack (ANN Model - Aman Ajay Singh)

Protocol: TCP
Source Address: Source Address
Destination Address: Destination Address
Time: Time
Length: Length

The input data is a DDoS attack.



Predict the DDoS Attack (LSTM Model - Jatin Shah)

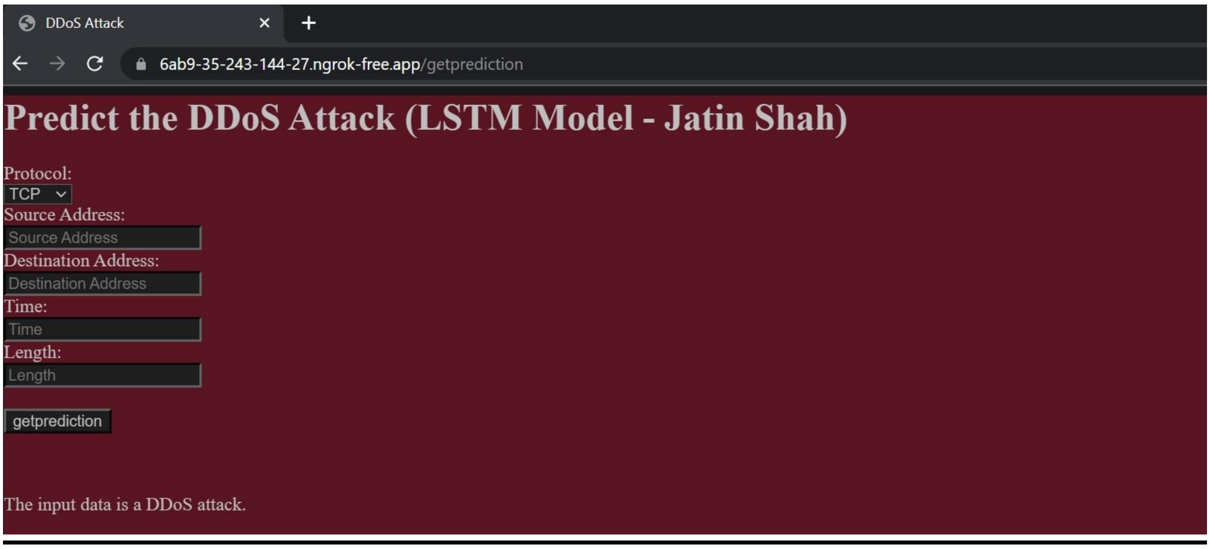
Protocol: UDP

Source Address: 192.102.32.12

Destination Address: 192.122.17.8

Time: 150

Length: 10000



Predict the DDoS Attack (LSTM Model - Jatin Shah)

Protocol: TCP

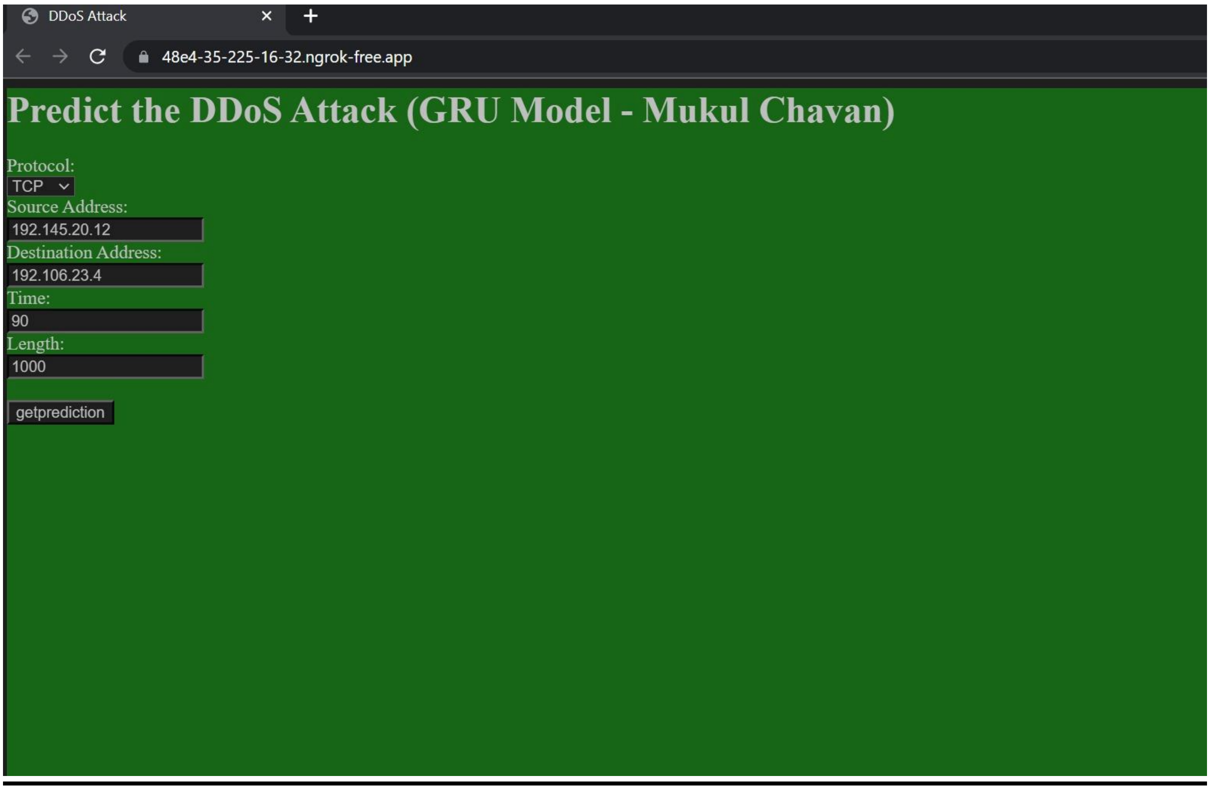
Source Address: Source Address

Destination Address: Destination Address

Time: Time

Length: Length

The input data is a DDoS attack.



Predict the DDoS Attack (GRU Model - Mukul Chavan)

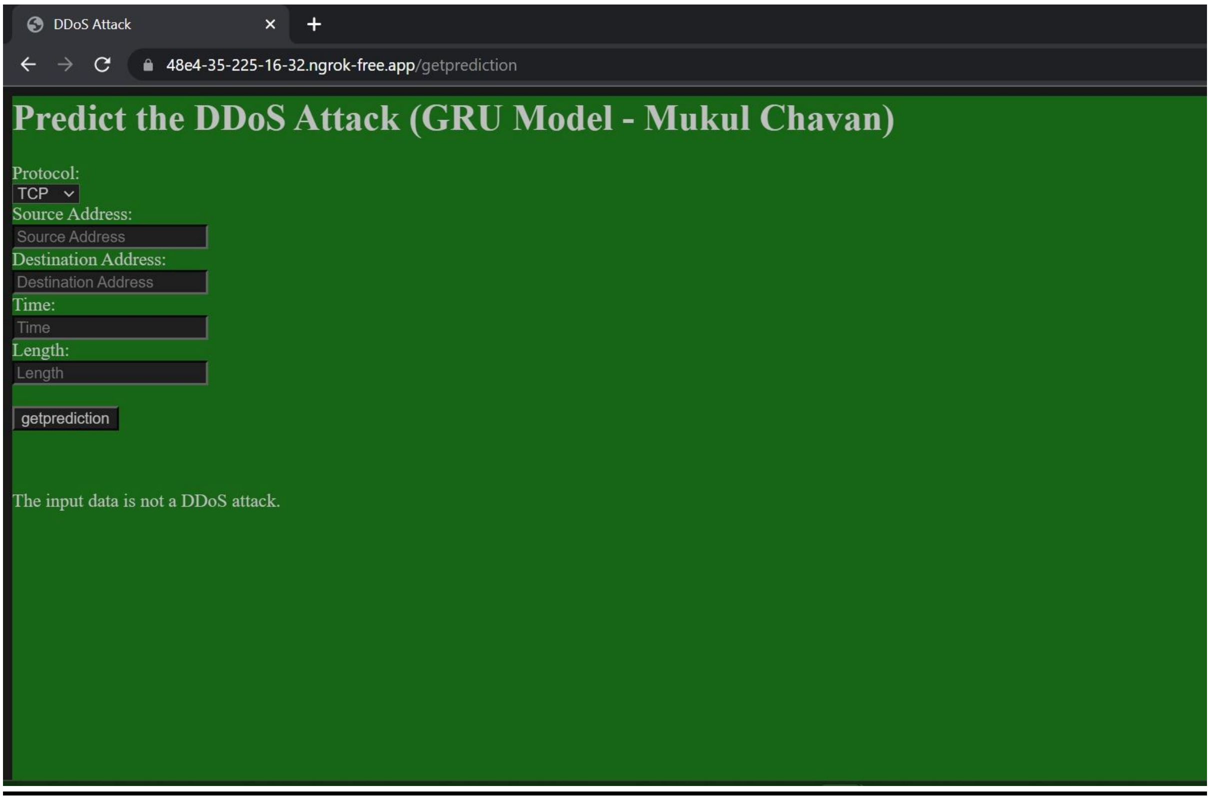
Protocol:
TCP

Source Address:
192.145.20.12

Destination Address:
192.106.23.4

Time:
90

Length:
1000



Predict the DDoS Attack (GRU Model - Mukul Chavan)

Protocol:
TCP

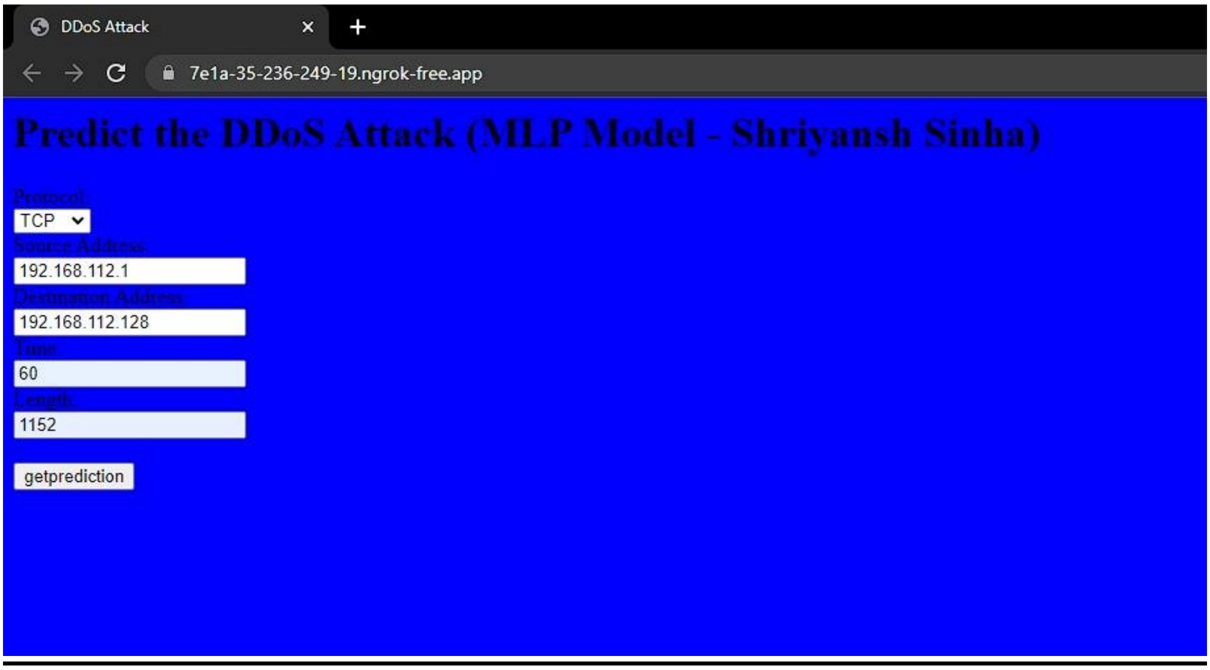
Source Address:
Source Address

Destination Address:
Destination Address

Time:
Time

Length:
Length

The input data is not a DDoS attack.



Predict the DDoS Attack (MLP Model - Shriyansh Sinha)

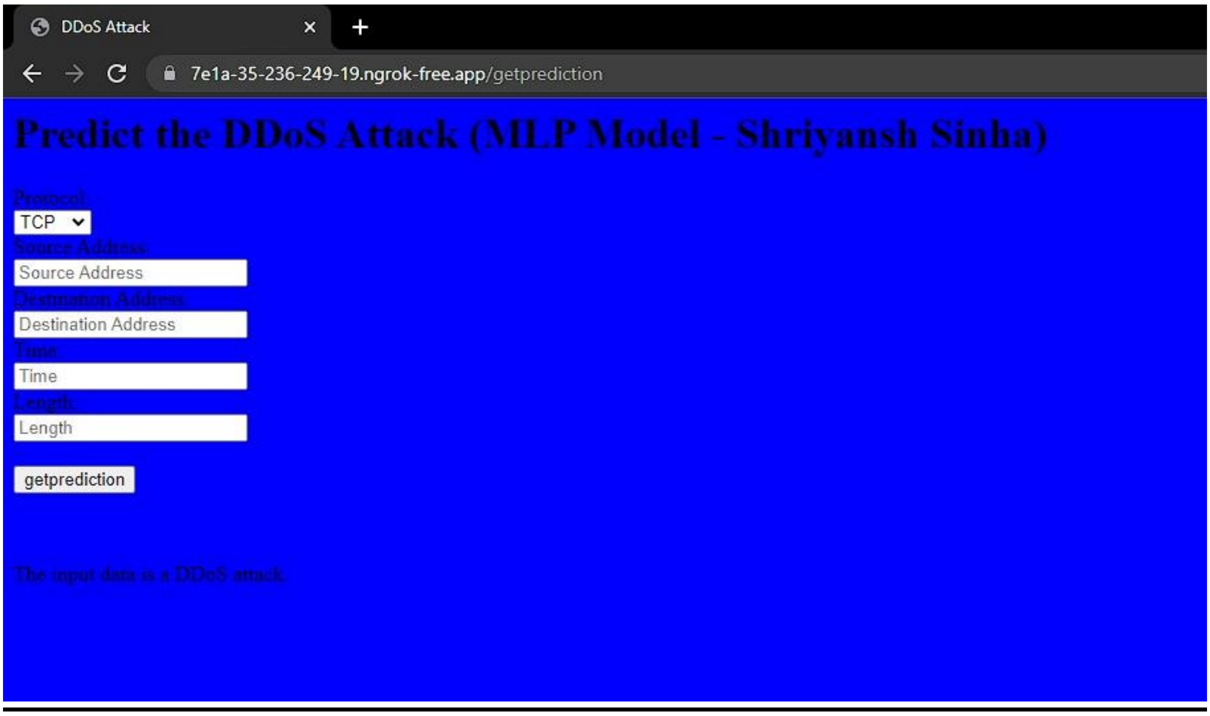
Protocol:
TCP

Source Address:
192.168.112.1

Destination Address:
192.168.112.128

Time:
60

Length:
1152



Predict the DDoS Attack (MLP Model - Shriyansh Sinha)

Protocol:
TCP

Source Address:
Source Address

Destination Address:
Destination Address

Time:
Time

Length:
Length

The input data is a DDoS attack

G. Conclusion & Future scope

In summing up, the findings of our research indicate that using deep learning algorithms is a method that has a lot of potential for identifying distributed denial of service assaults. These algorithms are able to identify both known and undiscovered forms of assaults with a high level of precision by capitalising on their capacity to understand complicated patterns in the data that represents network traffic. In addition, they have the capability of identifying assaults in real time, which enables a quick response and the prevention of further damage.

When we look to the future, we see that this field has a tremendous amount of promise for more research. One such area of concentration may be the creation of more advanced deep learning architectures that are especially optimised for DDoS detection. In addition, the application of transfer learning strategies, which involves adapting previously trained models to new datasets, may provide a method of developing DDoS detection models that is both more efficient and effective.

Integration of deep learning algorithms with other security measures, such as firewalls and intrusion detection systems, is another interesting and potentially fruitful area for future study. It is likely that by integrating these techniques, it will be feasible to develop a more complete strategy to cyber security that will be capable of identifying and mitigating a larger variety of threats.

In conclusion, the use of deep learning algorithms for the purpose of detecting distributed denial of service attacks is an interesting and fast developing area of research. These algorithms have the potential to greatly improve the security of internet infrastructure with continuous research and refining, and to limit the detrimental impacts of cyber assaults.

H. References

1. S. Al-Samarraie and S. A. Mahmood, "Deep Learning Framework for DDoS Attack Detection and Classification in Software Defined Networking," in IEEE Access, vol. 7, pp. 35239-35254, 2019.
2. M. A. Malik, A. Anpalagan and N. Javaid, "Anomaly Detection and Mitigation Using Deep Learning-Based Solutions: A Survey," in IEEE Access, vol. 7, pp. 107492-107512, 2019.
3. F. Al-Qahtani, M. A. Abdulsalam and S. S. Al-Ruwaili, "DDoS attack detection and mitigation using deep learning: A review," in Journal of Network and Computer Applications, vol. 171, pp. 102875, 2021.
4. S. S. Deepa and S. Lakshmi, "A Survey of Machine Learning Algorithms for DDoS Detection in Cloud Environment," in International Journal of Computer Science and Mobile Computing, vol. 7, no. 6, pp. 163-171, 2018.
5. K. Gautam and N. Jaiswal, "A survey on machine learning techniques for DDoS attack detection and prevention," in Journal of Network and Computer Applications, vol. 135, pp. 58-73, 2019.
6. X. Guo et al., "A DDoS attack detection approach based on deep learning and transfer learning," in PLOS ONE, vol. 15, no. 5, pp. e0232709, 2020.
7. B. Huang, X. Zhou and Y. Li, "A DDoS attack detection model based on convolutional neural network," in Journal of Intelligent & Fuzzy Systems, vol. 38, no. 5, pp. 5555-5564, 2020.
8. C. Liu et al., "A novel DDoS attack detection method based on convolutional neural networks," in Journal of Ambient Intelligence and Humanized Computing, vol. 11, no. 10, pp. 4251-4263, 2020.
9. S. Liu, X. Chen and J. Zhang, "Distributed denial of service attack detection using deep learning," in Journal of Intelligent & Fuzzy Systems, vol. 34, no. 6, pp. 3847-3855, 2018.
10. P. Mahesh and S. K. Vanga, "Detection of Distributed Denial of Service Attacks using Machine Learning Algorithms," in 2018 8th International Conference on Cloud Computing, Data Science & Engineering - Confluence, pp. 328-331, 2018.
11. G. B. Mondal and A. Pal, "Detection of DDoS Attack through Machine Learning Technique," in 2019 5th International Conference on Computing, Communication, Control and Automation (ICCUBEA), pp. 1-4, 2019.
12. S. S. Mukherjee, S. K. Saha and S. Ghosh, "A machine learning approach for detection of DDoS attacks in cloud computing," in Journal of Ambient Intelligence and Humanized Computing, vol. 11, no. 10, pp. 4229-4239
13. Singh, R. (2018). Deep learning based DDoS attack detection and mitigation. International Journal of Computer Applications, 179(45), 30-34.
14. Zhang, Y., Zheng, L., & Sun, Y. (2019). A deep learning approach to DDoS attack detection using convolutional neural networks. IEEE Access, 7, 17232-17241.
15. Lin, J., Yao, C., Chen, K., & Lin, J. (2018). A deep learning approach to DDoS attack detection in SDN-based networks. IEEE Transactions on Network and Service Management, 15(1), 176-189.

16. Gao, H., Xie, H., & Xie, G. (2019). A deep learning approach for DDoS attack detection using autoencoder-based unsupervised learning. Applied Sciences, 9(3), 510.
17. Ali, M., Iqbal, M. A., & Choo, K. K. R. (2019). DDoS attack detection using deep learning algorithms: a review. IEEE Access, 7, 50637-50652.
18. Bhuyan, M. H., Bhattacharyya, D. K., Kalita, J. K., & Sarmah, D. K. (2019). DDoS attack detection using deep learning algorithms. In Deep Learning and Parallel Computing Environment for Bioengineering Systems (pp. 145-161). Springer.
19. Khan, M. A., Rizwan, M., & Khan, M. A. (2019). DDoS attack detection and prevention using machine learning: A review. Journal of King Saud University-Computer and Information Sciences, 31(3), 347-362.
20. Qiu, Y., Li, S., Li, J., & Li, Q. (2019). DDoS attack detection using an improved deep learning model. Wireless Networks, 25(5), 2861-2872.
21. Singh, R., Kumar, N., & Chaudhary, A. (2019). A comparative analysis of machine learning based DDoS attack detection techniques. In Intelligent Computing Techniques for Smart Energy Systems (pp. 77-91). Springer.
22. Prakash, R., Gupta, B., & Sachdeva, N. (2018). Detection of DDoS attacks using deep learning algorithms. In 2018 IEEE 4th International Conference on Computational Intelligence and Networks (CINE) (pp. 67-71). IEEE.
23. Zou, Y., Cheng, Y., & Hu, X. (2020). An improved deep learning model for DDoS attack detection in cloud environment. Future Generation Computer Systems, 111, 39-49.
24. Sun, X., Huang, Y., & Cheng, X. (2020). An effective deep learning based DDoS attack detection system in a hybrid cloud. Applied Sciences, 10(11), 3769.
25. Jiang, J., Gao, Y., & Xu, L. (2020). DDoS attack detection based on convolutional neural network and ensemble learning. Cluster Computing, 23(3), 2063-2072.
26. Tsai, C. W., Kuo, Y. H., & Chen, Y. C. (2020). Deep learning-based DDoS detection in IoT using stacked autoencoder and LSTM. IEEE Internet of Things Journal, 8(14), 111
27. Arora, M. Kumar, and V. K. Sharma, "A Survey on Machine Learning Techniques for DDoS Attack Detection," in International Journal of Advanced Research in Computer Science, vol. 9, no. 2, pp. 12-18, 2018.
28. S. M. Kamruzzaman and A. Hussain, "A deep learning approach for DDoS attack detection and classification," in IEEE Access, vol. 6, pp. 61572-61584, 2018.
29. T. H. N. Le, T. N. Nguyen, T. N. Do, T. D. Nguyen and D. H. Tran, "A deep learning-based approach for DDoS attack detection in software-defined networks," in Computer Networks, vol. 129, pp. 61-73, 2017.
30. N. Gupta and N. Chilamkurti, "Deep learning approach for detecting DDoS attacks in cloud computing environments," in Future Generation Computer Systems, vol. 82, pp. 180-187, 2