

A critical analysis of machine learning algorithms for detecting Distributed Denial of Service attacks

Academic Internship
MSc in Cyber Security

Catherine Stack
Student ID: x20178573

School of Computing
National College of Ireland

Supervisor: Vikas Sahni

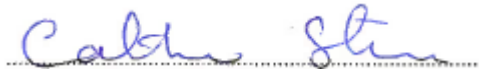
National College of Ireland
MSc Project Submission Sheet
School of Computing

Student Name: Catherine Stack
Student ID: X20178573
Programme: MSc in Cyber Security **Year:** 2023
Module: Academic Internship
Supervisor: Vikas Sahni
Submission Due Date: 25 April 2023
Project Title: A critical analysis of Machine Learning Algorithms for detecting Distributed Denial of Service attacks
Word Count: 7176 **Page Count** 22

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:



Date: April 24th 2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

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

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

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

A critical analysis of Machine Learning Algorithms for detecting Distributed Denial of Service of Service attacks

Catherine Stack
X20178573

Abstract

The rapid increase in remote working and cloud migration has led to increased cyber-attacks along with the enhanced opportunity for cyber threats. This has propelled cybersecurity front and centre. In the latter part of 2020, ransomware groups were incorporating Distributed Denial of Service (DDoS) attacks into their ransom attacks in a kind of twin extortion so that leaves the victim under constant DDoS attack until the ransom is paid. This is why organisations must use more intelligent defensive mechanisms using Machine Learning algorithms in cybersecurity protection. The motivation for this research paper was to critically analyse machine learning algorithms used in the detection of DDoS attacks. The algorithms Random Forest, K-Nearest Neighbour, Naive Bayes, Decision Tree and Logistic Regression were analysed for precision, accuracy, recall, f1 rating along with training time of each classification model.

The results show that out of the five ML algorithms assessed, Random Forest, and K-Nearest Neighbour satisfied the problem statement goal of predicting 95% or greater accuracy. The Random Forest classifier performed the best overall with a 99% accuracy followed by K-NN with 96% accuracy. Logistic Regression performing least favourably with a 50% accuracy with Naïve Bayes and Decision Tree having an 85% and 92% respective accuracy percentage rates. Training time on the other hand had the Random Forest classification model perform poor with a time of 422ms recorded which was many times slower than that of all of the other classification models with KNN perform the quickest with 15.6ms and the remaining having a time of 31.2ms each.

Keywords: Machine Learning algorithms (ML), Distributed Denial of Service (DDoS), Flooding attack, DDoS defense mechanisms, Botnets.

1 Introduction

Why businesses must upgrade defence strategies so rapidly is because it is more difficult than ever to prevent and defend against cyber-attacks which are evolving at an alarming rate. Over the recent past, awareness has grown on how important cybersecurity is for businesses. Transition to online working activity, along with migration to the cloud, compelled businesses to up their defenses quickly.

Businesses were moving in that direction anyway, but covid abruptly forced them to operate online, leaving little choice for businesses to either embrace the challenge or give up and go out of business. Significant reports of new cyber-attacks, data breaches or zero-day attack come to light almost on a daily basis. Cybercrime Magazine¹ reported report daily cyber attacks. A sample of these attacks since April 19th 2023 alone include; Air traffic control agency (Eurocontrol) in Europe was actively being targeted by pro-Russian threat actors; the American Bar Association had 1.5 million account details stolen; a New England healthcare provider, Point32Health and the state school district in Tucson Arizona were the victims of ransomware attacks and the list goes on and on. In 2022. Cybersecurity software firm Imperva increased the frequency of their reporting of new threat observations and information from annually to quarterly due to the rise in the threat landscape geopolitical events around the world. They reported a four-fold increase in cyber-attacks aimed at Russia and Ukraine locations and observed the highest number of attacks in March 2022.² The DDoS mitigation company Cloudflare [4] reported that they had prevented a DDoS attack to one of their financial services customers in July 2021, the size of which was nearly 3 times the size of any previous attack that they had come across. Thousands of bots, spread over 125 countries were used and made more than three million attack requests in just a few second time.^{3 4} Akamai reported there had been more DDoS attacks in the year 2020 than any other year previous with more companies requiring assistance due to constant attacks.⁵ The style of DDoS attacks were nowhere similar to the type of attacks from a decade ago, due to the evolving nature of the threat landscape in that time period. In addition, the most prevalent method of attack were UDP floods, SYN floods followed by packet fragmentation attacks due to their simplicity and success rate.

1.1 Background and Motivation

DDoS attacks are normally launched in two phases. The first phase involves the infection of insecure devices with some form of malware with the intention of controlling them. A threat actor infects an insecure device by installing malware that allows the malicious user to commandeer the device when required. This device is known as a bot and a network of infected devices is called a botnet⁶. Hijacked devices can be many in number and are all under the control of a malicious actor. An

¹ Cybercrime Magazine article "Who's Hacked? Latest Data Breaches And Cyberattacks" <https://cybersecurityventures.com/intrusion-daily-cyber-threat-alert/>.

² "DDoS Threat Landscape Report Q1 2022" <https://www.imperva.com/resources/resource-library/reports/ddos-threat-landscape-report/>

³ Cloudflare article "What is DDoS attack?" <https://www.cloudflare.com/learning/ddos/what-is-a-ddos-attack/>

⁴ "Record-Setting DDoS Attack Hits Financial Service Firm" Prajeet Nair 21/08/2021 <https://www.govinfosecurity.com/record-setting-ddos-attack-hits-financial-service-firm-a-17345>

⁵ "Cyberterrorists Target Record Number of Victims with DDoS Attacks in Q2" Craig Sparling September 07, 2022 <https://www.akamai.com/blog/security/cyberterrorists-target-record-number-of-victims>, "Relentless evolution of DDoS Attacks" <https://www.akamai.com/blog/security/relentless-evolution-of-ddos-attacks>

⁶ Botnet definition What is a Botnet? - Palo Alto Networks

infected bot can scan for other insecure devices to infect them in the same way. The infected device can be computers or IoT ⁷device and is usually unaware of their infection. The second phase involves the attacking party commanding every device on the botnet to simultaneously carry out a coordinated attack on a target machine or website. The botnet controller dictates commands to the bots to flood a target with TCP and/or UDP packets and legitimate users are prevented from accessing the server. Because the victim machine is flooded with bogus requests, it becomes overwhelmed rendering the network or website unusable and simply crash. As the source of the network traffic is distributed, DDoS attacks are very difficult to identify, because it is almost impossible to differentiate legitimate traffic from malicious traffic. Several services can be affected by a DDoS attack including access to network devices, websites, email, or online accounts.

Peer-to-peer networks are different to client server network. The nodes are peers in P2P networks and any node can communicate with any other node, usually for file sharing and usually span over large geographical areas. The peers are used by attackers to launch DDoS attacks on targets which makes them very difficult to deal with an attack because there is no head peer.

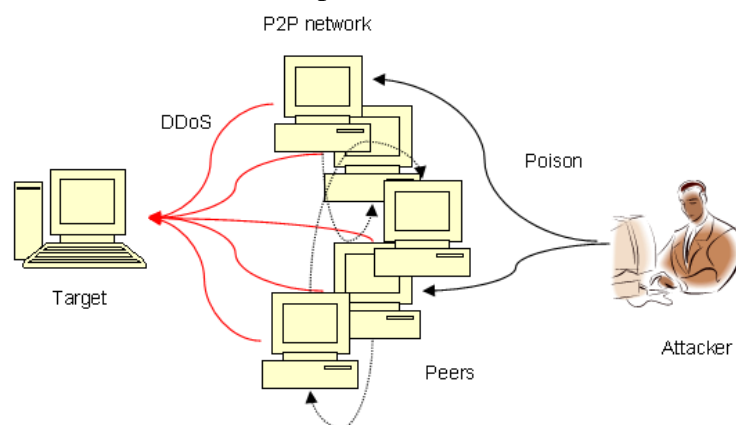


Figure 1 Overview of PDDoS attack over a P2P network [20]

Different types of DDoS attack exist. The most common are Application layer attacks, Network layer attacks and Protocol layer attacks.

- Application attacks involve sending request, such as a HTTP request, to a server to generate web pages which can require a lot of resources to fulfil the request. The objective is to overwhelm the server resources resulting in denial of service to other traffic.
- In Volumetric or Network layer attacks, the attacker will monopolise the entire network resources with the use of flooding User Datagram Protocol (UDP) requests. This is normally a two-pronged attack as reflection is used to masquerade the source IP address and amplification is used to send the request to UDP services to prompt large responses to the spoofed (victim)

⁷ Internet of Things (IoT) is a term used to describe a network of objects that have the technologies to connect to other objects either via the internet or different network connections such as Wi-Fi, Bluetooth and so on.

IP address. This amplifies the replies, overwhelming the victim and causing it to slow down or crash altogether.

- Protocol attacks operate by overburdening a server or network equipment such as firewalls and load balancers by consuming resources. SYN flood or half-open attack utilises a known vulnerability in the TCP connection sequence which is a three-way handshake sequence. Three-way handshake is where the client sends a SYN request/message, receives a SYN-ACK response from the server and the client sends an ACK back to complete the transaction. The malicious attacker sends a (flood of) SYN request to the victim but does not complete the handshake. The attacker continues doing this until all open ports are occupied and this leaves the victim machine busy and unusable. A smurf attack is where the attacker will broadcast Internet Control Message Protocol (ICMP) packets, but it spoofs the sender's IP address as the target machine's resulting in the target being overwhelmed with responses from the ICMP broadcast. The target becomes inoperable leading to a total denial of service.

Machine learning algorithms are used in Intrusion Detection Systems (IDS) to detect anomalies in network traffic that are indicative of a DDoS attack. Analysis of network traffic surmise if features are detected that would indicate a DDoS attack. Use of algorithms are commonplace in present day IDS, however there can be a lot of overhead with this operation. Extensive processing power and time can be necessary in the detection process itself. The related work section outlines papers that analysed ML algorithms in IDS environments and reported their findings on detection rate, precision, overhead cost and scalability. This paper was a critical analysis of ML algorithms in the area of detecting DDoS traffic and measured accuracy, precision, recall, f1 score and model training time.

1.2 Research Question

With a given dataset, which classification model is more accurate when data is processed and transformed to create a classification model so that the model will predict which observations in the dataset are likely to be DDoS attack data within a 95% accuracy rate.

The research made use of the dataset[1] which was used in other studies [9] [11] [5]. It was an amalgamation of three datasets that has all relevant information required to identify DDoS activity. Since this was a very large dataset, the most relevant features were selected to help faster training and detection. A split of 80%/20% was used for splitting the dataset for training and testing respectively. Precision, recall and f1 rating will be used to evaluate each algorithm along with training time for each.

2 Related Work

DDoS attacks are a potent yet simple method for a malicious actor to use to target a victim. A detailed literature review was carried out to research the classification and multiple approaches to DDoS attack detection.

2.1 DDoS Threats

Much has been written about the use of machine learning algorithms in the detection of DDoS threats. The attacker's ability to compromise a network or a combination of devices can make the attack difficult to deal with as described in the paper on DDoS defence systems.[18] Zargar et al. examined existing types of DDoS attacks and defence mechanisms. The motivations behind DDoS attacks are outlined, with financial gain being the top motivating factor, aimed primarily at corporations. Ideology and Cyberwarfare also contribute to attackers' motivation to carry out strikes aimed at political targets or nation states for such things as human rights violations or censorship restrictions. DDoS attacks are combined in other forms or attack also as was the case with the ransomware attack on the Health Service Executive (HSE) in Ireland fell victim of in May 2021 [6]. There are any number of reasons motivating threat actors thus the better course of action is to detect and prevent them. Jaideep etl al. [20] examined the problem of DDoS threats in the distributed P2P network environment. In the file sharing nature of the P2P setting, threat actors utilize them as a vehicle to launch DDoS attacks from the widespread nature of the P2P nodes. The research concentrated on a detection algorithm, P2P DDoS Detection (PDD) for detection and defence in a distributed environment. A cluster approach was proposed using a mechanism of time-to-live (TTL) value, distance of source from victim along with an agent, to manage this information about each node. The research supported the theory that the cluster-based approach outperformed traditional methods in overhead and downloading speed per node in detection and prevention of the DDoS attacks. The use of P2P network security within the realm of Bitcoin was analysed in [21] where Tapsel et al. tested bitcoin messages interactions (protocol exchanges) susceptible to spoofing and DDoS attack. The potential for a TCP sequence number being compromised leading to the connection handshake being used in a possible DDoS attack the victim. A proposed prevention method of a nonce (number used once only) with the VERACK (version acknowledge) message preventing the attacker from spoofing the connection handshake. Emphasising that the nonce generation must be suitably complex and randomly generated to prevent it from being guessed, they propose these methods to improve on the security of the bitcoin P2P network protocol.

In the client/server environment, Suresh et al. [16] discuss the insufficiencies in methods of detection and filtering and outlines the difficulty of the detection systems can be limited because of the changing nature of DDoS attack threats. Detection methods such as traffic filtering cause obstruction and displays the inadequacy of signature-based DDoS detection due to the changing nature of attack signatures. The

research concentrated on selecting features of datasets (CAIDA Dataset⁸) with DDoS traffic parameters and evaluate the performance of ML algorithms in detection using F-measure and receiver operating characteristic (ROC)⁹ to measure accuracy levels. The machine learning algorithms evaluated were Naïve Bayes, KNN, K-means, fuzzy c-means, SYM and C4.5. Among the findings reported was the Fuzzy c-means had a 98.7% accuracy with .15 seconds being the fastest of the group. Fuzzy c-means also performed best in f-measure with a measure of .987 followed by Naïve Bayes with .982 f-measure.

The problem of deciphering legitimate from malicious traffic is the subject of Kotey et al. [17] They discuss how even an attacker with low skill levels have caused much disruption because of the freely available tools that can be garnered online. Attackers have become much more sophisticated now with the use of botnets to facilitate an attack. The research examined the results four tested DDoS defence mechanisms, categorised as: detection only, attack traceback only, mitigation only mechanisms and finally detection plus mitigation defences. The research amalgamates and examined the findings of each on scalability, accuracy, good packet loss, precision and finally computing overhead. Findings reported no solution performed well with scalability with all non-performant in any large-scale attack. Detection rates and benign packet loss were found to be performant, however, apart from some very limited circumstances. Overhead costs were significant with both traceback and detection but reported that RADAR, SD-Anti-DDoS, and CMIYC had lowest overhead costs to the network. The researchers did acknowledge that much of the papers lacked complete results of their research and as such could not be included in the findings.

2.2 Machine learning algorithms effectiveness

The authors in [11] made a proposal to detect DDoS attack data with the application of morphological fractal dimension (MFD) with an online algorithm based on a sliding window. They tested publicly available dataset CICIDS2017¹⁰. Morphological, relating to form of things, is commonly applied for imaging or geometric shapes but is used in this study in the contest of intrusion detection systems for the detection of the DDoS attack data. The researchers also used a sliding window approach to dynamically improve on detection accuracy which to the researchers' knowledge had not been proposed in an intrusion detection system previously. The research resulted in a 99.30% detection accuracy after fine tuning of hyper-parameters for optimal performance for the application of MFD. Alduailij et al. approached DDoS attack detection in cloud computing with the aim of minimising the misclassification rate in the detection results.[5] On the CICDDoS2017 and CICDDoS2019¹¹ dataset, the most relevant features were extracted using mutual

⁸ The Cooperative Association for Internet Data Analysis (2001)https://catalog.caida.org/dataset/telescope_codered_worm

⁹ F-measure a measure to test accuracy using statistical analysis of binary classification. (ROC) curve a graph to plot two parameters, true positive and false positive

¹⁰ Intrusion Detection Evaluation Dataset (CIC-IDS2017) <https://www.unb.ca/cic/datasets/ids-2017.html>

¹¹ DDoS Evaluation Dataset (CIC-DDoS2019) <https://www.unb.ca/cic/datasets/ddos-2019.html>

information (MI) and random forest feature importance (RFFI). Logistic Regression, K Nearest Neighbour, Gradient Boosting, Random Forest and Weighted Voting Ensemble Classifier (WVEC) were applied for testing for recall, accuracy, precision and F score. Their tests were carried out on three sets of the data and the results on each set found that Random Forest had a 99% prediction accuracy compared to the other methods. However, on the smaller feature dataset of 16, WVEC method performed the best reporting that they proved that feature selection was especially important if accuracy is the goal as the research.

In [14] Saranya et al. analysed how machine learning performed in the context of IDS operating in different environments. The research examined the application of ML algorithms in IDS, operating in the areas of Internet of Things (IoT), Smart City (smart water distribution works), Big Data environment, Fog Computing¹² and Mobile computing. The evaluation metrics used were accuracy, precision, recall and f-score. RF yielded the best on all environments in accuracy with a 99.65% result with Linear Discriminant Analysis (LDA) and Classification and Regression Tree (CART) just behind it with a 98% result. Support Vector Machine had the highest false positive rate and lowest true positive rate. They analysed the KDD99¹³ Cup dataset testing LDA, CART and RF classifiers. The results from this analysis matched the published literature they had analysed showing that the operating environment is critical to the application of machine learning classifiers for intrusion detection systems.

Hoon et al. concentrated a study the DDoS forensics and for the aim of recommending the best learning model for this purpose.[13] The research took two perspectives in big data forensics: locating a small piece of key information in a large dataset and finding out undiscovered facts in big data. This research concentrated on the comparison of supervised ML with unsupervised ML algorithms for DDoS forensics and employing big data for this purpose. NSL-KDD¹⁴ dataset was analysed for this study and was pre-labelled ready for classification. This study tested for accuracy, precision and recall on supervised and unsupervised algorithms but also compared the data mining tools used. The selected tools were Waikato Environment for Knowledge Analysis (WEKA) and H2O¹⁵. WEKA has visualisation ability and has its own GUI as well as many learning algorithms compared to H2O as its optimised learning algorithm number is limited however H2O's performance analysis of the classification models is superior. The testing was carried out on the same machine with the same parameters. Results found supervised learning algorithms performed better than the unsupervised. Gradient Boosting Machine had the highest performance with 90%, 99.7% and 100% for accuracy, precision and recall

¹² Fog Computing a decentralised computing infrastructure to bring data closer to where it is managed using the cloud. It is closer to end user

¹³ KDD Cup 1999: Computer network intrusion detection. Database contains a standard set of data & includes a wide variety of intrusions simulated in a military network environment. <https://www.kdd.org/kdd-cup/view/kdd-cup-1999/Data>

¹⁴ oN-Line System – Knowledge Discovery & Data mining (NSL-KDD) most common data set is the NSL-KDD, and is the benchmark for modern-day internet traffic

¹⁵ WEKA is a set of data mining tools that run on Java. H2O.ai is an open-source, distributed in-memory machine learning platform with linear scalability

respectively. The other supervised classifiers were deep Learning, Distributed Random Forest (DRF) and Naïve Bayes. Unsupervised classifiers chosen were Canopy, Farthest First, Filtered Cluster and Make Density Based Cluster (MDBC) of which MDBC and Farthest first outperformed the others in that class.

This research paper [9] (Prasad, Babu & Amarnath 2019) aims to remove the need for human intervention in anomaly-based intrusion security technologies to detect DDoS attacks and thereby making the process automatic. The problem of signature-based threats was again the motivation for this research whereby very small tweaks to an underlying attack threat can change a signature and can appear non-malicious and go undetected in a system. Amarnath et al. point out anomaly-based systems falling short also and used Stochastic Gradient Boosting (SGB) ensemble learning classification, to detect DDoS attack data. SGB resulted in perfect accuracy rate of 100% or zero false positives with this model by applying precise tuning of hyperparameters in their machine learning model. They compare their results with other ML algorithms to gauge the effectiveness of their research. This is no mean feat to achieve as many of the we would expect that there will always be a percentage of false positive/false negative in results with very few or no classifier being correct all the time. They compared SGB with K-NN, Naïve Bayes, Decision tree and Random Forest and measured recall, F1-score, precision, and accuracy. SGB performed with an accuracy of 100% over the other classifications while Naïve Bayes had the lowest accurate rate of 91.8% but had the fastest execution time of 20 seconds. Pei et al. [22] researched the inaccuracy detection levels owing to the various size and range of DDoS attacks. They outline the use of 3-5 DDoS attack characteristics for identifying many to one attack indicators: source IP addresses, traffic flow density and destination port information. However, they point out most DDoS detection use only a portion of these indicators leading to poor detection results. The researchers here used an attack tool (TFN2K - Tribe Flood Network 2000¹⁶) to obtain DDoS attack traffic data and using the Bootstrapping method, they extract sets of samples. The results found RF performed better than SVM in the monitoring of TCP, UDP and ICMP flood traffic. In false positive rates RF averaged 15% compared to 50% for SVM and in the detection rate, RF averaged 98.5% compared to 95.7% for SVM.

Li et al. [15] focus on detecting DDoS attacks with the use of deep belief network for feature extraction and long short-term memory (LSTM) machine learning algorithm. They proposed using deep belief ¹⁷nets for feature extraction as deep belief learns one layer at a time and, is made up of layers of latent variables ("hidden units"). Extraction of the IP packet features using deep belief network was the first step of the technique. Next, they launched LSTM network traffic prediction model resulting in detection of DDoS attack traffic based on the model. Comprised of multiple layers of latent variables, it accurately detected the trend of normal network traffic and picked out the anomalous traffic. In practice recurrent neural networks

¹⁶ TFN2K works in a client/server environment where the client issues commands simultaneously to a set of TFN2K servers and those agents then conduct the DDoS attacks against a target. <https://www.igi-global.com/dictionary/tribe-flood-network-2000-tfn2k/36475>

¹⁷ Deep belief networks; Scholarpedia: http://www.scholarpedia.org/article/Deep_belief_networks

are subject to gradient loss or vanishing gradient problem. The paper recommended the use of LSTM to overcome this problem so that recalling long term behaviour becomes the default pattern behaviour rather than processing input sequences of random time series.

The literature above covers various models and approaches and are tested on various forms of datasets to improve the detection rates of DDoS attacks. Some, use different feature selection methods while others propose concentration on the classification models for a more effective approach. All underscore the importance of the feature selection phase of the process though on their selected datasets, along with the speed and resources or, speed of identifying the DDoS threats.

Summary of Literature Review

| Researcher | Description |
|--|--|
| Zargar, Joshi & Tipper [18] | ML algorithm assessment using big data analytics for DDoS forensics |
| Jaideep & Battula [20] | Proposed P2P DDoS Detection algorithm to detect and defend P2P networks from DDoS |
| Tapsell, Akram & Markantonakis [21] | Security evaluation of Bitcoin P2P networks and identified possible solutions to the identified weaknesses and vulnerabilities |
| Kotey, Tchao & Gadze [17] | Evaluation discussion on current DDoS defence mechanisms along with strengths & weaknesses |
| Suresh & Anitha [16] | Machine learning algorithms evaluation to detect DDoS attacks |
| Baldini & Amerini [11] | DDoS detection using novel algorithm with sliding window and MFD (morphological fractal dimension) |
| Alduailij, Mona, Khan, Tahir, Sardaraz, Alduailij, Mai & Malik [5] | Method of detecting DDoS attacks in cloud computing |
| Saranya, Sridevi, Deisy, Chung & Khan [14] | Performance analysis on ML algorithms on IDS in applications such as IoT, big data |
| Hoon, Yeo, Azam, Shunmugam & De Boer [13] | Critical review of ML approaches on DDoS forensics and big data analytics |
| Amarnath, Babu & Prasad [9] | Meticulously tuning hyperparameters with ML model to improve performance of detecting DDoS attack data |
| Pei, Chen & Ji [22] | DDoS attack detection method based on machine learning, which includes two steps: feature extraction and model detection |
| Li, Liu, Zhai & Chen [15] | DDoS attack detection based on deep belief network feature extraction and LSTM model |

3 Research Methodology

The main objective of this research was to analyse the accuracy of a sample of machine learning algorithms in the use of detecting DDoS data and compare the results and their training times. The research used an appropriate dataset, edited it for appropriate testing and employed machine learning algorithms on the dataset so that the results could be compared in respect of respect to accuracy, precision, recall and

f-1 score. The training time was also recorded for this research. This section describes the steps.

3.1 Dataset

The dataset used in this study is an open-source dataset available on the Kaggle dataset repository website. It was made up of an amalgamation of DDoS traffic extracted from three different Canadian Institute for Cybersecurity (CIC) datasets¹⁸ for the purpose of mimicking real-world network traffic. These were combined into a single dataset with DDoS and Benign data observation traffic [9] [11] [5]. The dataset contains 85 features and just over 12.7 million datapoints (12794627 rows × 85 columns) and while large volumes of data can be advantageous, the main disadvantage is data processing capability in a hardware environment. Large volumes of data require more processing power and processing time. Due to hardware constraints in my lab environment, I have used a sample of 8000 observations. The sample consisted of 4000 randomly selected Benign label and DDoS label observations each.

3.2 Data Pre-processing and feature extraction

As many machine learning algorithms are unable to work on label data directly, the raw data was transformed into a usable format. This is part of the pre-processing and is an important step due to the need for ML algorithms to operate on numeric data. Categorical data was altered into a discrete form of 0 for benign and 1 for the DDoS entries. The unnecessary string data was dropped. As this was a large dataset to begin with, any null values were dropped and only features that were needed remain. A smaller dataset would have required a different action to pad out missing values however this was not the case here. Features with little or one value were removed as were features with missing values more than 50% due to the volume of the dataset.

Pearson's correlation coefficient was used to visually see a linear correlation between the features in the dataset. The correlated test determined features that were related and how they would balance each other. Figure 2 displays sample test.

¹⁸ DDoS Dataset DDoS Balanced & Unbalanced Datasets <https://www.kaggle.com/datasets/devendra416/ddos-datasets>. The base Datasets are available at; CSE-CIC-IDS2018-AWS: <https://www.unb.ca/cic/datasets/ids-2017.html>, CICIDS2017: <https://www.unb.ca/cic/datasets/ids-2018.html>, CIC DoS dataset(2016) : <https://www.unb.ca/cic/datasets/dos-dataset.html>

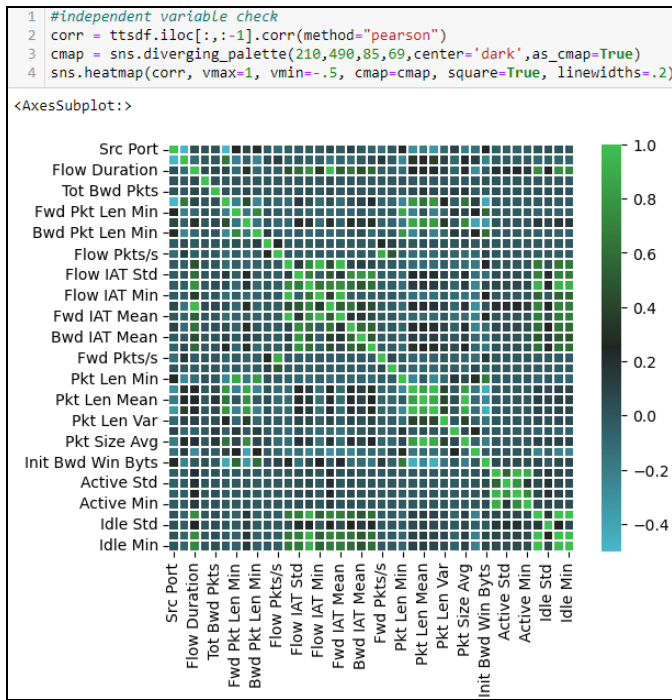


Figure 2 The scatter plot showing the positive correlation between 2 features (TotLen Fwd Pkts & TotLen Bwd Pkts)

The use of scatter plot was used to visually identify trends in the dataset and to determine correlation between different features in the dataset in the process of feature extraction.

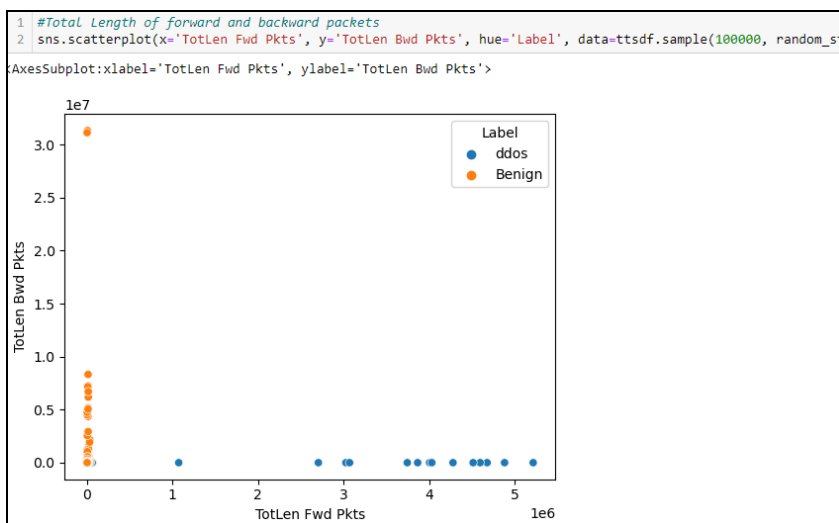


Figure 3 The scatter plot showing the positive correlation between 2 features (TotLen Fwd Pkts & TotLen Bwd Pkts)

The results indicated the positive and negative correlations between the various features in the dataset which then allowed a better understanding of their relationship. Therefore, the features of interest only were extracted using a python function which had a strong relationship with the target variable (the 'Label' feature). Table 2 outlines the features of interest.

Table 1: Features of interest of the dataset

| Number | Feature | Description |
|--------|-------------------|--|
| 1 | Flow Duration | Duration of transmission flow |
| 2 | Src IP | Source IP address |
| 3 | Src Port | Source Port number |
| 4 | Dst IP | Destination IP address |
| 5 | Dst Port | Destination Port number |
| 6 | Tot Fwd Pkts | Total transmitted packets |
| 7 | Init Bwd Win Byts | Number of transmitted bytes |
| 8 | Protocol | Type of Protocol |
| 9 | Label | Attack classification Labels - Class label which indicates whether the traffic type is benign (0) or malicious (1) |

The features are 9 in number at this stage, however the observations were too many for processing. At this stage I scrambled the dataset to produce 8000 observations encompassing an even split between benign and DDoS entries in the resulting balanced dataset.

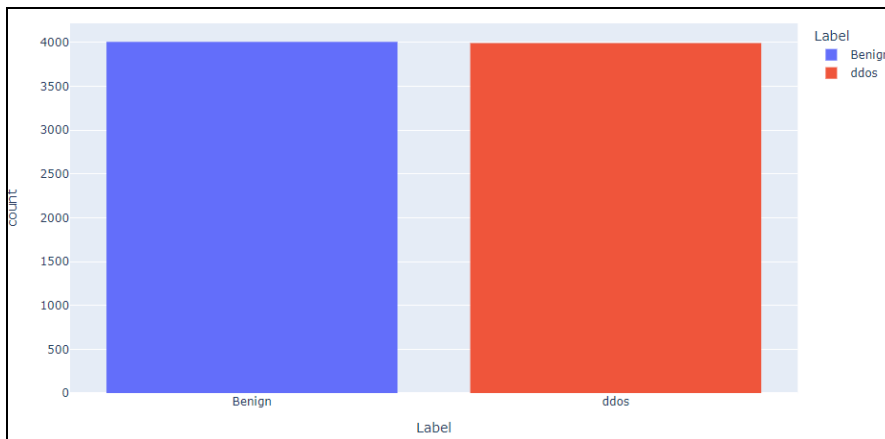


Figure 4 The distribution is an even spread of benign and ddos classes in the balanced dataset

Once the processing of the data was completed then the remaining was saved as a separate dataset that contained numerical data. The new file (*preprocessed.csv*) was a manageable size for the ML models to deal with.

3.3 Training the Models

After performing the pre-processing on the dataset, the data was trained using Naïve Bayes, Decision Tree, K Nearest Neighbour, Random Forest and Logistic Regression classification models in order to do a comparative analysis.

3.4 Evaluation Metrics

The performance of the classification models were evaluated using the values reported by the confusion matrix along with the time model took to train (measured in CPU time).

| | | ACTUAL | |
|------------|----------|-------------------|-------------------|
| | | Negative | Positive |
| PREDICTION | Negative | TRUE NEGATIVE | FALSE NEGATIVE |
| | Positive | FALSE POSITIVE | TRUE POSITIVE |

Confusion Matrix

Figure 5 Graphical view of the confusion matrix

This reported the number of True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN) for each model.

The use of the classification report was used to evaluate the Accuracy, Precision, Recall And F-1 values for each model.

Accuracy measures the proportion of correctly classified network traffic. The proportion of the correct results achieved = $(TP+TN)/(TP+TN+FP+FN)$. Accuracy will detect presence or absence of ddos traffic.

Precision measures the proportion of predicted attacks that were actual attacks, which helps to determine the consistency of the model in detecting the presence ddos traffic. The proportion of returned positive that is actually positive = $TP/(TP+FP)$

Recall measures the proportion of actual attacks that were accurately predicted, or true positive rate, reflecting how capable the models are at identifying malicious traffic. The proportion of actual positives returned = $TP/(TP+FN)$

F1-score can be the most trustworthy because it balances tradeoffs between precision and recall. Mathematically it is defined as $2 * TP / (2 * TP + FN + FP)$. It computes the harmonic mean of the precision and recall. A high F1 score can mean the ddos traffic is correctly being identified and there are low false alarms.

| | |
|------------------|--|
| Precision | $Precision = \frac{\text{Number of True Positives}}{\text{Number of Predicted Positives}}$ |
| Recall | $Recall = \frac{\text{Number of True Positives}}{\text{Number of Actual Total Positives}}$ |
| Accuracy | $Accuracy = \frac{\text{Number of True Positives} + \text{True Negatives}}{\text{Total Observations}}$ |
| F1-Score | $F1 - Score = 2 \frac{Precision \times Recall}{Precision + Recall}$ |

Figure 6: Outline of the procedures for calculating Precision, Recall, Accuracy & F1 score

4 Design Specification

The hardware and software specification is described in section two of the configuration manual subsequently, in this section, an outline of the steps of the research are covered and seen in Figure 4 below.

There were essentially 3 steps to this research:

- Data acquisition and pre-processing
- Classification model training
- Classification model evaluation on the specified metrics

The model was developed in python code using Jupyter notebook platform. The initial dataset was loaded on the platform and from there it was processed so that it was manageable from the classification model's perspective. That included filling in any null values, dropping unnecessary features and cutting down the observation count to a more manageable size for processing due to hardware limitations.

Before applying the machine learning classification models, the data set randomised and then was split into training data and testing data using stratified sampling. The split was 80% of the data was selected as the training data and 20% was selected as the testing dataset. The Scikit-Learn library was utilized to implement the machine learning algorithms. The algorithms implemented were Naive Bayes, K-nearest neighbour, Logistic Regression, Decision Tree and Random Forest. The results were then analysed for the evaluation and compared accuracy, precision, recall and F1 scores.

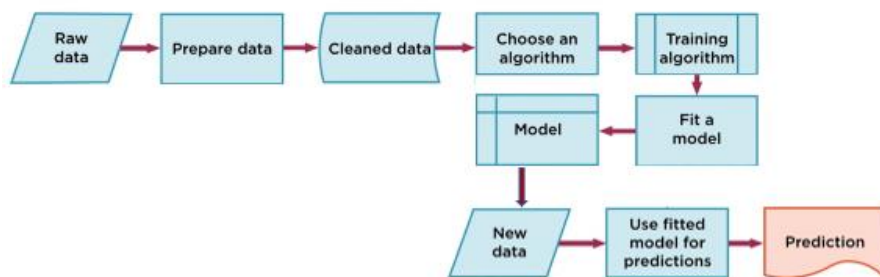


Figure 7 The outline of the design of the research

5 Implementation

The implementation details are outlined in the Configuration Manual.

This research project implementation used the Python programming language on the Jupyter notebook platform utilising packages such as NumPy, pandas, Sklearn, Matplotlib, and Seaborn libraries. The hardware environment was run on 64bit Windows operating system running Intel i5-7300 CPU and 16GB RAM.

The initial raw datafile, final_dataset.csv, of 12 million plus observations was imported and converted to a more readable format to begin with. The data was then checked for various discrepancies such as null values or missing values and cleaned appropriately. Pre-processing and feature selection was carried out on the dataset before a subset of 8000 was randomly selected and saved out to a smaller csv file named ‘preprocessed_dataset.csv’. This new smaller dataset was used for the classification model training and testing due to hardware processing limitations.

The smaller dataset was then split into two parts, 80% for training and 20% for testing

Splitting the data set then into train and test datasets. Then the classification models were trained and tested before all results were gathered and evaluated for time, accuracy, precision, recall and f1-score.

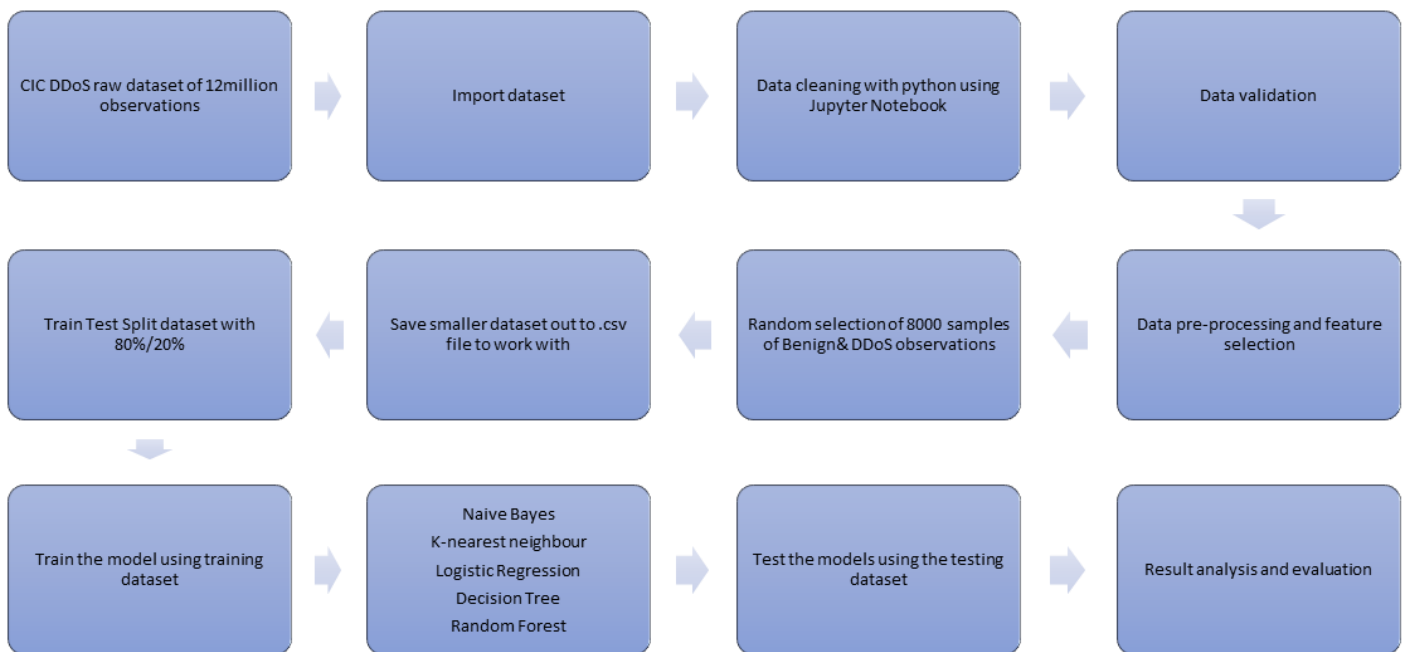


Figure 8 Event Flow Diagram

6 Evaluation

This segment describes the experiments conducted in the research and presents an analysis on the results obtained in the classification model scenarios.

From the results obtained, the conclusion found that Random Forest, followed by and K-Nearest Neighbour, achieved best results on accuracy, precision, recall and F1 score. However, Random Forest was the slowest when it came to training time. Random Forest training time came in at 422ms CPU time whereas the fastest, KNN, came in at 15.6ms and Naïve Bayes, Logistic Regression and Decision Tree came in at 31.2.

6.1 Experiment 1 Accuracy Comparison

Accuracy is a classification mode’s ability to correctly detect the presence of DDoS samples in the dataset. It is the true positives and true negatives. The scikit-learn metrics

library was used to calculate the accuracy. The below table displays the accuracy scores for the classification models. The two highest performing models were Random Forest at .998% accuracy followed by K-Nearest Neighbour at .958% accuracy. Decision Tree and Naïve Bayes followed with .998% and .845% respectively but Logistic Regression performed least well with .496% accuracy.

Table 2: Accuracy scores for the classification models

| | Naive Bayes | K-nearest neighbour | Logistic Regression | Decision Tree | Random Forest | Average |
|-----------------|--------------------|----------------------------|----------------------------|----------------------|----------------------|----------------|
| Accuracy | 0.845 | 0.958 | 0.496 | 0.915 | 0.998 | 0.84 |

6.2 Experiment 2 Precision Comparison

Table 3: Precision scores for the classification models

| | Naive Bayes | K-nearest neighbour | Logistic Regression | Decision Tree | Random Forest | Average |
|------------------|--------------------|----------------------------|----------------------------|----------------------|----------------------|----------------|
| Precision | 0.870 | 0.994 | 0.500 | 0.928 | 0.997 | 0.86 |

The precision measurement is largely designed to check for false positive values and as can be seen in Table 3 the highest precision score achieved was .997 for Random Forest, .994 for KNN, .928 for Decision Tree and Naïve Bayes at .850. The lowest precision was achieved by Logistic Regression at .500.

6.3 Experiment 3 Recall Comparison

Recall records how well the model is predicting DDoS from the dataset when it really is a DDoS observation. From Table 4, the 2 highest performing models were Random Forest and KNN with a score of .997 each. Decision Tree scored .900 and Naïve Bayes recorded .850. Logistic Regression classification model again had a modest .500.

Table 4: Recall scores for the classification models

| | Naive Bayes | K-nearest neighbour | Logistic Regression | Decision Tree | Random Forest | Average |
|---------------|--------------------|----------------------------|----------------------------|----------------------|----------------------|----------------|
| Recall | 0.850 | 0.997 | 0.500 | 0.900 | 0.997 | 0.85 |

6.4 Experiment 4 F-1 Comparison

F1 score is defined as the harmonic mean of precision and recall and is measured in a range of between 1 and 0 with 1 being a near perfect model for identification of an observed class and 0 signifying a model unable to identify accurately. As can be viewed on Table 5 below, Random Forest & K-NN perform well within the statement goal of predicting 95% or greater with a reading of .998 and .989 respectively. The Decision Tree and Naïve Bayes model again followed next with a .911 and .840 score respectively but Logistic Regression model falling closer to 0 with an F-1 score of .351.

Table 5: F1-scores for the classification models

| | Naive Bayes | K-nearest neighbour | Logistic Regression | Decision Tree | Random Forest | Average |
|-----------------|-------------|---------------------|---------------------|---------------|---------------|---------|
| F1 score | 0.840 | 0.989 | 0.351 | 0.911 | 0.998 | 0.82 |

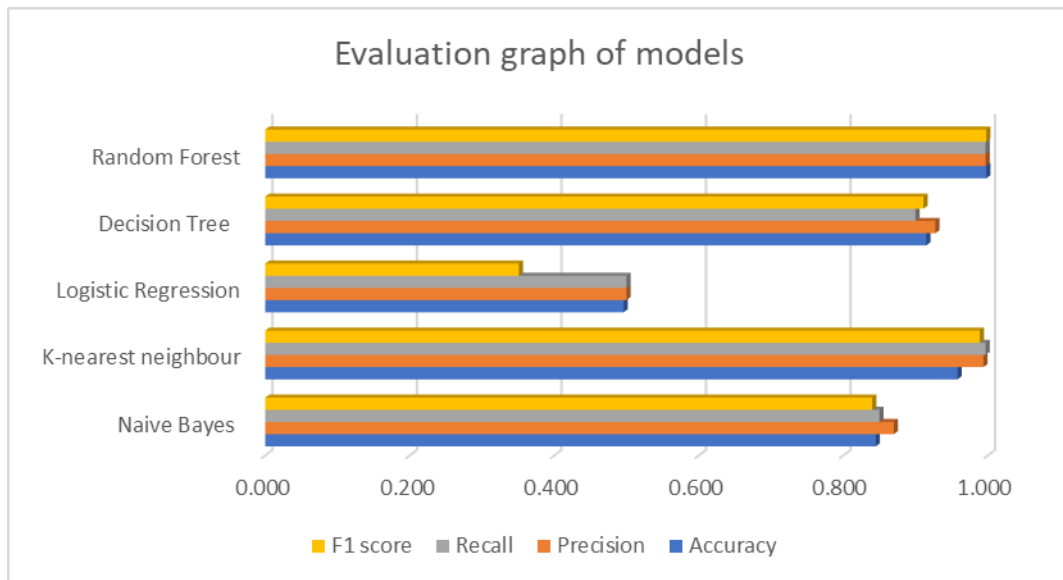


Figure 9 Graph overview of each classification model performance for Accuracy, Precision, Recall & F1-score

Figure 9 shows the performance graph of the overall performance of the classification models with Random Forest near the top for F1-score, accuracy, recall and precision. Easily visible from the graph is the poor performance of the Logistic Regression model and is quite notable in comparison to the other models in the graph.

6.5 Experiment 5 Time

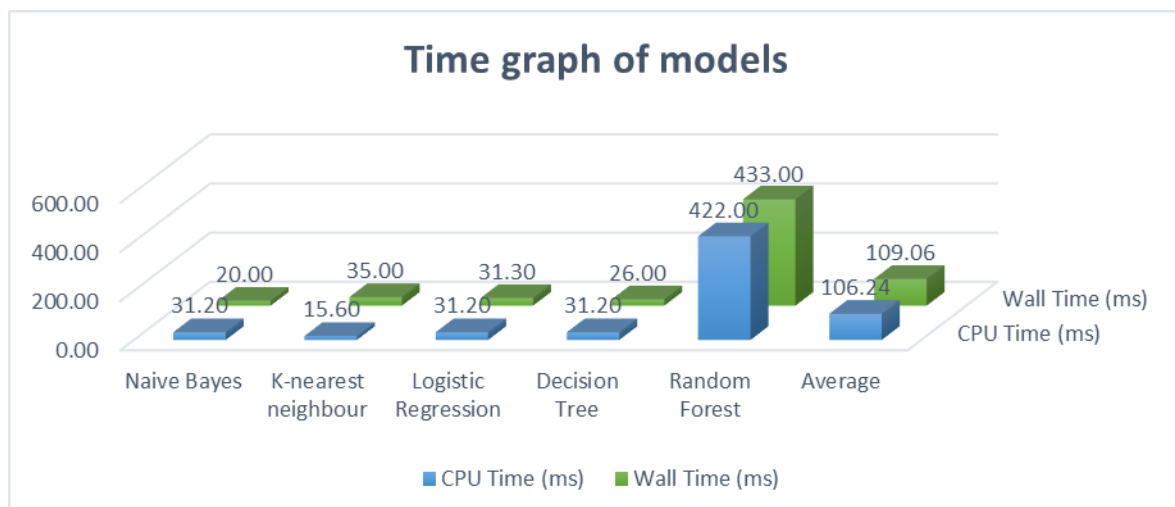


Figure 8 Training time overview

Table 6 outlines the CPU training time of each classification models. The best time performer was KNN outperforming Naïve Bayes, Logistic Regression and Decision Tree by 50% taking 15.6ms over 31.2ms CPU time for each of the other three models.

Table 6: CPU and Wall Time in ms for the classification models

| | Naive Bayes | K-nearest neighbour | Logistic Regression | Decision Tree | Random Forest | Average |
|-----------------------|-------------|---------------------|---------------------|---------------|---------------|---------|
| CPU Time (ms) | 31.20 | 15.60 | 31.20 | 31.20 | 422.00 | 106.24 |
| Wall Time (ms) | 20.00 | 35.00 | 31.30 | 26.00 | 433.00 | 109.06 |

However, the poorest performer of training time lay with the Random Forest model taking 422.0ms CPU time which was substantially more than any of the other models and this is much more notable in the Figure 8 graph.

7 Conclusion and Future Work

The aim of the experiment was to analyse the accuracy of a sample of machine learning algorithms in the use of detecting DDoS data and compare those results. The research used an applicable dataset, edited it for appropriate testing and employed machine learning algorithms on the cleaned dataset so that the results could be compared in respect of accuracy, precision, recall and f-1 and the training times of each. Out of the five ML algorithms assessed, Random Forest and K-NN were the only two classification models that achieved the problem statement goal of predicting 95% or greater. The Random Forest classifier performed the best overall in the test, achieving .998% followed by K-NN achieving .958%. Logistic Regression performing least favourably with a 50% accuracy with Naïve Bayes and Decision Tree having a 85% and 92% respective accuracy percentage rates.

Training time performance showed that the Random Forest classification model perform least favourably with a time of 422ms recorded which was multiple times slower than that of all of the other classification models with KNN perform the fastest with 15.6ms and the remaining having a time of 31.2ms each.

Even though Random Forest did perform best overall of the other classification models, the large training time would render it unsuitable as a model solution when it came to new datasets and so an alternative model should be chosen in this scenario.

Limitations of hardware resources and time did not allow the possibility of using a more realistic dataset size and complexity for assessment. In future work a more varied realistic larger dataset could be used to assess these classification models which may result in a different set of performance results for the classification models.

8 References

- [1] Curzon, James, Tracy Ann Kosa, Rajen Akalu, and Khalil El-Khatib. 'Privacy and Artificial Intelligence'. *IEEE Transactions on Artificial Intelligence* 2, no. 2 (April 2021): 96–108. <https://doi.org/10.1109/TAI.2021.3088084>. Citing = 1
- [2] M. Xue, C. Yuan, H. Wu, Y. Zhang and W. Liu, "Machine Learning Security: Threats, Countermeasures, and Evaluations," in *IEEE Access*, vol. 8, pp. 74720-74742, 2020, doi: 10.1109/ACCESS.2020.2987435. Citing = 35
- [3] Pantridge, Michael. 'NCIRL Penetration Testing Module January 2021'. Lecture 2 Slides 67 January 2021. Source: <https://devcentral.f5.com/articles/the-ascendancy-of-the-application-layer-threat>.
- [4] Brooks, Chuck. 'Alarming Cybersecurity Stats: What You Need To Know For 2021'. Accessed 14 February 2022. <https://www.forbes.com/sites/chuckbrooks/2021/03/02/alarming-cybersecurity-stats---what-you-need-to-know-for-2021/>.
- [5] Alduailij, Mona, Khan, Q.W., Tahir, M., Sardaraz, M., Alduailij, Mai, Malik, F., 2022. Machine-Learning-Based DDoS Attack Detection Using Mutual Information and Random Forest Feature Importance Method. *Symmetry* 14, 1095. <https://doi.org/10.3390/sym14061095> Citing = 2
- [6] National Cyber Security Centre. 'Ransomware Attack on HSE Network'. National Cyber Security Centre, 14 May 2021. https://www.ncsc.gov.ie/pdfs/HSE_Conti_140521.pdf.
- [7] Mahadevappa, P., Muzammal, S.M., Murugesan, R.K. 'A Comparative Analysis of Machine Learning Algorithms for Intrusion Detection in Edge-Enabled IoT Networks', n.d., 6. Citing =
- [8] Jazi,H.H, Gonzalez, H., Stakhanova, N., Ghorbani, A.A. "Detecting HTTP-based application layer DoS attacks on web servers in the presence of sampling" | Elsevier Enhanced Reader [WWW Document], n.d. <https://doi.org/10.1016/j.comnet.2017.03.018>
- [9] Amarnath. C, Babu. P., Prasad, M.D., 2019. "Machine Learning DDoS Detection Using Stochastic Gradient Boosting". *Int. J. Comput. Sci. Eng.* 10.
- [10] Ali, Abdinur, Yen-Hung Hu, Hsieh Chung-Chu (George), and Mushtaq Khan. 'A Comparative Study on Machine Learning Algorithms for Network Defense', 2017. <https://doi.org/10.25778/PEXS-2309>. Citing = 9
- [11] Baldini, G., Amerini, I., 2022. "Online Distributed Denial of Service (DDoS) intrusion detection based on adaptive sliding window and morphological fractal dimension". *Computer Networks* 210, 108923. <https://doi.org/10.1016/j.comnet.2022.108923> Citing = 0
- [12] Cybersecurity & Infrastructure Security Agency (CISA) "Understanding and Responding to Distributed Denial of Service Attacks," n.d., 9. <https://www.cisa.gov/uscert/ncas/current-activity/2022/10/28/joint-cisa-fbi-ms-isac-guide-responding-ddos-attacks-and-ddos> Original release date: October 28, 2022 Citing =
- [13] Hoon, K.S., Yeo, K.C., Azam, S., Shunmugam, B., De Boer, F., 2018. "Critical review of machine learning approaches to apply big data analytics in DDoS forensics", in: Presented at the 2018 International Conference on Computer Communication and Informatics (ICCCI), IEEE, Coimbatore, pp. 1–5. <https://doi.org/10.1109/ICCCI.2018.8441286> Citing = 0
- [14] Saranya, T., Sridevi, S., Deisy, C., Chung, T.D., Khan, M.K.A.A., 2020. "Performance Analysis of Machine Learning Algorithms in Intrusion Detection System". *Procedia Comput. Sci.*, Third

International Conference on Computing and Network Communications (CoCoNet'19) 171, 1251–1260. <https://doi.org/10.1016/j.procs.2020.04.133> Citing = 0

[15] Li, Y., Liu, B., Zhai, S., Chen, M., 2019. "DDoS attack detection method based on feature extraction of deep belief network". IOP Conf. Ser.: Earth Environ. Sci. 252, 032013. <https://doi.org/10.1088/1755-1315/252/3/032013> Citing=2

[16] Suresh, M., Anitha, R., 2011. "Evaluating Machine Learning Algorithms for Detecting DDoS Attacks", in: Wyld, D.C., Wozniak, M., Chaki, N., Meghanathan, N., Nagamalai, D. (Eds.), *Advances in Network Security and Applications, Communications in Computer and Information Science*. Springer, Berlin, Heidelberg, pp. 441–452. https://doi.org/10.1007/978-3-642-22540-6_42 Citing =

[17] Kotey, S., Tchao, E., Gadze, J., 2019. "On Distributed Denial of Service Current Defense Schemes". *Technologies* 7, 19. <https://doi.org/10.3390/technologies7010019>

[18] Zargar, S.T., Joshi, J., Tipper, D., 2013. "A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks". *IEEE Commun. Surv. Tutorials* 15, 2046–2069. <https://doi.org/10.1109/SURV.2013.031413.00127> Citing = 752

[19] Alawida, M., Omolara, A.E., Abiodun, O.I., Al-Rajab, M., 2022. "A deeper look into cybersecurity issues in the wake of Covid-19: A survey". *Journal of King Saud University - Computer and Information Sciences* 34, 8176–8206. <https://doi.org/10.1016/j.jksuci.2022.08.003> Citing = 3

[20] Jaideep, G., Prakash Battula, B., 2018. "Detection of DDOS attacks in distributed peer to peer networks". *IJET* 7, 1051. <https://doi.org/10.14419/ijet.v7i2.7.12227>. Citing = 11

[21] Tapsell, J., Naeem Akram, R., Markantonakis, K., 2018. "An Evaluation of the Security of the Bitcoin Peer-To-Peer Network", in: 2018 IEEE International Conference on IoT & IEEE Green Computing and Communications (GreenCom) & IEEE Cyber, Physical & Social Computing (CPSCoM) & IEEE Smart Data (SmartData). Presented at the 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCoM) and IEEE Smart Data (SmartData), IEEE, Halifax, NS, Canada, pp. 1057–1062. https://doi.org/10.1109/Cybermatics_2018.2018.00195. Citing=9

[22] Pei, J., Chen, Y., Ji, W., 2019. "A DDoS Attack Detection Method Based on Machine Learning". *J. Phys.: Conf. Ser.* 1237, 032040. <https://doi.org/10.1088/1742-6596/1237/3/032040> Citing=13