ES CHAKI

IMPLEMENTING A MACHINE LEARNING-BASED (ML-BASED) HYBRID MODEL TO MITIGATE DISTRIBUTED DENIAL OF SERVICE (DDoS) ATTACKS IN MOBILE EDGE COMPUTING (MEC)

2023

IMPLEMENTING LEARNING-BASED (ML-BASED) HYBRID MODEL TO MITIGATE DISTRIBUTED DENIAL OF SERVICE (DDoS) ATTACKS IN MOBILE EDGE COMPUTING (MEC)

MASTER OF SCIENCE IN COMPUTER SCIENCES

ES CHAKI

IMPLEMENTING LEARNING-BASED (ML-BASED) HYBRID MODEL TO MITIGATE DISTRIBUTED DENIAL OF SERVICE (DDoS) ATTACKS IN MOBILE EDGE COMPUTING (MEC)

by

EMMANUEL SIBUSISO CHAKI

DISSERTATION / THESIS

Submitted in fulfilment of the requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE

in the

FACULTY OF SCIENCE & AGRICULTURE

School of Mathematical and Computer Science

at the

UNIVERSITY OF LIMPOPO

Supervisor: Professor MTHULISI VELEMPINI

Co-Supervisor: MR SEKGOARI SEMAKA MAPUNYA

2023

DECLARATION

I **Emmanuel Sibusiso CHAKI**, declare that the Dissertation hereby submitted to the University of Limpopo, for the degree of **Master of Science in Computer Science** has not previously been submitted by me for a degree at this or any other university; that it is my work in design and in execution, and that all material contained herein has been duly acknowledged.

Chaki ES **2023/09/21**

Surname, Initial (Mr) **Date**

ACKNOWLEDGEMENTS

I am grateful to every person who supported me to complete this dissertation.

- Primarily, I appreciate and thank the divine presence of God, who bestowed upon me the necessary strength to overcome challenges and successfully finalise this dissertation.
- I extend my appreciation to my supervisor, Professor Mthulisi Velempini, and my co-supervisor, Mr. Mapunya Sekgoari Semaka, for their invaluable assistance, guidance, encouragement, and support.
- I express my gratitude for the sponsorship received from SenTech throughout my Master of Science in Computer Sciences.
- I am grateful for the support I received from my parents (Queen Maria Chabalala) throughout my educational journey.
- Lastly, I thank the University of Limpopo for allowing me to further my studies.

ABSTRACT

5G technology constitutes a considerable part of solving the problem of security in mobile communications. Multi-Access Edge Computing or mobile edge computing (MEC) extends the capabilities of cloud computing by locating them near the edge of the network. By outsourcing cloud processing to specific local servers, MEC decreases latency in 5G, thereby improving the end-user experience. This study explores a security vulnerability present in 5G MEC. Specifically, we examined distributed denial of service (DDoS) attacks occurring at both the network and the application layer. The vulnerability of MEC to DDoS attacks poses significant challenges that are addressed in this research. We evaluated different Machine Learning (ML) algorithms and subsequently implemented hybrid models (Stacking/Blending, and Random Forests (RF) model) which are classified under supervised ML. The purpose of this study is to identify the most effective techniques for mitigating DDoS attacks in MEC systems.

ML techniques such as Random Forest (RF), Decision tree (DT), Naïve Bayes (NB), K-Nearest Neighbour (K-NN), Logistics regression (LR), Blending/Stack Model are evaluated on the basis of a variety of performance metrics (including accuracy, detection/recall, precision, F1-Measure, Matthews correlation coefficient (MCC), Receiver operating characteristic (ROC), and Area Under Receiver operating characteristic (AUROC)) for each of the algorithms. Probability density function (PDF) and hypotheses testing are statistical techniques deployed to support the findings of our study.

Based on the literature in the field, ML techniques are recommended to reach our solution. The best ML algorithms yielding the best performance in mitigating the DDoS attacks are optimized to enhance their performance ability. This study outlines the overview of MEC environment’s existing mitigation scheme, and the implemented mitigation schemes towards DDoS attacks. According to our evaluated findings, Hybrid models outperformed ML models based on the computed scores of performance metrics. PDF and hypotheses testing successfully supported our findings by showing that hybrid models indeed outperformed ML models. Among the mitigation techniques, RF outperformed all supervised ML models by effectively mitigating DDoS attacks in MEC.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: RESEARCH PROPOSAL

1.1 INTRODUCTION AND BACKGROUND

Among five technologies (millimetre-wave, Massive multi-user Multiple-Input Multiple-Output (MIMO), Small cell stations, Beamforming, and Non-Orthogonal Multiple Access (NOMA)) of Fifth Generation (5G) technologies, Mobile Edge Computing (MEC) is considered a key technology critical to cellular communication. From these 5G technologies, our study focused on MEC. In the year 2020, the deployment of the first phase of 5G mobile network began, and as compared to the fourth generation (4G), it is expected to be ten times faster. 5G is expected to exhibit higher peak download speeds and more efficient bandwidth usage in comparison to 4G [1]. 5G supports a wide variety of platforms, including cloud systems, augmented reality, industrial automation, mission-critical apps, and self-driving cars.

Increasing the efficiency, stability, and integrity of a 5G network is one of the primary goals of MEC. The successful implementation of this technology depends on security, because if it is vulnerable to attacks such as Distributed Denial of Service (DDoS) attacks, then it implies that the end-user experience becomes compromised. The security device is used to defend against attacks from the enterprise network towards the carrier network. Defending against attacks from the carrier network to the enterprise network is performed through using the security device on the enterprise network. Network quality services can be improved by enhancing security measures. Secure application systems require high security measures in conjunction with 5G technologies.

Security attacks against MEC are the focus of this study. MEC has been subjected to nine major security attacks namely: Man-in-the-Middle (MITM), Phishing, Spear phishing, SQL injection, Cross-site scripting (XSS), Eavesdropping, Malware, Denial of Services (DoS), and Distributed Denial of Service (DDoS) attacks [2]. This study focuses on DDoS attacks among nine attacks mentioned and seeks to develop a mitigation scheme that deals with such attacks. DDoS attacks aim to restrict the functionality of a program or impede legitimate access to networks, systems, and applications. A DDoS attack thus refers to a scenario where multiple compromised computers collaborate to target a server, website, or any other network resource, resulting in service disruption for other users.

In MEC, DDoS attacks, service interruptions and disruptions are some of the biggest concerns [3]. The DDoS attack either targets the network layer or the application layer. The purpose of this study is to compare the effectiveness of the following ML algorithms: NB, RF, KNN, LR, DT, and Hybrid-model (Blending or Stacking Model) in mitigating the disruptions. A performance evaluation is conducted for their capacity to mitigate interruptions based on the following metrics namely: accuracy, recall/detection, precision, and F1-measure, Mathew correlation coefficient (MCC), Receiver operating characteristics (ROC), and Area under receiver operating characteristics (AUROC).

We also configured analysis of ML algorithms to find the optimal model. Performance of ML models is directly affected by the hyper-parameter configuration. In that case, the best optimal ML algorithms in terms of mitigating the DDoS attacks is evaluated. The existing ML algorithms (LR, DT, RF, and NB) have been ensembled to design a hybrid-model to mitigate DDoS attacks.

1.2 RESEARCH PROBLEM

The fundamental requirements of the 5G technology are privacy and security [4]. The level of trust than mobile users have when using their devices and applications is determined by the level of security in the technology. MEC technology encompasses a network structure that expands the potential of cloud computing and IT services by bringing them closer to the network's edge. Unfortunately, due to several devices connecting from the edge, MEC is subjected to security challenges, including DDoS attacks. Malicious users can also deploy their devices and systems to the MEC, thereby compromising the security.

High-density connections and ultra-low latency are some of the key features of 5G, which can accommodate Fourth Industrial Revolution (4IR) technologies such as robotics, the Internet of Things (IoT), Cloud computing, edge computing and self-driving vehicles [4]. However, in all these developments, security remains a challenge. Researchers have addressed many technical concerns of 5G technology and MEC environment, and all have reached the conclusion that security and privacy require further attention. DDoS attacks on edge computing have been identified as a threat to MEC availability; therefore, urgent attention is required to mitigate this phenomenon.

The mitigation of DDoS attacks involves several technical challenges such as a distributed attack, several compromised hosts, called zombies, that are used to launch a simultaneous attack, making it difficult to identify and defend against the attackers. Malicious packets can modify their own patterns to evade detection. This is accomplished through mimicking normal traffic and hiding IP address. The majority of DDoS attacks employ IP address spoofing to conceal the attacker's identity from the target, evade traceback, or execute reflector attacks. Additionally, these attacks often mimic regular traffic patterns. Consequently, during the attack detection process, some legitimate packets may be incorrectly identified as malicious and rejected, while certain malicious packets may enter the edge computing environment without being detected.

A thorough assessment of this area of compromised security is necessary to resolve the above-mentioned issues that affect the quality of edge and computing services. We developed a model that can accurately detect DDoS attacks in edge-based systems.

To develop an effective mitigation plan for detecting DDoS attacks, it is essential to examine the security risks associated with MEC and explore appropriate countermeasures. Existing literature suggests various statistical techniques for detecting DDoS attacks. However, creating a real-time detector for such attacks continues to pose a significant challenge [5]. Although, it is challenging to design a mitigation scheme for DDoS attacks, our study evaluated ML algorithms, and hybrid models using the performance metrics. Thereafter, the existing ML algorithms based on the evaluated results were integrated to design a hybrid model that ultimately improves the mitigation of DDoS attacks.

1.3 MOTIVATION FOR THE STUDY

The motivation for this research is that MEC has become a scalable technique for delivering valuable 5G networks and resources to edge users over the Internet. This study intends to ensure that hackers are unable to generate traffic and transmit it to the Internet servers, preventing edge users from using the networks or internet. The results in literature illustrate and confirm an improvement in the efficiency of ML based schemes. This study implements counter measures (detection of DDoS attacks), which include protecting confidentiality, integrity, and availability of MEC services which are ML-based. A major challenge with cloud computing and adoption lies in ensuring services and resource availability in edge computing. The DDoS attacks on edge computing have been identified as the most challenging threat to MEC availability (such as the presence of high network bandwidth at the network edge). Therefore, further research is required.

ML utilizes accessible data for the purpose of acquiring knowledge. In supervised learning (SL) and artificial intelligence (AI), a computer algorithm undergoes training using labelled input data to produce specific outputs. Through supervised ML, the model is trained to recognize patterns and correlations between input data and corresponding output labels, so that it can enhance results when presented with unfamiliar data. The supervised ML algorithms: RF, NB, KNN, LR and DT have been trained and tested to detect DDoS attacks using DDoS Evaluation Dataset [6].

Our study trains the above-mentioned supervised ML algorithms using the datasets which have been labelled. The reason for not using other ML algorithms such as unsupervised ML, semi-supervised ML, and reinforcement learning are as follows:

i. Unsupervised ML - training set of unlabeled data must be provided to the algorithm, which may achieve incorrect results. In our case, we use labelled datasets.
ii. Semi-supervised ML - the results of semi-SL have been proven to be accurate and can be applied to a wide variety of real-world problems, but the mere fact that it is used for partially labelled data (labelled and unlabelled data) means that supervised ML remains the best for our case.
iii. Reinforcement ML - aims to maximize rewards through autonomous interpretation and learning from trial and error experience.

Our study uses supervised ML algorithms because it provides accurate results, and is the best in the predicting category. Supervised ML algorithms are ideal for ensembling algorithms, which is the goal of our study, as the implementation of any hybrid model requires the process of ensembling or intergrating two or more algorithms.

1.4 RESEARCH AIM

The study aims to implement a hybrid model which incorporates ML techniques to detect DDoS attacks in the MEC environment.

1.5 RESEARCH QUESTIONS

This study provides answers to the following research questions:

- What is the most efficient methodology for evaluating DDoS mitigation schemes?
- Which algorithms are the most effective in mitigating DDoS attacks?
- Which are the best ML algorithms to stack for improving detection efficiency of DDoS attacks?
- What design factors contribute to an efficient DDoS attacks detection scheme?

1.6 RESEARCH OBJECTIVES

The study's objectives are designed to:

- Evaluate and compare existing DDoS attack mitigation schemes.
- Evaluate DDoS security schemes in mitigating the DDoS attacks.
- Design a hybrid model which incorporates ML techniques to detect the DDoS attacks.
- Evaluate the performance of the proposed hybrid scheme.

1.7 BRIEF LITERATURE REVIEW

Due to skilled attackers that use various methods to flood the network with DDoS attacks, implementing mitigation systems for application layer DDoS attacks has become a complex task. IoT devices that are compromised are increasingly vulnerable to application-level DDoS attacks. Existing solutions are ineffective due to the apparent legitimacy in such attacks. Consequently, mitigation is extremely costly due to the large volume and dispersion of the resulting traffic [7]. ShadowNet was developed as a way to mitigate application-level IoT-DDoS attacks by exploiting emergent edge technology.

According to some of the reviewed paper [8], the DDoS attacks have grown in popularity. They pose a significant menace to the Internet that has intensified due to the escalation of attack traffic, consuming in the process large amounts of bandwidth or computing resources. As a consequence, DDoS attack tools have become more widely available. This paper [8] proposed cooperative reinforcement learning with Hidden Markov Models (HMMs), principally developed to optimize the detection rate, as well as to ensure accuracy in terms of dealing with DDoS attacks.

Sood et el., in paper [9], based on recent trends, demonstrated that DDoS attacks constitute most network attacks. Network systems face challenges in accurately differentiating between legitimate and malicious traffic. To address this issue, the authors proposed a mitigation scheme that utilizes real-life intrusion detection datasets to train and evaluate ML-based classifiers. Mood et al., have carried out research that illustrates that the RF classifier outperforms other options for DDoS detection. Furthermore, the experimental results indicate that the accuracy achieved in their trials exceeds 96 percent when tested on a real-world dataset.

ML models are being applied to detect DDoS attacks. Their work is motivated by two questions: What is the most accurate supervised learning (SL) algorithm to detect DDoS attacks? How accurate would these algorithms be if trained on real-life data? To support their findings, they presented a detailed analysis.

Mamolar *et el.*, in [10] highlighted that no viable security solution existed to effectively detect cyber-attacks on 5G networks. The presence of unique characteristics such as multitenancy and user mobility presented considerable challenges for existing security solutions in addressing DDoS threats. However, their findings were useful to 5G users, since they proposed a transversal detection system to protect tenants, infrastructure, and 5G users simultaneously in the edge and core network segments of the 5G multi-tenant infrastructure at the same time.

Gupta * et al., in paper [11] demonstrated that DDoS attacks represent a type of network security threat, and attackers have expanded their reach to target various technologies such as cloud computing, IoT, and edge computing to enhance their capabilities. While several defensive measures have been proposed, their effectiveness is limited due to attackers leveraging automated tools for training. In light of this, the authors proposed a ML-based classification approach to detect DDoS attacks specifically in cloud computing environments.

Most reviewed papers have used unsupervised, reinforcement, and semi-SL algorithms to tackle the DDoS attacks over the networks such as fog computing [12, 13], and mobile cloud computing (MCC) [14] [12]. The future recommendations of the reviewed papers provide justification for what our study focuses on, which is detecting the DDoS attacks in MEC using the supervised ML algorithms. A hyper-parameter optimization, and hybrid model via combination of optimal algorithms are both used to make certain that dealing with DDoS attacks is successful in terms of detection.

1.8 RESEARCH METHODOLOGY AND ANALYTICAL PROCEDURES

A Jupyter notebook refers to a web-based interactive development environment that uses notebooks, code, and data to enable the development of interactive applications. With its flexible interface, users can configure and arrange data and ML workflows. The modular design facilitates the addition of new features and functionality. ANACONDA3-2021 is a Python and R programming language distribution designed for scientific computing (ML applications, and predictive analytics), which simplifies package management and deployment.

A Jupyter notebook is used to simulate DDoS mitigation algorithms through ANACONDA3-2021. The use of ANACONDA together with Jupyter notebook provides full access to the simulation process and ML-based classification results. The Python programming language, which is commonly used to analyse data in ML approaches, has been utilized in our research. A Jupyter notebook generates the desired results, and this has assisted in the evaluation of mitigation schemes. We have also utilized the Jupyter Notebook to implement the hybrid mitigation scheme.

1.9 SCIENTIFIC CONTRIBUTION

This study outlines the evaluation of existing schemes (RF, NB, LR, KNN, DT, and hybrid model), and the performance of each scheme has been evaluated using the following performance metrics: detection rate/Recall/true positive rate, accuracy, precision, F1-Measure, MCC, ROC, and AUROC. The best and optimal scheme is based on whether the F1-measure rate is high or not, while the accuracy level and detection rate would be high as well. A hybrid scheme has been implemented and evaluated. The hybrid model has been designed to enhance the security system techniques in detecting DDoS attacks. Literature has proven that hybrid models tend to be more efficient in terms of DDoS detection. Hence, the implementation of the hybrid model has been trained, and tested to detect the DDoS attacks.

1.10 OVERVIEW OF THE STUDY

The initial chapter outlined the research problem and justified the need for such a study. Chapter 2 is a review and summary of related works in mitigating cyber-attacks that compromise security on 5G. Chapter 3 presents the details of the simulation tools and selected algorithms developed to address the effects of DDoS attacks. Chapter 4 summarizes and interprets the results generated in this study. Chapter 5 concludes the study.

CHAPTER 2: LITERATURE REVIEW

2.1 INTRODUCTION

In this chapter, we reviewed literature that is recent and related to our research. Based on our critical analysis, we identified gaps that warranted further exploration in our research. We provide here a brief overview of cyber security attacks in the MEC, FOG, IoT and MCC. In most of the cited literature, plausible solutions are provided for such attacks.

2.2 RELATED RESEARCH WORK

The reviewed papers are discussed and categorized in comparison of dealing with DDoS attacks by utilizing different mitigation schemes in edge computing technologies such as FOG, MCC, and other environmental technologies like IoT, and SDN. To be more precise this section entails gaps of MEC DDoS attacks proposed mitigation schemes against investigated or current published mitigations of DDoS attacks in the mentioned technological environments.

2.2.1 MCC vs MEC DDoS attack Mitigations

In a paper in [11], the authors proposed an approach for detecting and preventing DDoS attacks on cloud servers that is based on ML. The results in [11] show that in DDoS attacks, attackers may exploit any technology such as cloud computing, IoT, and edge computing. A DDoS attack uses available resources such as memory, CPU, or network to overwhelm the victim's computer or server. Although numerous defence mechanisms are proposed, they are not as effective as the attackers. Hence, the authors in [11] proposed the statistical mitigation approach and extracted statistical features from the dataset they collected. They observed that their proposed method detected DDoS attacks with a high degree of accuracy of 99.68% and only a few instances of false positives. Their recommendations for future research centre on unsupervised learning (UL) or reinforcement learning (RL) since they only focused on supervised learning (SL) techniques. Instead of concentrating solely on cloud servers, our study compares ML algorithms in MEC. Although these researchers utilized ML algorithms to detect DDoS attacks, the main difference in our study is that they focused on cloud computing while ours focused on MEC. In this regard, our study enhances the investigation by implementing a new hybrid-model mitigation scheme.

In yet another paper [17], the authors provided evidence that sensor edge clouds (SECs) are capable of detecting DDoS attacks even when resource information is incomplete, and the state of virtual machines is unknown. By employing resource allocation strategies and ensuring the fulfilment of task offloading requirements, both the sensor unit and edge virtual machine (VM) work together to safeguard the network against DDoS attacks. Using probability distributions, they characterized the partial knowledge of the resource allocation strategy and the unpredictable states of the edge virtual machine. The researchers developed a formal attack defence model for countering DDoS attacks by employing Bayesian game theory. Through the utilization of a resource allocation technique, they successfully maximized rewards for cooperative defenders, while also achieving marginal distribution and interval. Furthermore, they demonstrated the effectiveness of incorporating ML platform interaction in finding a viable resource allocation approach, utilizing a search technique based on Q-learning. Based on the outcomes of their numerical simulation, they concluded that the Bayesian Q-learning game scheme outperforms alternative defence mechanisms when dealing with imperfect information. As indicated already, their research concentrated on outlining a method for preventing DDoS assaults in a sensor cloud, whereas our study is strictly concerned with MEC.

In a recent paper [22], it is demonstrated that DDoS assault traffic flowing via fog defender can be identified and filtered out with the rules put in place at the network layer, ensuring that only legitimate requests are routed to their cloud server. Considering this, the requests that are sent to the cloud are valid. At the network periphery, rather than in the cloud, DDoS assaults are discovered and countered. Thus, cloud response times and resource use are enhanced. In contrast, this method only defends against HTTP and TCP attack traffic. It might then be improved to protect other protocol traffic, including ICMP and UDP. They demonstrated that by utilizing servers as fog devices, there is a potential to enhance intelligence at the network's edge. This is made possible by their load balancing capabilities and the ability to make swift decisions, particularly beneficial for mission-critical applications.

This paper [12] explores the rise of cloud computing as a robust alternative to conventional IT platforms, characterized by its cost-effectiveness, pay-per-use model, and flexible service provisioning. Organizations and governments have transitioned their entire IT infrastructure to the cloud. However, the surge in IoT devices and big data has led to an exponential increase in data sent to the cloud. Consequently, the traditional cloud computing paradigm is proving inadequate. Moreover, as the demand for IoT solutions in business expands, the ability to process data rapidly, efficiently, and on-site has become increasingly vital.

Consequently, Fog computing has emerged as a solution to tackle the challenges encountered in cloud computing by bringing intelligence closer to the network's edge through smart devices. Additionally, this paper [12] discussed how DDoS affects cloud environments, the use of fog computing in cloud environments to solve various problems, and how to use fog computing in the cloud environment. The difference is that our study focuses on MEC security issues (DDoS attacks), whereas their paper [12] concentrates on cloud and Fog computing security issues (DDoS attacks).

In this paper [14], they suggested methodologies, and their main goal was to successfully identify DDoS attacks by combining many phases such as pre-processing, feature selection, and classification. Their approach commences with a pre-processing step that normalizes all variables to a certain scale range.

The suggested feature selection - whole optimization algorithm (FS-WOA) model - is used to choose the best collection of features from the normalized data. Using a deep learning classifier, both attacked and non-attacked information is categorized. In order to improve system performance, non-attacked data is stored in cloud storage using security standards.

Due to this, the suggested feature selection-whole optimization algorithm-deep neural network (FS-WOA-DNN) model assists applicants in protecting data from DDoS attacks.

Furthermore, the suggested detection technique aids in preventing DDOS attacks from entering large-scale companies.

The suggested model achieves an overall detection accuracy of 95.35 per cent, which is a more efficient measure than current DDoS attack detection models. According to this paper [14], in the future, instead of identifying individual attacks, IDS techniques could be developed for detecting unique attacks. The difference is that this paper [14] utilized a deep learning model (DLM) to detect DDoS attacks in cloud storage applications; whereas our study specifically adopts supervised ML algorithms to detect DDoS attacks in MEC.

2.2.2 FOG vs MEC DDoS attack Mitigations

In paper [15], DDoS attacks are mitigated using a Naive Bayesian-based intrusion detection system (IDS), a Markov Model, and a Virtual Honeypot. To optimize the customer dataset, it is recommended to consolidate all modules into a single package, considering the large number of attributes involved. This consolidation facilitates the removal of duplicate and unnecessary features by choosing a subset that is relevant to DDoS detection. The authors carefully selected features that significantly contributed to improving detection accuracy and identifying DDoS attacks at an earlier stage.

To identify and assess DDoS attacks in real-world networks, the authors performed experiments involving different DDoS attack scenarios. Extensive network traffic data was collected using Wireshark as part of the evaluation process. As a result, Bayesian and Markov's models were used to reduce the likelihood of false positives. The authors in this paper [15] developed a model for determining key parameters from traffic requests for DDoS attack detection in fog networks. To overcome this limitation, our study introduces ML algorithms to detect DDoS attacks and employs a hybrid model to optimize the detection rates of these attacks.

According to this paper [13], security concerns limit the adoption and expansion of fog computing. They mention that the most frequent type of network attack among the numerous security worries is DDoS. DDoS attacks may cause a reduction in the resource usage of fog nodes. They demonstrated that the significance of mining DDoS intentions from incursions using association analysis cannot be overstated. Their research utilised hypergraph clustering to examine the connections between fog nodes that had experienced DDoS attacks. They utilized simulation to verify the effectiveness of their model's resource usage since DDoS attacks make system resources and servers inaccessible. They subsequently made the decision to combine their efforts since DDoS attacks may be exceptionally detrimental to a system's resources.

Their simulation results show that their methodology and the proposed solution surpasses alternative approaches in terms of efficiently utilizing fog node resources usage for an intrusion response strategy. Since they only looked at a few references for this paper [13], it is likely that their perspectives may not be universally shared. As a result, it has been recommended that future researchers undertake a comprehensive examination of DDoS defence within the domain of fog computing. Our work, however, addresses this need by conducting a detailed investigation in the context of MEC (rather than solely focusing on Fog computing). The issue that FOG computing has risen to security threats – IDS was proposed to resist the security threats such as DDoS attacks. They also proposed a hypergraph clustering model based on the Apriopi algorithm which is most efficient in DDoS attacks. Hence, our study utilizes the most effective mitigation scheme (hybrid model) for DDoS attacks.

In [21], theoretically, according to the authors, the integration of 5G and fog computing simplifies the deployment of security solutions for IoT networks. Devices can interact fast and effectively over 5G networks, whereas fog networks can only offer resources (storage and processing) for security tools like anomaly detection. A fog computing-based mitigation strategy is suggested to improve the detection and mitigation of DDoS attacks. Their architecture employs a database, and a classification technique called the k-NN algorithm to create a strategy for mitigating anomalies was developed. The database keeps track of the signatures of identified attacks, facilitating quicker detection the next time an attack is launched. They tested the framework's proposed k-NN classifier using the DDoS evaluation dataset (CIC-DDoS 2019 dataset). According to their research, the k-NN classifier can reliably identify DDoS attacks. To further evaluate their strategy, they intended to apply the framework to already-in-use fog computing systems in the future. Our work is focused on MEC security vulnerabilities (DDoS attacks) in 5G and their suggestions for further research.

2.2.3 SDN vs MEC DDoS attacks Mitigations

In one other paper [27], a hybrid approach combining statistical and ML techniques are described. The proposed approach combines statistical analysis and ML, integrated with SDN security measures, to effectively detect DDoS attacks. This hybrid method leverages the strengths of statistical analysis and ML algorithms, while also benefiting from the enhanced security capabilities provided by SDN. Each data set is analysed iteratively and compared to a dynamic threshold. ML is utilized to assess correlation measurements between the features once sixteen characteristics have been retrieved. With software-defined security (SDS), a dynamically configured SDN is used to offer a powerful policy framework to safeguard the availability, integrity, and privacy of all networks while allowing for fast reaction repair. ML is also currently employed to enhance the precision of detection.

Through this improvement, the accuracy rate is enhanced significantly, rising from 88.6% to 99.86%, while concurrently reducing the false positive rate (FPR). These results are derived from analysis conducted on experimental datasets; the results obtained outperformed existing techniques. Our study uses performance metrics to evaluate different ML techniques for mitigation purposes in terms of detecting the DDoS attacks. Whereas this paper [27] utilized a hybrid method for mitigating the DDoS attacks, they also configured SDN to ensure the quality of safeguarding the availability, integrity, and privacy as specified. Their paper [27] also showed that there is no conflict of interest with their findings, which implies that there are no apparent shortcomings.

The paper [29] examined an SDN-based detection system for DDoS attacks utilizing ML systems. In the first approach, algorithms with an accuracy of 98.3% were employed to detect attacks without considering the traffic type. Another proposed system categorised DDoS attacks into regular traffic and attack traffic, and KNN algorithms achieved 97.7%

sensitivity in this categorization, significantly reducing the workload on the controller. In their paper [29], they initially selected 12 features utilizing feature selection technique and trained classifiers on those selected subset of features.

The selection of features relied on the algorithm or threshold value employed. By modifying the threshold value, it was possible to choose different quantities of characteristics and train them to the classifier, resulting in different levels of accuracy. Overall, the models performed consistently well, surpassing an 80% accuracy threshold, indicating the effectiveness of the strategies employed for the dataset.

Consequently, this approach enabled the detection of malicious software, network browsing, and inter-layer attacks within SDN. To further safeguard and enhance the SDN infrastructure, the second approach involving NB could be implemented. Our study proposed ML methodologies along with performance measures to determine the most effective ML approach for mitigating DDoS attacks, thereby bridging the gap with the research outlined in paper [29].

According to this paper [31], SDNs have emerged as an innovative solution for enhancing computer networks by providing flexibility, The objective is to minimize operational expenses, provide protection against DDoS attacks, and detect both high-volume and low-volume attacks by combining statistical and ML techniques.

The detection technique consists of three primary components: the collector, entropy-based module, and classification section. Through evaluation and analysis, it is determined that the entropy-based module with a fixed threshold produces inadequate outcomes when tested with experimental datasets. On the other hand, employing a dynamic threshold leads to better outcomes, albeit with a higher false positive rate (FPR).

To address this issue, various classification algorithms are employed to obtain more accurate results. The dynamic threshold approach is considered superior to its counterparts due to its exceptional performance, exhibiting higher accuracy compared to similar methods. While the proposed model focuses on post-attack solutions, the exploration of DDoS attack prevention in SDN networks was deemed essential by the authors.

In the study cited [31], a solitary controller within an SDN framework was responsible for detecting DDoS attacks. However, future researchers are advised to involve multiple controllers to enhance the method's effectiveness across networks. Our study, and their study both propose solutions related to DDoS attacks. The gap is that their study proposed SDN as a mitigation scheme, while our study focuses on utilizing supervised ML algorithms for DDoS attack detection on MEC. Furthermore, their study integrates statistical methods with ML techniques, while our study leverages existing ML approaches to improve the detection rate within MEC.

In [33], the authors successfully demonstrated the effectiveness of SDN in mitigating DDoS attacks by offering a comprehensive view of all networks. They emphasize that relying on a few individual network entities, such as routers, is insufficient for effective prevention, as these entities possess limited knowledge about specific paths and neighbouring nodes.

To address these limitations and concerns, the authors proposed an alternative architecture that tackles the limitations mentioned above. They highlight that this model (SDN) is particularly suitable for certain environments, such as military networks, where centralized controllers can monitor and manage all network resources within a centralized infrastructure. To showcase the feasibility of their framework, they developed a prototype using the Dirichlet process mixture model.

The proposed algorithm not only outperforms a nonparametric mean shift (MS) clustering method in accurately identifying DDoS attack traffic, but also demonstrates effectiveness in identifying traffic flows for popular network applications like hypertext transmission protocol (HTTP) and file transfer protocol (FTP). As part of their future work, they plan to implement the proposed mitigation technique, conduct further performance evaluations on a larger scale, and deploy it in a real network environment.

In paper [36], SDN was defined by the authors as a networking paradigm that makes network devices programmable by redefining the term network. SDN allows network engineers to monitor and control the network efficiently, identify malicious traffic and link failures with ease and efficiency. In addition to the flexibility that SDN provides, the network is vulnerable to disruptive attacks like DDoS, capable of bringing down the entire system. They suggested the utilization of ML techniques to differentiate between legitimate network traffic and DDoS attack traffic, aiming to mitigate the impact of such attacks. Among the major contributions of their study is the identification of novel features that can be used to detect DDoS attacks. ML algorithms are trained using the created dataset, which is a CSV file containing feature data.

Previous researchers on DDoS attacks detection have either employed non-SDN or SDN. In these studies, a new hybrid ML model is utilized for the classification purposes. Specifically, for traffic classification, the hybrid SVC-RF (Support Vector Classifier-Random Forest) model demonstrated the highest detection accuracy of 98.8% while maintaining a low false alarm rate. The difference is that this paper [36] has implemented a hybrid SVC-RF model for detecting DDoS attacks on SDN, whereas our study will implement a hybrid model for detecting DDoS attacks on MEC.

According to this paper [37], in today's modern world, computer networks and systems play a critical role, and the integrity and confidentiality of these assets are of utmost importance. Among the various risks faced by these networks, DDoS attacks are particularly concerning. Detecting DDoS attacks is a complex task that must be accomplished before mitigation strategies can be put into place. ML/DL systems have been successfully used to detect DDoS attacks. However, these ML/DL frameworks have limitations, particularly in terms of optimal feature selection, which cannot be fully achieved.

In certain scenarios, ML/DL algorithms have shown limitations in accurately detecting DDoS attacks. The use of ML classifiers and traditional feature encoding methods has led to unexpected predictions in forecasting DDoS attacks. Additionally, previous attempts utilizing DNNs for feature extraction did not effectively capture the sequence information required for accurate detection.

To accurately predict DDoS attacks using benchmark data, this study introduces a novel approach called Hybrid Deep Learning (HDL) model, specifically the combination of Convolutional Neural Network (CNN) with Bidirectional Long/Short-Term Memory (BiLSTM). By leveraging the strengths of both CNN and BiLSTM, the proposed model aims to improve the accuracy and effectiveness of DDoS attack prediction. Based on ranking and selecting features from the data set, features were selected based on their high scores. With the CIC-DDoS2019 data set, the proposed CNN-BI-LSTM achieved an accuracy of 94.52 percent during training, testing, and validation. The difference is that this paper [37] investigated detection methods using DL algorithms on SDN whereas our study investigated detection methods based on supervised ML algorithms on MEC.

2.2.4 Mitigations of DDoS attacks in Different Network Environment

In another study [10], they discuss the creation, evaluation, and empirical validation of a new strategy for the efficient defence of multi-tenant 5G networks against DDoS attacks. The proposed approach guarantees simultaneous security of infrastructure providers, end users, and the network in a 5G network. The proposed approach offers a notable benefit to mobile edge security as it provides security for virtually any segment of the 5G network. During the evaluation of the proposed solution, a realistic scenario was considered, simulating a use case where more than 256 attackers simultaneously launched a flood of malicious traffic at a rate of 100 Mb/s, specifically utilizing the user datagram protocol (UDP), and targeting the 5G network.

The proposed solution builds upon an extension of Snort, which can be further adapted to ensure compatibility with IDS that produces events using a standardized format. The technique has proven to be scalable, exhibiting essentially constant behaviour even under the most challenging conditions for attackers or attack types. Their research focused on DDoS attacks and mitigation strategies for dealing with them effectively in terms of safeguarding 5G networks. This paper [10] under consideration did not discuss the utilization of a framework in a mitigation plan system for preventing attacks on the correct site, and for that reason our study delves deeper into this aspect. This technique for mitigation and detection has the added benefit of closing the cognitive management loop envisaged for 5G networks in the future. The cognitive management loop that is intended to be completed in future 5G networks has been added as a benefit of this detection and mitigation combo.

According to another paper [19], in spite of internet developments, DDoS threats have become more advanced. They demonstrated a feedback mechanism built on the autonomous systems (AS) edge router, taking advantage of the ease of control offered by AS. The technique mainly makes use of the study of concrete items through mathematics. The modelling of their findings indicates that the system maintains a high survival rate for lawful traffic. They demonstrated that the approach may also be enhanced to achieve a better survival rate, which they intend to investigate in further detail in subsequent research. Based on this paper [19], it is recommended that future research focus on the analysis of security measures that facilitate the detection of DDoS attacks and reduce DDoS damage, which is, indeed, the primary emphasis of our research.

Paper [24] addresses DDoS attacks at the network edge. The researchers propose a concept where the attack volume is distributed among edge servers in close proximity to the targeted server. They classify edge DDoS mitigation (EDM) as an NP-hard problem and present optimal solutions for small EDM problems using integer programming and suboptimal solutions for large EDM problems using game theory. Theoretical and experimental evidence supports the effectiveness and efficacy of their methods in handling EDM. Our research, along with theirs, aims to mitigate DDoS attacks at the network interface.

In paper [26], topology is used for actuation, allowing actions to be performed nearby the attack source or destination while abstracting the underlying infrastructure. A cognitive layer determines the appropriate mitigations based on the network topology. As a result of the tests conducted, the findings verified that the self-managed loop can effectively manage the bandwidth, handle the complexity of attack packets and network topology, all within a timeframe of about one second. This holds true even when the attack is launched concurrently by 256 devices transmitting harmful data at a speed of 100 Mbps. The proposed architecture in their study enables actuation based on topology awareness, abstracting the actual infrastructure topology. The cognitive layer decides on the appropriate mitigation based on network topology.

In a similar study, it was found that the proposed method can protect against future 5G network attacks. There is no restriction on how new modules can be added to the design suggested in this paper [26]. Additionally, their findings have the potential to be applied to novel use cases to enhance network performance. This includes the ability to incorporate new sensors, actuators, rules, and policies based on these emerging use cases. Our study specifically proposed techniques for detecting DDoS attacks on MEC, enabling 5G networks to be vigilant against potential attacks. In contrast, the paper referenced as [26] addressed the safeguarding and mitigation of 5G networks against DDoS attacks specifically targeting edge computing environments. Our research focused on application layer DDoS attacks (DNS flooding), whereas their work focused solely on UDP.

In a related study [7], researchers demonstrated the growing significance of application-level DDoS attacks via compromised IoT devices. These attacks pose a major concern as they generate traffic that appears legitimate at the application level. Consequently, conventional solutions prove ineffective in mitigating such attacks, and countering them becomes exceptionally challenging due to the enormous volume and dispersed nature of the generated traffic. ShadowNet was developed to mitigate application-level IoT-DDoS attacks by exploiting emergent edge technology. That has sped up the discovery and arrest of such attacks, reducing the damage they do. It not only protects web services by identifying IoT-DDoS 10 times quicker than current techniques, but it also blocks 82% of traffic from entering the Internet backbone, thereby significantly decreasing damage. With a prototype implementation, they offer a positive early evaluation.

Future researchers are recommended to further the study of ShadowNet implementation and assessments to explore the trade-offs in real-world scenarios. This paper [7], has investigated IoT-DDoS attacks, and ShadowNet as their mitigation scheme, but our study concentrates on DDoS attacks at the network edge, and we propose different ML algorithms to mitigate the attacks. The difference is that the authors of this paper [7], advised future researchers to continue with their proposed scheme (ShadowNet), whereas in our study, we compared the analysis of different ML algorithms in terms of mitigating the DDoS attacks, and we also optimize those proposed mitigation schemes (ML techniques) to ensure the efficiency of mitigating DDoS attacks vulnerable to multi-access edge computing.

In this paper [28], the researchers used a semi-supervised ML strategy to categorize DDoS attacks in their research. It starts with unlabelled traffic information collected against three victim-end defensive mechanisms, such as the web server. Traffic rate, processing latency, and CPU use are some of the aspects that were considered. The unlabelled data is grouped using two distinct clustering techniques, and the final classification of traffic flows is decided using a voting process. As the data was labelled, the researchers added an additional class called 'Suspicious' to instances falling into opposite clusters. Using the provided labelled data, SL algorithms such as KNN, Support Vector Machine (SVM), and RF were utilized to classify DDoS attacks. Through optimized parameter tuning, they achieved accuracy scores of 95% for KNN, 92% for SVM, and 96.66% for RF models. The accuracy of label assignments was further validated by testing their schemes on a subset of the benchmark CICIDS2017 dataset, as well as novel attack vectors.

They obtained greater than 82% accuracy of label assignments. For their future recommendations, they proposed improving voting methods for labelled data and using ML algorithms in clustering and classification, which is what our study explores. Having explained their findings, the gap is that our study utilized the supervised ML approach, and thereafter the best algorithms are optimized for the purpose of getting certainty in terms of efficient mitigation of DDoS attacks.

For identifying DDoS attacks, [30] applied an organized flow of feature engineering and ML. With the help of engineering features, they obtained datasets with significant features of different dimensions, using backward elimination, chi2, and information gain scores. To demonstrate the adaptability of datasets for ML under optimal tuning of parameters within a given value, several supervised ML models are applied to feature-engineered datasets. It has been demonstrated that substantial feature reductions can be affected to make DDoS detection optimal and more efficient with a minimum impact on performance. An experimentation flow that is clearly defined is proposed as part of a strategic-level framework that integrates feature engineering and ML.

In addition, cross-validation and areas-under-curve analysis is performed to validate the models. Their study concurs that data overfitting and collinearity problems can be prevented, and DDoS attacks can be detected using its comprehensive solutions. The K-Nearest Neighbours algorithm generally performs the best in the study of DDoS datasets, followed by SVM. RF performs better with low-dimensional datasets with discrete features than high-dimensional datasets with numerical features.

A significant reduction in processing overhead is achieved with all ML models when datasets with the fewest features are used. Experimental results show that approximately 68% of feature space can be reduced while accuracy is only affected by 0.035%. However, the difference is that our study is only using a supervised ML algorithm, whereas their paper [30] utilized engineering features and ML techniques.

In a related study [8], authors proposed a new Hidden Markov Model (HMM)-based anomaly detection method that utilizes the DDoS source IP monitoring mechanism and the concept of distributed detection, and a framework is developed to detect DDoS attacks. The authors introduce a distributed reinforcement learning (DRL) approach that aims to enhance both the accuracy of detection and minimize communication costs among multiple detection agents. Through experiments, it is demonstrated that the HMM-based detection model achieves exceptional accuracy while completely avoiding false alarms.

Regarding the application of HMMs in distributed detection, the proposed DRL is shown to be very promising for optimizing detection accuracy. According to the authors in this paper [8], there is still a substantial amount of work that remains to be accomplished in the near future. Future tasks include assessing the effectiveness of the HMM-based approach in real-time DDoS detection scenarios and implementing new DRL algorithms to achieve improved trade-offs between detection accuracy and communication load. The difference is that our study proposes ML algorithms as the mitigation scheme for DDoS attacks, and thereafter optimizes the best ML algorithms using hyperparameter optimization.

In a related study [9], due to the sophistication of DDoS attack methods and the ease of finding related tools over the internet, detection and mitigation have become very challenging. Anomaly detection techniques such as ML are accurate and practical ways of distinguishing legitimate traffic from DDoS traffic. They used Real-life intrusion detection datasets to train and test ML-based classifiers. They ultimately compared five ML algorithms based on what was described above.

Based on their findings, they also discussed the research concerns with regard to DDoS attacks. Their findings establish the following: (1) the Random Forest classifier is a superior choice for DDoS detection, and (2) the accuracy gained in the trials is above 96 percent on a real-world dataset. The difference is that our paper optimizes the ML algorithms to detect DDoS attacks, and thereafter designs a hybrid scheme using those evaluated best ML algorithms for the purpose of generating certainty in detecting DDoS attacks.

In [32], deep learning approach was developed, aiming to achieve higher accuracy in detecting and classifying DDoS attacks within network traffic. In this investigation, a Deep Neural Network (DNN) model was selected as it outperforms shallow ML methods by combining feature extraction and classification operations within its architecture. The comprehensive review of existing literature supports the proposition that the suggested model is the optimal choice for studies utilizing deep learning in DDoS attack detection, as it has demonstrated significant success across various DDoS datasets.

In [34], a hybrid deep learning model, combining convolutional neural networks (CNNs) and long short-term memory (LSTMs) cells, was utilized to detect DNS flooding attacks. The researchers employed the CICIDS dataset, which contains real-world data related to DNS flood attacks, for their experiments. The dataset was used to train and test the hybrid model, providing a comprehensive basis for its performance evaluation.

Remarkably, the proposed model achieved an accuracy rate of 99.87 percent in categorizing test data, without the need for feature selection techniques, coding, or normalization. Out of a total of 51,652 data points, only 40 false positives were identified. This indicates a significantly improved performance in terms of time efficiency for data preparation. The researchers conducted their analysis on the entire dataset, allowing for meaningful comparisons with results obtained from other classification methods.

Overall, the integration of CNNs and LSTMs in the hybrid deep learning model proved to be highly effective in detecting DNS flooding attacks. The model exhibited exceptional accuracy and a minimal false positive rate, demonstrating its superiority over alternative classifiers.

To acquire the best classification results, they tested with several different CNN and LSTM models. They also supplied hyper parameters that helped enhance classification results. The suggested approach for accuracy has less false positives than previous methods for identifying anomalous traffic, despite the lack of a pre-processing phase. As a result, DNS flooding has become less susceptible.

The paper [34] presents a system approach for detecting DNS flooding from DDoS attack patterns. The attacker's conduct is shown by these patterns. When the patterns of subcategories of each attack are recognized, the perpetrators' nefarious objectives become more apparent. DNS flooding attacks might be detected with a low false-positive rate in the future, making service providers more precise. Consequently, service providers can better serve their customers.

In [35], the researchers provided a definition of DDoS attacks, as malicious activities aimed at disrupting the regular flow of traffic to a specific server or network by overwhelming it with a significant volume of internet traffic. These attacks have been recognized as a significant threat to the overall security of network environments. In order to improve the identification and detection of DDoS attacks, the authors proposed a framework called PCA-RNN (Principal Component Analysis-Recurrent Neural Network).

To simplify the detection process, most of the network characteristics are selected to represent traffic. Then they used the PCA algorithm to reduce the time complexity. According to their findings, PCA can significantly reduce prediction time while retaining most of the original information. The evaluation results showed that PCA-RNN achieves significant performance compared to numerous existing DDoS attack detection methods in terms of accuracy, sensitivity, precision, and F-score.

In [38], DNS, being a fundamental and crucial service on the internet, holds immense significance in terms of security and reliability. DDoS attacks pose a persistent threat to the reliable functioning of DNS systems, including top-level domain (TLD) servers. In response to this challenge, the authors introduced an innovative approach to mitigate

DDoS traffic on TLD servers. This method involves implementing a traffic filter that leverages machine learning algorithms on prominent recursive DNS servers across the Internet. Based on spark, the classification model performs with a 0.0% false positive rate (FPR) and 4.36% false negative rate (FNR), so both accuracy and performance requirements are met. In future work, they recommended features to be extracted and the model will be applied in a streaming manner suitable for real-time firewall rules. This traffic filtering model will also be used to study real-time detection and prevention.

The DNN model, employed as a deep learning approach, demonstrates nearly 100% accuracy in detecting DDoS attacks on the CICDDoS2019 dataset. It has also exhibited a high level of effectiveness with approximately 95% accuracy in classifying DDoS attacks. The combination of the DNN model with the CICDDoS2019 dataset provides valuable insights for other researchers in the field of DDoS intrusion detection. These findings suggest that incorporating the DNN model into IDS and security layers for software-based network datasets would be beneficial due to its exceptional accuracy in network analysis.

Their study primarily concentrated on the detection of DDoS attacks using a feed-forward-based DNN model; whereas our study concentrates on detecting DDoS attacks using supervised ML algorithms and design a new hybrid model to detect DDoS attacks. Our study makes use of supervised ML instead of DNN, because SL have the ability to produce the data outputs from previous experience than DNN.

In another paper [16], the researchers proposed the edge coordination-based traffic scheduling (ECTS) algorithm for scheduling traffic in a time and wavelength division multiplexed passive optical network (TWDM-PON) during DDoS attacks. TWDM-PON was designed to address the needs of edge computing optical networks (EC-ONU), aiming to mitigate the impact of DDoS attacks on time-sensitive services and maintain a high quality of service (QoS). According to the researchers, ECTS demonstrates increasing performance advantages as the number of targeted nodes rises. In the case of eight attacks, ECTS reduces the impact on time-sensitive services by 7.92% through the cooperation of EC nodes. While their research focuses on addressing DDoS attacks in TWDM-PON, our study concentrates on utilizing ML algorithms for detecting DDoS attacks on MEC. To evaluate and compare the effectiveness of each ML algorithm in detecting DDoS attacks on MEC, various performance metrics were utilized. Although this paper [16] also focuses on edge computing, the main difference from ours is that their specified proposed technique is different from the one we proposed. Specifically, they used TWDM-PON as their mitigation scheme for DDoS attacks, whereas our study uses supervised ML algorithms – as well as a hybrid model.

Authors in [18], examined attack models in MEC systems with an emphasis on caching and mobility offloading methods. The results of this study show that MEC systems are susceptible to a wide range of attacks, including DoS and rogue attacks. This document offers proposed security measures [18]. A particular method, such as Reinforcement Learning (RL), is available for ensuring data privacy. They examined potential problems and assessed the effectiveness of the RL-based security approach for MEC. They discussed the study of several security issues in their work, contributing significantly to an understanding of the complexities involved in compromised security.

This paper [18] examines various security challenges and proposes solutions, specifically addressing secure mobile offloading to counter jamming attacks and intelligent attacks. Additionally, they discussed edge security solutions, presenting fixed strategies based on specific network configurations or attack models. Simulation results demonstrated the effectiveness of the reinforcement learning (RL)-based approach in defending the MEC system against a range of threats with minimal overhead. Their main emphasis was on examining threat models commonly found in MEC, including jamming, DoS, spoofing attacks, smart assaults, MITM, and privacy breaches. Differently, our work specifically targets an area that is not covered in their study, which is DDoS attacks. Although their paper [18] did not outline the shortcomings of their findings, our study focuses on DDoS (DNS) attacks, and the implementation of a hybrid model by integrating supervised ML.

According to another paper [20], in the development of 5G networks, multi-access edge computing (MAEC) is a key enabler. A promising DDoS mitigation architecture is made possible by MAEC systems, which allows distributed computation at the network's edge. MAECX is a hybrid solution that protects targets from DDoS attacks directly at the source. It was also highlighted in the paper [20] that MAEC computation allows for the localisation of sophisticated DDoS prevention algorithms for the efficient management of enemy traffic. Their results are consistent with the paper's [20] proposal to evaluate how well the MEC performs in terms of security, particularly when dealing with DDoS attacks.

In paper [25], the authors proposed a DDoS mitigation architecture based on Multi-Access Edge Computing (MEC). In this architecture, intelligent filters are implemented both at the attack source and edge destinations to safeguard the network against malicious traffic originating from a wide range of Internet of IoT devices. Their experiments demonstrated that self-organizing map (SOM) filters effectively to detect local traffic. The distributed design and control strategy of the MEC shield reduce CPU utilization by approximately 10% compared to other solutions during DDoS attacks.

Authors in [23], conducted a study that specifically examined edge computing, DDoS attacks, and job offloading within edge computing. The findings from their study provided valuable insights that aided in the planning and execution of our own study. This survey suffers from several fundamental flaws, including the inability to conduct experiments. In addition, the study's limitations include its inability to investigate the practical consequences of DDoS attacks on edge servers. Furthermore, multiple facets of edge computing remain unexplored and require further exploration. As part of the edge computing paradigm, limited-capacity edge servers are used to support advanced calculations that are sensitive to latency. Due to this, it is crucial that the servers are available when task requests are received.

It is important to note that these servers are vulnerable to security threats, just like other internet technologies. Additionally, edge servers are less capable of processing such attacks, so their consequences are more severe. The paper [23] extensively covers various facets of DDoS attacks, including attackers, handlers, zombie hosts, and target hosts. It delves into the impact of these attacks on edge servers, specifically focusing on the agent-handler, IRC-based, and web-based models. The study also explores the broader implications of DDoS attacks on edge computing.

2.3 CHAPTER SUMMARY

This chapter focused on a literature assessment of previous work in network access techniques. It also assessed how techniques have been improved over time. A thorough examination of the needs for a 5G network, including security concerns, low latency, widespread device connection, data management and synchronization, bandwidth and data transfer, heterogeneity has been made in order to contextualise the current study.

MEC has been seen as a solution to a growing need for mobile network security and difficulties such as extensive device connection, data management and synchronization, capacity, and data transmission, and latency could be resolved. According to the literature surveyed, MEC can handle more massive connectivity in 5G.

A limited number of studies in literature have focused more on mitigating the security attacks in FOG, MCC, SDN, TWDM-PON and other network technologies. Hence, the aim of our study addresses security issues (DDoS attacks) in MEC. Furthermore, most studies have not focused on hybrid techniques in countering the DDoS attacks, whereas our study implements a hybrid-model to mitigate DDoS attacks in MEC.

Although there are a number of mitigation schemes for DDoS attacks detection, designing a real-time detection of DDoS attacks is still one of the major concerns. This study focusses on real time detection of DDoS attacks (which are ML algorithms), and thereafter implements an optimal mitigation technique (hybrid-model) of addressing DDoS attacks.

On the whole, most papers have focused on different types MEC security risks in general, without devoting specific focus on DDoS attack in MEC. Examining what other researchers have done, such as network layer and application layer DDoS attacks are the most studied. MEC performance in the presence of DDoS attacks was examined by comparing the best performing mitigation schemes for DDoS attacks and this is the specific gap that our study fills.

CHAPTER 3: METHODOLOGY

3.1 INTRODUCTION

This chapter describes the methods and proposed schemes (LR, NB, DT, and RF) used to implement the hybrid model. We discuss the utilized simulation tools, the parameters used in the simulation, and the performance metrics applied to evaluate our proposed DDoS mitigation schemes. The dataset used is based on reflection attack, therefore the specific type of reflection attack (DNS) is discussed in this chapter. The collected real-time DNS dataset that was used to train, and test the proposed schemes, and to implement the hybrid model as the efficient mitigation scheme for DNS amplification attacks are examined.

3.2 DDoS attacks

DDoS disrupts normal traffic on a server, network, or service by flooding it with Internet traffic. DDoS attacks usually occur at two open systems interconnection reference models – OSI model (which are network, and application layer). Several types of DDoS attacks are categorized in the network layer, and application layer of DDoS attacks. The network layer is the third level layer of OSI model, and it provides data routing paths for network communications, whereas the application layer is the seventh layer of OSI model which provides web application services, and it enables applications on different computers and networks to communicate effectively.

Network layer DDoS attacks are user datagram protocol flooding attack (UDP flooding attack), Internet control message protocol flooding attack (ICMP flooding attacks), and transfer control protocol which synchronizes flooding attacks (TCP SYN flooding attack). Under application layer DDoS attacks, we have Extensible Markup Language flooding attacks (XML flooding attacks), Domain name server flooding attack (DNS flooding attacks), Simple network management protocol (SNMP flooding attack), and Hypertext transfer protocol flooding attacks (HTTP flooding attacks). From this array of attacks, our study specifically focused on the application layer DNS flooding attacks.

The protocols available at the network layer can be used for carrying out a DDoS attack at the network layer. Firewalls may be burdened by DDoS attacks and their potential impact is compromised if available bandwidth is affected. The most common types of these attacks are TCP SYN flooding, UDP flooding, and ICMP flooding.

MEC has experienced a growing number of DDoS attacks on its application layer throughout the years [3]. Productivity and service quality are negatively affected by these attacks. In application layer attacks targeting edge network services, flood packets containing extensive HTTP floods are often dispatched at high speeds. This poses a potential risk of service depletion caused by these DDoS attacks.

3.2.1 TCP SYN Flooding attacks

In order for a client to establish a connection with a server, the server needs to bind to a specific port and listen for incoming connections. This initial step is referred to as a passive open. Once the passive open is established, the client can initiate an active open. The process of establishing a connection involves a three-way handshake, as depicted in the figure below.

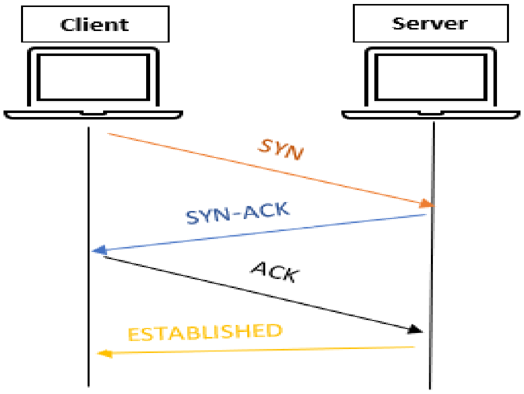


Figure 3.1 TCP Session Diagram

TCP, which is a connection-oriented protocol within the TCP/IP model stack, operates at the transport layer. To enable communication between hosts, a three-way handshake is necessary. The process begins with the initiating host sending an SYN message, followed by the remote server responding with an SYN + ACK message to acknowledge the request. The initiating host then completes the handshake by sending an acknowledgement. Once both hosts have received acknowledgements from each other, the connection is established.

However, attackers have exploited the use of half-open connections, causing the kernel to run out of memory due to excessive transmission block allocation [39]. Coordinated attacks on vulnerable internet nodes can also be carried out by fictitious IP addresses that are frequently employed in TCP SYN flooding DDoS attacks. Hosts that respond with an RST flag or do not respond at all are considered fake hosts. Due to this gap, the handshake procedure remains incomplete [40].

3.2.2 UDP Flooding attacks

The UDP protocol is used when packet transmission reliability is not a critical requirement. Real-time applications such as voice and video transmission, as well as internet games and instant messaging, commonly utilize UDP [40]. However, UDP is susceptible to certain vulnerabilities, including protocol 16 vulnerability, which can be exploited to launch DDoS attacks, particularly flooding attacks. In such attacks, the target's cloud ports are flooded with UDP packets (specifically port 13), causing an overwhelming volume of traffic. Exploiting UDP's connectionless and unreliable nature, the attack floods the target system with malicious traffic, resulting in congestion of the response queue and preventing legitimate users from receiving responses [39]. Due to UDP's unstable characteristics, the target system is unable to effectively limit the transmission rate of the attackers [39].

3.2.3 ICMP Flooding attacks

The ICMP protocol, which operates at the IP layer, is commonly used to check the connectivity status of a host's network. However, this protocol has been exploited in DDoS attacks, specifically smurf and ping flood attacks. These attacks aim to exhaust bandwidth and overwhelm a targeted device by sending large volumes of ICMP messages. The excessive ICMP traffic floods the target, making it unable to respond to legitimate requests from various sources. The intention of these attacks is to disrupt the target's network connectivity and potentially cause the device to crash.

3.2.4 HTTP Flooding attacks

HTTP floods, also referred to as HTTP DoS attacks, are tactics employed to inundate web servers and applications hosted in the cloud by utilizing malicious HTTP packets. Unlike other attacks, HTTP floods do not necessarily require a massive volume of traffic. For instance, an HTTP GET attack can incapacitate a target by flooding it with numerous request sessions, resulting in the infection of numerous internet nodes. The primary objective is to disable the target server or application by overwhelming it with a high number of requests.

3.2.5 XML Flooding attacks

This attack usually occurs when the attackers intercept XML data as it is being sent and adds malicious code to it. Private information may be disclosed when the application is processed. This type of attack allows the attacker to view the file system and, sometimes, interact with the back-end services that the application can access. Legitimate users are then denied access to web services by XML flooding attacks. Such attacks are carried out by sending a large number of XML-based requests and letting the server parse them individually. Due to the ease of implementation, an Extensible Markup Language (XML) DoS attack can be carried out with less sophisticated tools [3]. These attacks often result in XML Denial of Service (DoS) attacks.

3.2.6 SNMP Flooding attacks

A simple network management protocol (SNMP) is used for configuring and collecting information from network devices such as servers, hubs, switches, routers, and printers. SNMP flooding attacks take place when the perpetrator (or attacker) sends out a large number of SNMP queries with a forged IP address (the victim's) to numerous connected devices. As increased devices reply, the attack volume grows until SNMP responses overwhelm the target network, bringing it down. This DDoS attack uses the SNMP to generate attack volumes of up to thousands of gigabits per second to clog up the target's network.

3.2.7 DNS Flooding attacks

DNS flooding attacks is also known as amplification attack. DNS floods are DDoS attacks intended to overwhelm a target DNS server. DNS floods are DDoS in which attackers flood DNS servers for a specific domain to disrupt DNS resolution. A DNS flood attack compromises the ability to respond to legitimate traffic to a website, API, or web application by disrupting DNS resolution. Flood attacks are difficult to distinguish from regular heavy traffic because they often use a plethora of unique addresses to query for real records in the domains, mimicking legitimate traffic. A DNS flood attack should be clearly distinguished from a DNS amplification attack. A DNS amplification attack occurs when an attacker sends out a small DNS query with a spoofed target IP, causing the spoofed target to receive much larger DNS responses.

The following diagram shows how DNS flood attacks work**.**

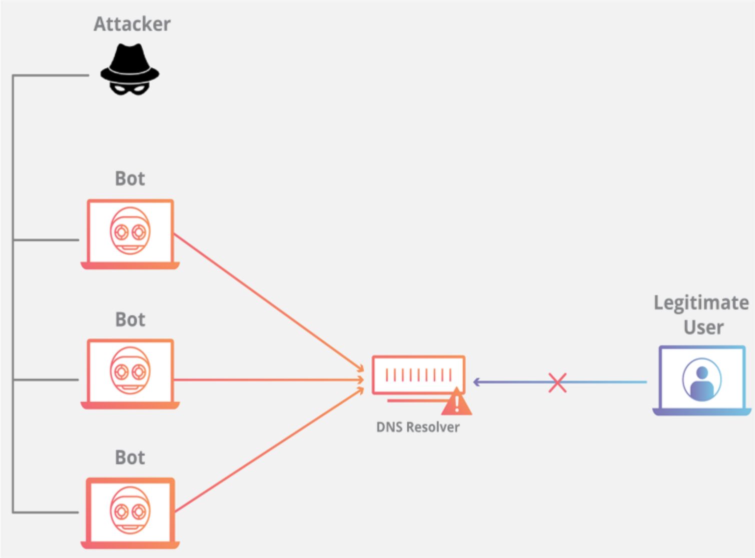


Figure 3.2 DNS flood attack work

With DNS attacks, scripts running on compromised botnet machines generate a flood of UDP requests to exhaust server-side resources such as memory, and CPU. DNS flood attacks are a variant of UDP flood attacks, since DNS servers use UDP to resolve names, and are application layer attacks. DNS flood attacks are conducted by running scripts from multiple servers to attack the DNS server. In these scripts, malformed packets are sent from spoofed IP addresses. An application layer attack such as DNS flood requires no response, so the attacker can send packets that are neither accurate nor formatted correctly. Attackers can spoof all packet information, including IP address, to appear as if they are coming from multiple sources. Randomized packet data also makes it easier for offenders to get by conventional DDoS defences, making IP filtering techniques such as utilizing Linux IPtables worthless.

3.3 DDoS (DrDoS_DNS) Dataset

DDoS attack is a threat to network security that exhausts target networks with malicious traffic. Even though many statistical methods have been implemented to detect DDoS attacks, designing a real-time detector with low computational overhead remains a challenge. Alternatively, the evaluation of new detection algorithms and techniques relies on well-designed datasets.

We used a real-live time generated existing dataset [5]. We proposed ML algorithms and trained them with the dataset in order to detect DNS flooding attacks. We then implemented a hybrid model to effectively detect DNS flooding attacks using ML algorithms (LR, DT, NB, and RF) on the datasets.

There have been several studies that propose taxonomies for DDoS attacks. Despite all the commendable taxonomies proposed so far, the mitigation schemes have remained quite limited. It is therefore necessary to implement a new mitigation scheme to detect DDoS attacks. The information from the collected datasets shows that the most up-to-date common DDoS attacks, which resemble real-world data (PCAPs), have been generated. Our study focuses therefore on DNS flooding attacks.

Exploitation-based attacks and reflection-based attacks can be classified as DDoS attacks. A reflection-based DDoS attack hides the attacker's identity by using third-party components authorised by the attacker. These assaults can be carried out through the use of application layer protocols using transport layer protocols like User datagram protocol (UDP) and Transmission control protocol (TCP), separately or in combination. TFTP, CharGen and NTP are examples of UDP-based attacks, while MSSQL, SSDP are examples of TCP-based attacks as shown in Figure 3.3 below. There are a variety of attacks that can be conducted through TCP or UDP, including those against DNS, LDAP, NETBIOS, and SNMP. This suggests that the primary subject of our investigation shall be DDoS attacks based on reflection.

Exploitation-based attacks, also referred to as attacks using legitimate third-party components, employ various tactics to conceal the attacker's identity. These attacks involve sending packets to reflector servers, where the attacker manipulates the source IP address to match the target IP address of the victim. The intention is to overwhelm the victim with an influx of response packets. These attacks can utilize application layer protocols that rely on transport layer protocols like TCP and UDP. In the case of TCP, an exploitation attack called SYN flood is employed, while UDP flood attack is used for UDP-based attacks. By sending a large volume of UDP packets to the destination host, a UDP flood attack is initiated.

By transmitting UDP packets at an extremely rapid pace to arbitrary ports on the targeted devices, the network's capacity is depleted, leading to system crashes and a decline in overall performance. Furthermore, SYN floods exploit the three-way handshake of TCP to consume server resources. An SYN attack is initiated by sending repeated SYN packets to a target machine until that machine crashes or malfunctions. A UDP-Lag attack disrupts the connection between the client and server. Generally, this attack is used when players want to slow down/interrupt other players to outmanoeuvre them in online games. This attack can be carried out in two ways: using a hardware switch known as a lag switch or by a software program that runs on the network and hogs the bandwidth of other users. The categories of exploitation-based attack and reflection-based attack is displayed in the

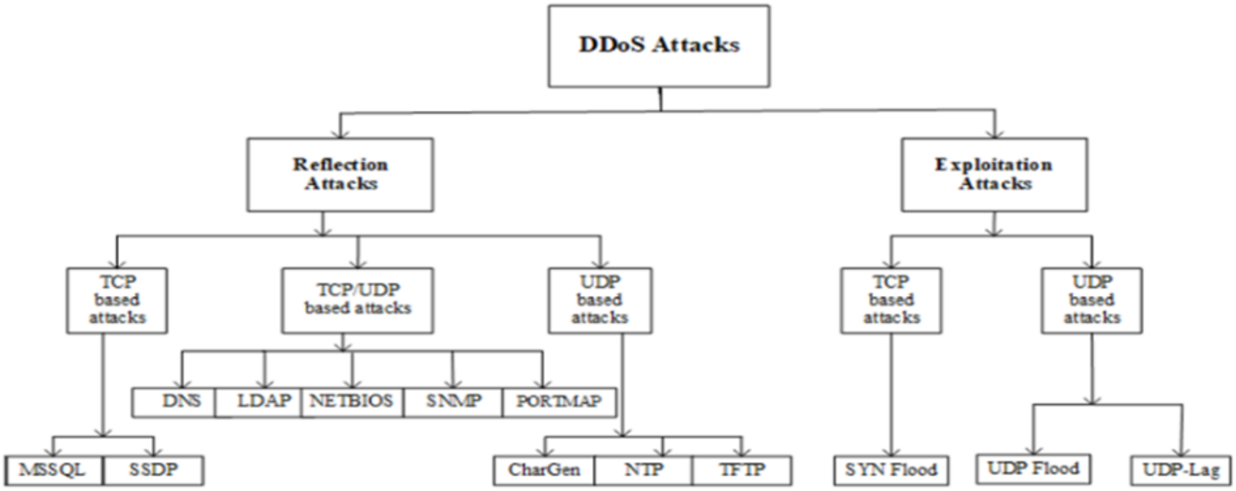


Figure 3.3 Network & Application DDoS *[6]*

3.4 SIMULATION TOOL

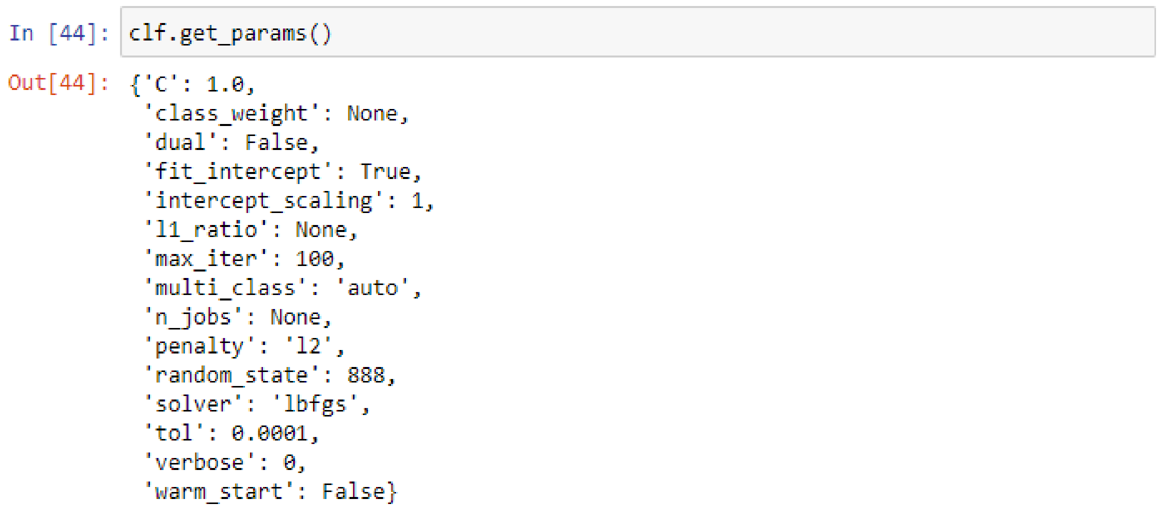
DrDoS_DNS is the name of the csv file that contained the dataset used to train our ML models and implement hybrid models, and it focused on DNS flooding attacks which is a type of a DDoS attack. In simulating the models, the DDoS (DrDoS_DNS) evaluation dataset [5] was collected. Performance metrics were utilized to evaluate different ML's algorithms. The findings were derived using ANACONDA3-2021 and Jupyter notebook. The reason we selected Jupyter notebook is that it is an open-source simulator of ML algorithms. For charts, we used Microsoft excel, and Jupyter notebook along with Python programming language. For implementing a hybrid model, we integrated ML algorithms.

In order to assess the efficiency of hybrid models and ML algorithms, we used the method of computing performance measures (accuracy, precision, F1-measure, recall, Matthew's correlation coefficient (MCC), Area Under Receiving Operating Characteristic (AUROC)). The first focus of the study is to evaluate the five ML (LR, DT, NB, RF, and KNN) algorithms using the collected DDoS attacks dataset [6]. The second focus of the study evaluates four ML algorithms (LR, DT, NB, and RF), and uses the evaluated algorithms (RF, LG, DT, & NB) to implement the hybrid model.

3.4.1 PARAMETER I

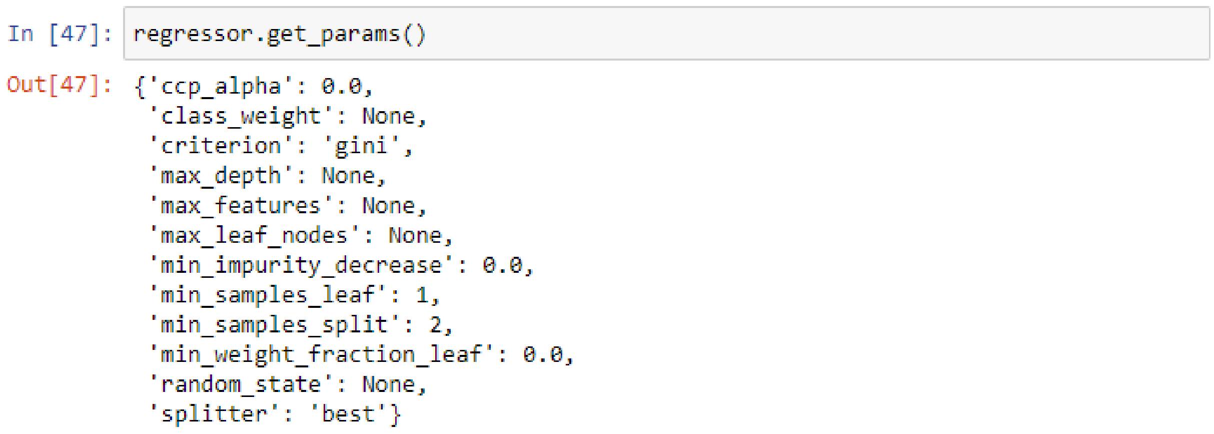
The parameters for the first focus of the study are shown below .

LR Parameters



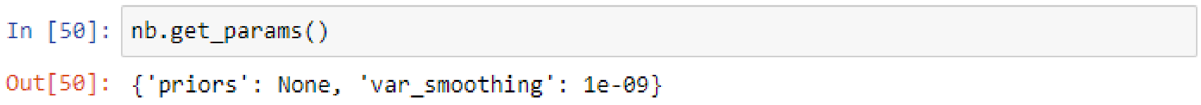
```
In [44]: clf.get_params()

Out[44]: {'C': 1.0,
          'class_weight': None,
          'dual': False,
          'fit_intercept': True,
          'intercept_scaling': 1,
          'l1_ratio': None,
          'max_iter': 100,
          'multi_class': 'auto',
          'n_jobs': None,
          'penalty': 'l2',
          'random_state': 888,
          'solver': 'lbfgs',
          'tol': 0.0001,
          'verbose': 0,
          'warm_start': False}
```

DT Parameters

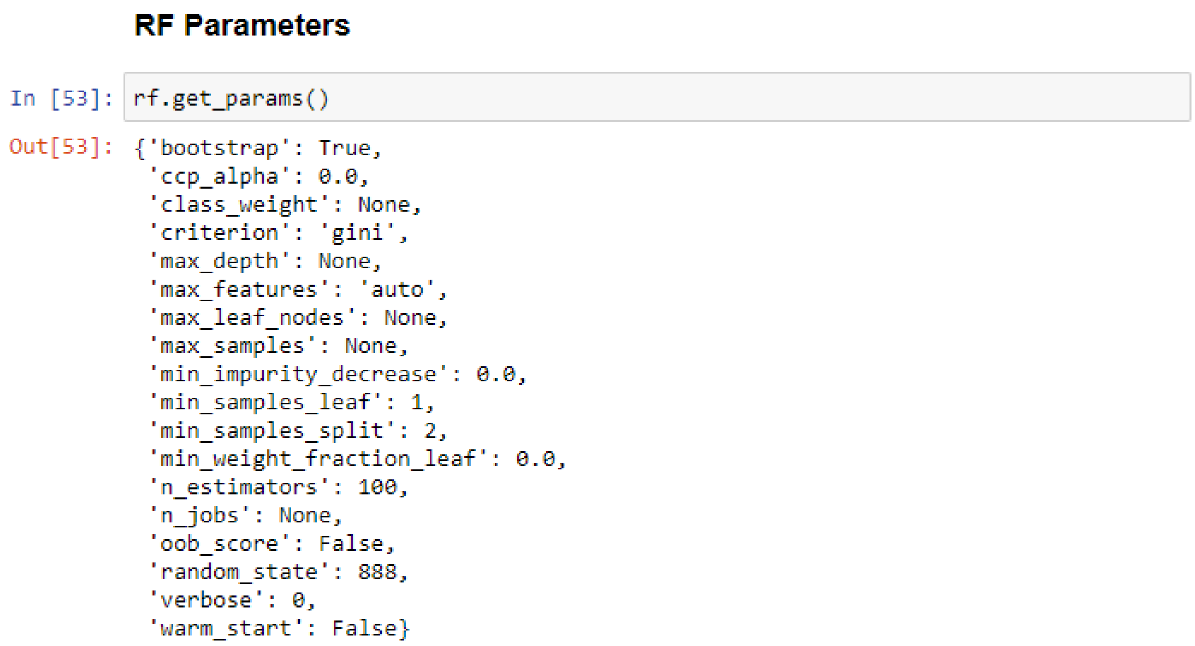


```
In [47]: regressor.get_params()

Out[47]: {'ccp_alpha': 0.0,
          'class_weight': None,
          'criterion': 'gini',
          'max_depth': None,
          'max_features': None,
          'max_leaf_nodes': None,
          'min_impurity_decrease': 0.0,
          'min_samples_leaf': 1,
          'min_samples_split': 2,
          'min_weight_fraction_leaf': 0.0,
          'random_state': None,
          'splitter': 'best'}
```

NB Parameters

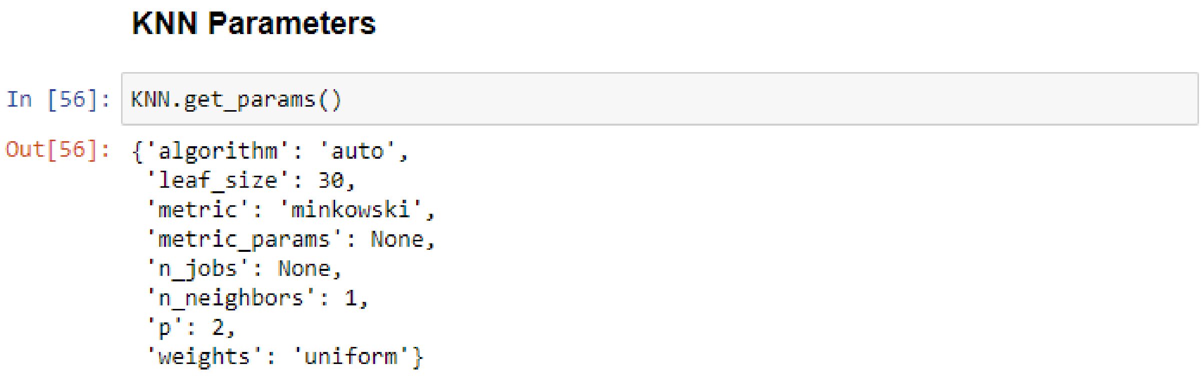


```
In [50]: nb.get_params()

Out[50]: {'priors': None, 'var_smoothing': 1e-09}
```



RF Parameters

```
In [53]: rf.get_params()

Out[53]: {'bootstrap': True,
          'ccp_alpha': 0.0,
          'class_weight': None,
          'criterion': 'gini',
          'max_depth': None,
          'max_features': 'auto',
          'max_leaf_nodes': None,
          'max_samples': None,
          'min_impurity_decrease': 0.0,
          'min_samples_leaf': 1,
          'min_samples_split': 2,
          'min_weight_fraction_leaf': 0.0,
          'n_estimators': 100,
          'n_jobs': None,
          'oob_score': False,
          'random_state': 888,
          'verbose': 0,
          'warm_start': False}
```



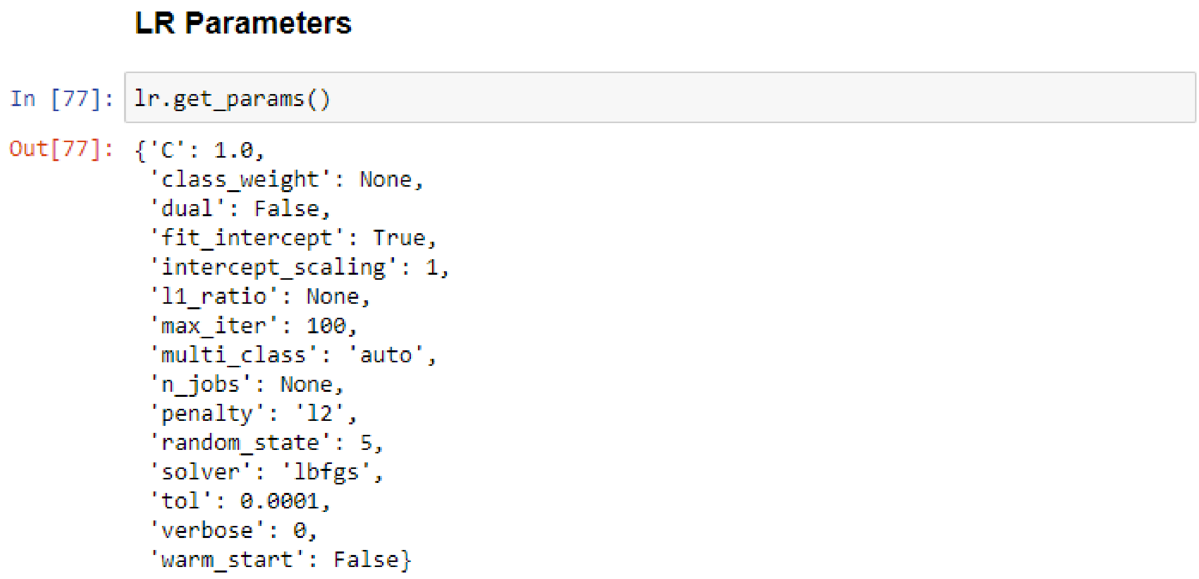
KNN Parameters

```
In [56]: KNN.get_params()

Out[56]: {'algorithm': 'auto',
          'leaf_size': 30,
          'metric': 'minkowski',
          'metric_params': None,
          'n_jobs': None,
          'n_neighbors': 1,
          'p': 2,
          'weights': 'uniform'}
```

The above parameters are based on the approach of ML techniques utilized.

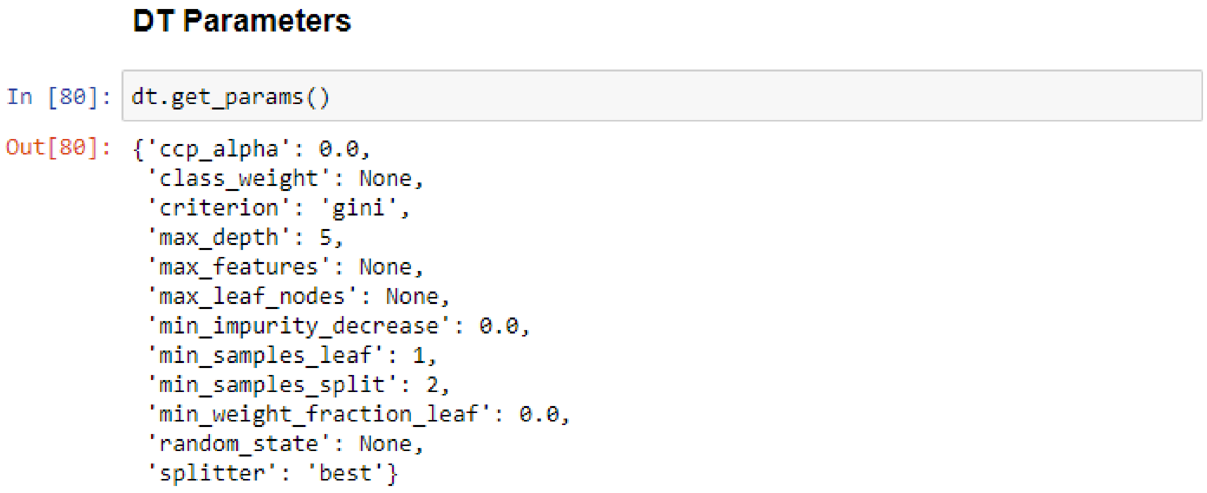
3.4.2 PARAMETERS II

The parameters of the final focus of the study are shown below.



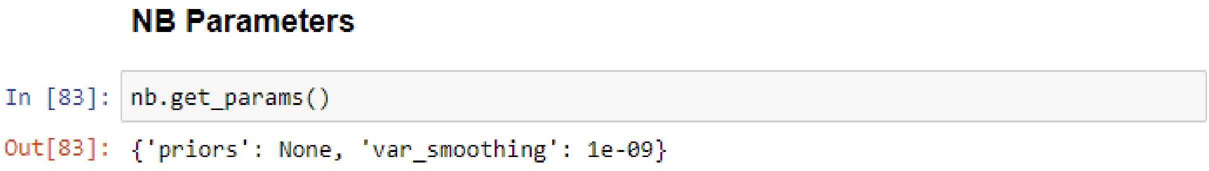
LR Parameters

```
In [77]: lr.get_params()

Out[77]: {'C': 1.0,
          'class_weight': None,
          'dual': False,
          'fit_intercept': True,
          'intercept_scaling': 1,
          'l1_ratio': None,
          'max_iter': 100,
          'multi_class': 'auto',
          'n_jobs': None,
          'penalty': 'l2',
          'random_state': 5,
          'solver': 'lbfgs',
          'tol': 0.0001,
          'verbose': 0,
          'warm_start': False}
```



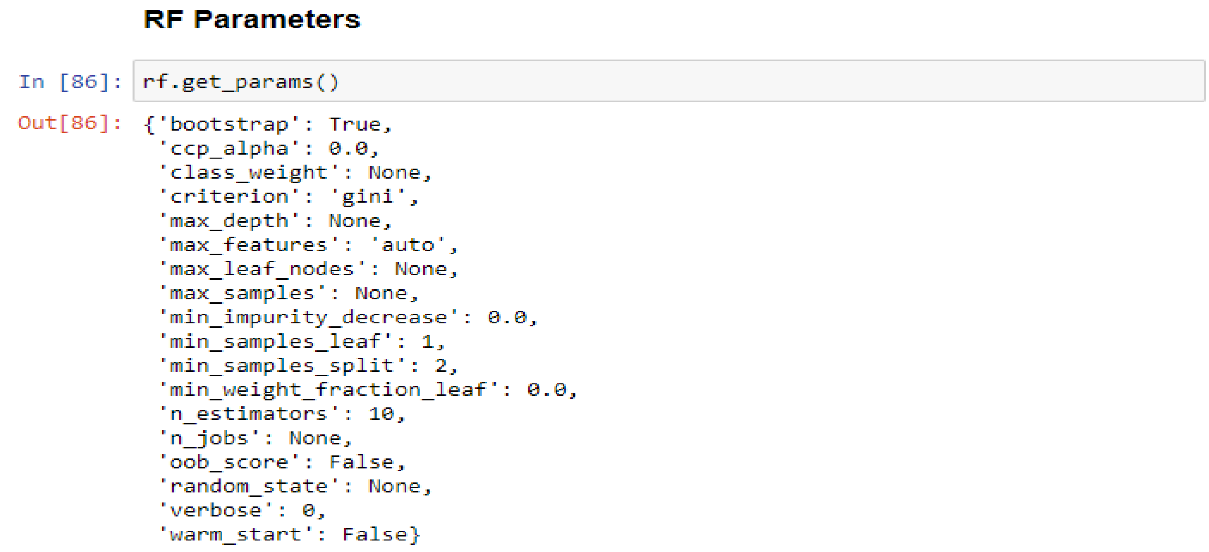
DT Parameters

```
In [80]: dt.get_params()

Out[80]: {'ccp_alpha': 0.0,
          'class_weight': None,
          'criterion': 'gini',
          'max_depth': 5,
          'max_features': None,
          'max_leaf_nodes': None,
          'min_impurity_decrease': 0.0,
          'min_samples_leaf': 1,
          'min_samples_split': 2,
          'min_weight_fraction_leaf': 0.0,
          'random_state': None,
          'splitter': 'best'}
```



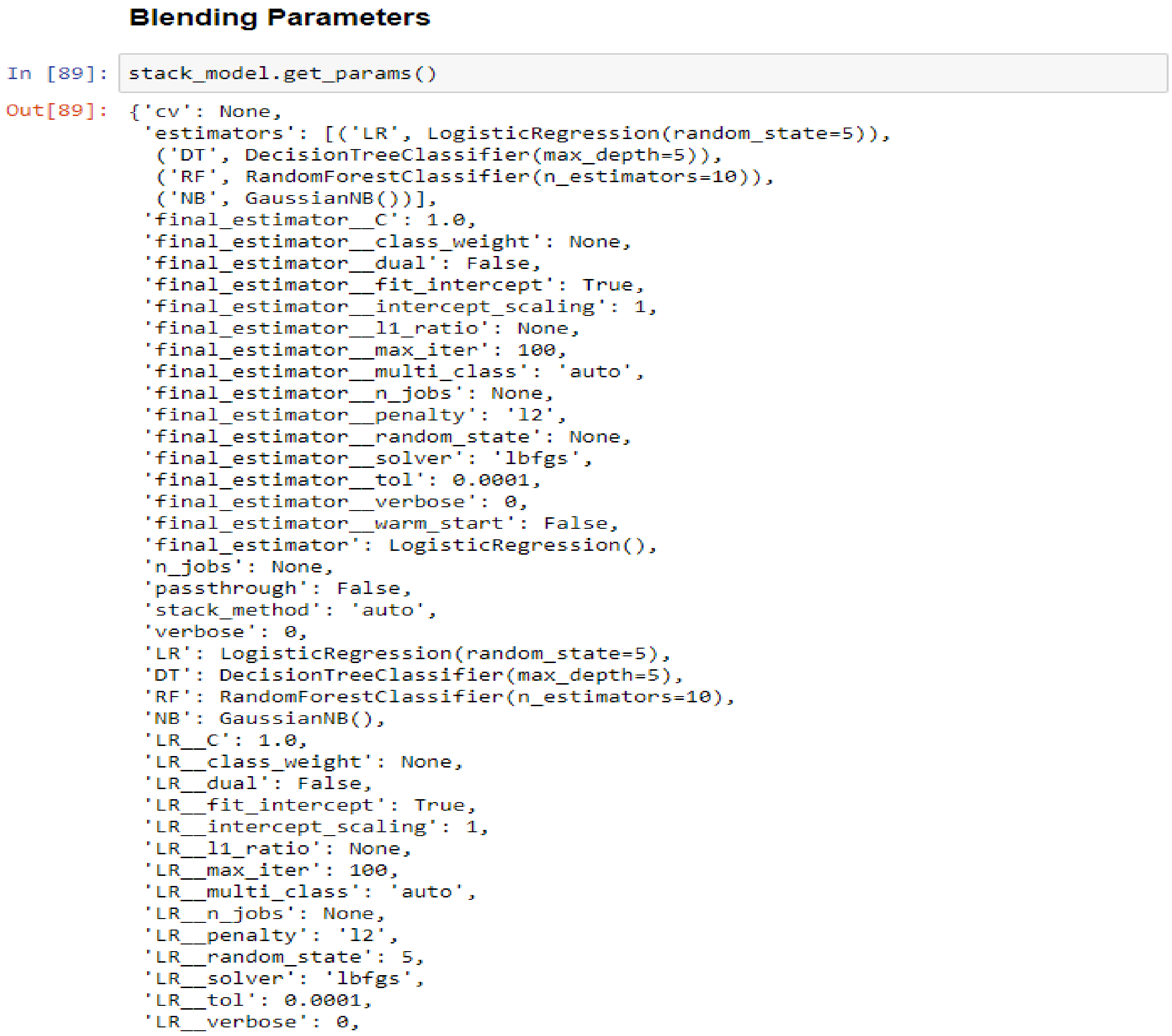
NB Parameters

```
In [83]: nb.get_params()

Out[83]: {'priors': None, 'var_smoothing': 1e-09}
```



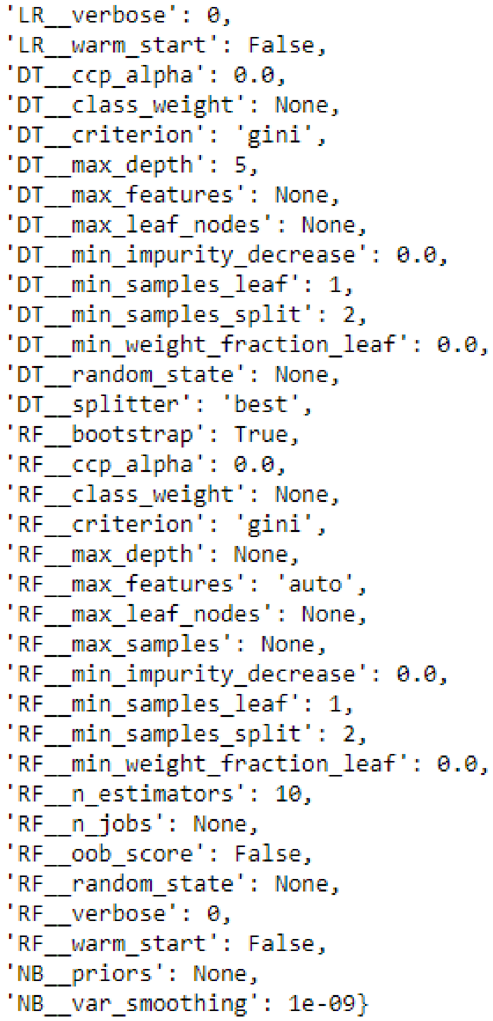
RF Parameters

```
In [86]: rf.get_params()

Out[86]: {'bootstrap': True,
          'ccp_alpha': 0.0,
          'class_weight': None,
          'criterion': 'gini',
          'max_depth': None,
          'max_features': 'auto',
          'max_leaf_nodes': None,
          'max_samples': None,
          'min_impurity_decrease': 0.0,
          'min_samples_leaf': 1,
          'min_samples_split': 2,
          'min_weight_fraction_leaf': 0.0,
          'n_estimators': 10,
          'n_jobs': None,
          'oob_score': False,
          'random_state': None,
          'verbose': 0,
          'warm_start': False}
```



Blending Parameters

```
In [89]: stack_model.get_params()

Out[89]: {'cv': None,
          'estimators': [('LR', LogisticRegression(random_state=5)),
           ('DT', DecisionTreeClassifier(max_depth=5)),
           ('RF', RandomForestClassifier(n_estimators=10)),
           ('NB', GaussianNB())],
          'final_estimator__C': 1.0,
          'final_estimator__class_weight': None,
          'final_estimator__dual': False,
          'final_estimator__fit_intercept': True,
          'final_estimator__intercept_scaling': 1,
          'final_estimator__l1_ratio': None,
          'final_estimator__max_iter': 100,
          'final_estimator__multi_class': 'auto',
          'final_estimator__n_jobs': None,
          'final_estimator__penalty': 'l2',
          'final_estimator__random_state': None,
          'final_estimator__solver': 'lbfgs',
          'final_estimator__tol': 0.0001,
          'final_estimator__verbose': 0,
          'final_estimator__warm_start': False,
          'final_estimator': LogisticRegression(),
          'n_jobs': None,
          'passthrough': False,
          'stack_method': 'auto',
          'verbose': 0,
          'LR': LogisticRegression(random_state=5),
          'DT': DecisionTreeClassifier(max_depth=5),
          'RF': RandomForestClassifier(n_estimators=10),
          'NB': GaussianNB(),
          'LR__C': 1.0,
          'LR__class_weight': None,
          'LR__dual': False,
          'LR__fit_intercept': True,
          'LR__intercept_scaling': 1,
          'LR__l1_ratio': None,
          'LR__max_iter': 100,
          'LR__multi_class': 'auto',
          'LR__n_jobs': None,
          'LR__penalty': 'l2',
          'LR__random_state': 5,
          'LR__solver': 'lbfgs',
          'LR__tol': 0.0001,
          'LR__verbose': 0,
```



```
'LR__verbose': 0,
'LR__warm_start': False,
'DT__ccp_alpha': 0.0,
'DT__class_weight': None,
'DT__criterion': 'gini',
'DT__max_depth': 5,
'DT__max_features': None,
'DT__max_leaf_nodes': None,
'DT__min_impurity_decrease': 0.0,
'DT__min_samples_leaf': 1,
'DT__min_samples_split': 2,
'DT__min_weight_fraction_leaf': 0.0,
'DT__random_state': None,
'DT__splitter': 'best',
'RF__bootstrap': True,
'RF__ccp_alpha': 0.0,
'RF__class_weight': None,
'RF__criterion': 'gini',
'RF__max_depth': None,
'RF__max_features': 'auto',
'RF__max_leaf_nodes': None,
'RF__max_samples': None,
'RF__min_impurity_decrease': 0.0,
'RF__min_samples_leaf': 1,
'RF__min_samples_split': 2,
'RF__min_weight_fraction_leaf': 0.0,
'RF__n_estimators': 10,
'RF__n_jobs': None,
'RF__oob_score': False,
'RF__random_state': None,
'RF__verbose': 0,
'RF__warm_start': False,
'NB__priors': None,
'NB__var_smoothing': 1e-09}
```

The above parameters are based on algorithms ensembled for the process of Implementing a hybrid-model.

3.5 REQUIREMENTS

In this section, we provide an overview of the system requirements used to perform our experiments. Requirements are divided into two categories: hardware and software requirements.

3.5.1 Hardware Requirements:

These are the hardware requirements of a device (HP all in one PC) used.
Processor – i5 (11th Gen Intel(R) Core (TM) i5-1135G7 @ 2.40GHz   2.42 GHz).
Random access memory (RAM) – 8.00 GB.
System Free Space – Minimum 15GB.
System type – 64-bit operating system, x64-based processor

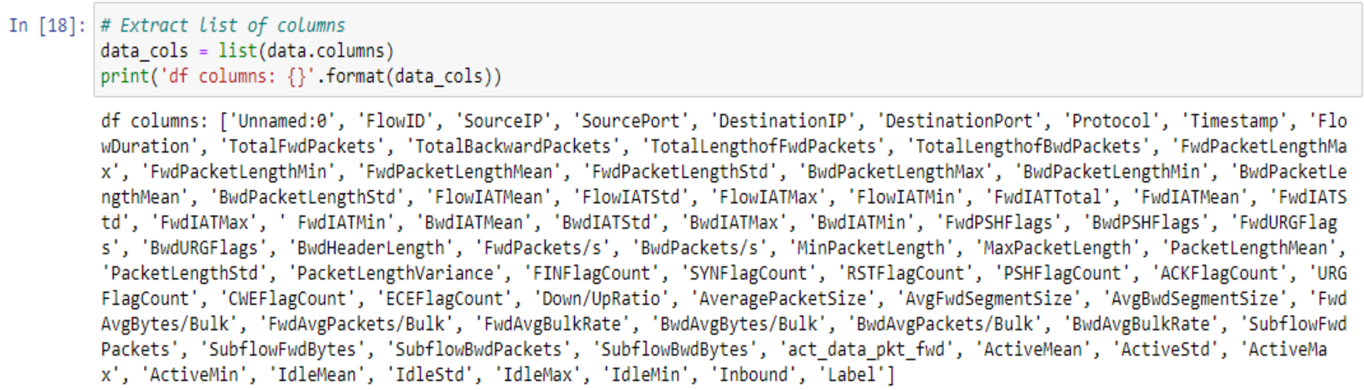
3.5.2 Software Requirements:

Programming Language – Python (as explained in Chapter 1). In software development, an integrated development environment (IDE) facilitates the development of software code. Hence, in our case IDE software utilized is: Jupyter Notebook (ANACONDA).

3.6 PROPOSED & IMPLEMENTED SCHEMES

3.6.1 Design Approach

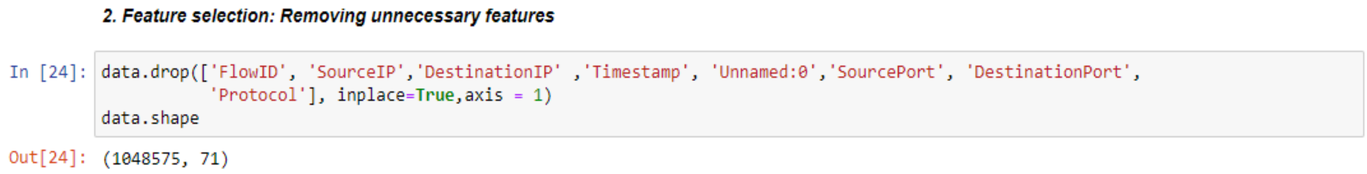
The proposed ML approaches are preferred to overcome the challenges of DDoS (DNS flooding) attacks on MEC environment. The proposed system analyses a dataset [5] with 79 features:



```
In [18]: # Extract list of columns
         data_cols = list(data.columns)
         print('df columns: {}'.format(data_cols))
```

```
df columns: ['Unnamed:0', 'FlowID', 'SourceIP', 'SourcePort', 'DestinationIP', 'DestinationPort', 'Protocol', 'Timestamp', 'Flo
wDuration', 'TotalFwdPackets', 'TotalBackwardPackets', 'TotalLengthofFwdPackets', 'TotalLengthofBwdPackets', 'FwdPacketLengthMa
x', 'FwdPacketLengthMin', 'FwdPacketLengthMean', 'FwdPacketLengthStd', 'BwdPacketLengthMax', 'BwdPacketLengthMin', 'BwdPacketLe
ngthMean', 'BwdPacketLengthStd', 'FlowIATMean', 'FlowIATStd', 'FlowIATMax', 'FlowIATMin', 'FwdIATTotal', 'FwdIATMean', 'FwdIATS
td', 'FwdIATMax', ' FwdIATMin', 'BwdIATMean', 'BwdIATStd', 'BwdIATMax', 'BwdIATMin', 'FwdPSHFlags', 'BwdPSHFlags', 'FwdURGFlag
s', 'BwdURGFlags', 'BwdHeaderLength', 'FwdPackets/s', 'BwdPackets/s', 'MinPacketLength', 'MaxPacketLength', 'PacketLengthMean',
'PacketLengthStd', 'PacketLengthVariance', 'FINFlagCount', 'SYNFlagCount', 'RSTFlagCount', 'PSHFlagCount', 'ACKFlagCount', 'URG
FlagCount', 'CWEFlagCount', 'ECEFlagCount', 'Down/UpRatio', 'AveragePacketSize', 'AvgFwdSegmentSize', 'AvgBwdSegmentSize', 'Fwd
AvgBytes/Bulk', 'FwdAvgPackets/Bulk', 'FwdAvgBulkRate', 'BwdAvgBytes/Bulk', 'BwdAvgPackets/Bulk', 'BwdAvgBulkRate', 'SubflowFwd
Packets', 'SubflowFwdBytes', 'SubflowBwdPackets', 'SubflowBwdBytes', 'act_data_pkt_fwd', 'ActiveMean', 'ActiveStd', 'ActiveMa
x', 'ActiveMin', 'IdleMean', 'IdleStd', 'IdleMax', 'IdleMin', 'Inbound', 'Label']
```

Using feature selection, some features are left out of the training of DNS datasets that we collected. The section below outlines those features that were not utilised.



2. Feature selection: Removing unnecessary features

```
In [24]: data.drop(['FlowID', 'SourceIP','DestinationIP' ,'Timestamp', 'Unnamed:0','SourcePort', 'DestinationPort',
                    'Protocol'], inplace=True,axis = 1)
         data.shape

Out[24]: (1048575, 71)
```

In order to detect DDoS attacks, we classified the values based on the significant features using ML techniques including RF, NB, DT, KNN, and LR.

3.6.2 Classification Block

Block diagrams illustrate the relationships between components (ML's, DDoS attacks, and server) of a system using blocks connected by lines that show their relationships. The diagram displayed below shows the classification block [41]

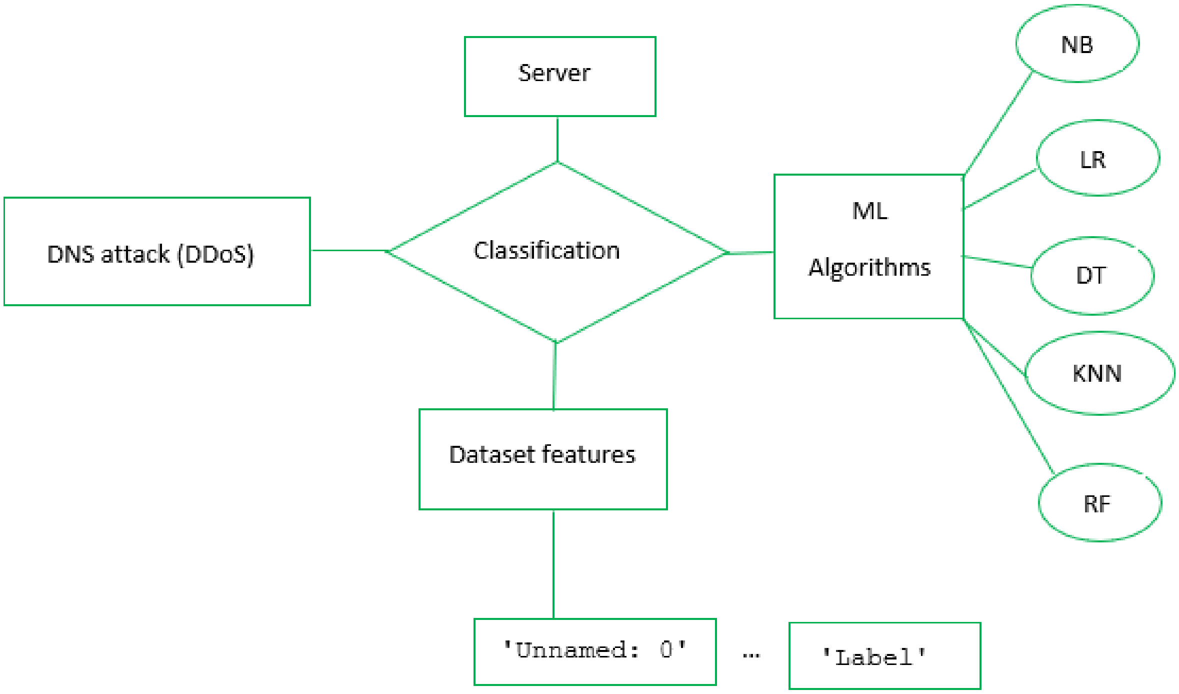


Figure 3.4 Classification Block Diagram *[41]*

3.6.3 MITIGATION SCHEME

High level overview of supervised ML algorithms is provided in detail. SL is also known as supervised ML which is category of ML, and artificial intelligence (AI). SL are methods that utilize labelled datasets for training algorithms that classify data or predict future outcomes. The reasons for choosing the below MLs are outlined under each algorithm.

a) NB

Naïve Bayes (NB) is a SL algorithm based on the Bayes theorem (Bayes' rule or Bayes' law) and is used for solving classification problems. Naïve Bayes Classifier is one of the easiest and most effective classification algorithms. It can be used to build fast ML models that can predict quickly. The Naive Bayes classifiers aim to process, analyse, and categorize data using probabilistic methods. There are three types of the Naïve Bayes Model:

Gaussian NB (GaussianNB) model assumes that features follow a normal distribution. In this case, if predictors take continuous values instead of discrete ones, then the model assumes that they are drawn from a Gaussian distribution. Hence, this is best suited for our study.

Multinomial NB (MultinomialNB) classifiers are used in the case of multinomial data distributions. Classifiers are primarily used to classify documents. The predictors are based on word frequencies.

Bernoulli NB (BernouliNB) classifier works similarly to the Multinomial classifier, but the predictor variables are independent Boolean variables. A document can be examined to determine if a specific word is present or not. Document classification tasks are also well suited to this model.

b) LR

Logistic regression (LR) is a widely used ML algorithm classified under SL. It enables the prediction of a categorical dependent variable using a set of independent variables. LR specifically focuses on predicting the outcome of categorical dependent variables, which means the result must be a discrete or categorical value. Examples of such values include "Yes" or "No" and "0" or "1".

c) DT

Decision Trees (DT) are SL techniques that can be used to solve either classification or regression problems; however, they are mostly used to solve classification problems. This is the reason for proposing it in this study. An internal node represents the features in a dataset, a branch represents the decision rules, and a leaf node represents the outcome.

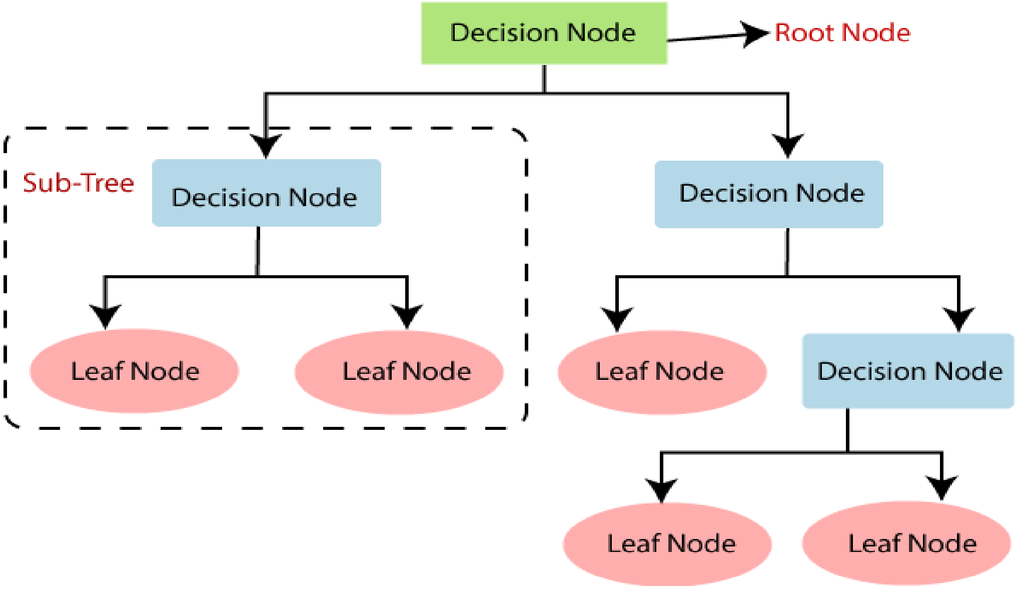


Figure 3.5 DT diagram classification *[42]*

d) KNN

A K-Nearest Neighbour (KNN) algorithm is a simple ML algorithm based on SL. K-NN algorithm can be used for both regression and classification, but it is mostly used for classification. In K-NN, there are no assumptions made about the underlying data, which makes it a non-parametric algorithm. K-NN is a lazy learner algorithm that stores the training data and performs actions on it at the time of classification instead of learning from data immediately.

3.6.4 IMPLEMENTED HYBRID MODEL.

Ensemble learning is one of the best ML techniques for solving problems involving computation intelligence by combining the outputs of several models and weak learners. For example, the RF algorithm is a combination of various DT's. There are various ways of ensemble learning which include **tagging**, **boosting**, and **stacking** [42].

Among the ensembling techniques mentioned above, our study used Stacking. Stacking is an ensemble ML method for predicting multiple nodes, which is used to build a new model and improve the performance of the model. The hybrid model is implemented by integrating LR, DT, NB, and RF. The purpose of implementing a hybrid-model in our study is to enhance the accurate detection of DDoS attacks.

a) Stacking Ensembling Technique

The stacking ensemble technique considers the collective predictions made by multiple weaker learners and meta learners to generate an enhanced output prediction model. A Blending algorithm takes inputs from sub-models and attempts to combine them into a better output prediction by learning how to combine them. Stacking/Blending process is also called stacked generalization.

- **Architecture of Stacking**

The architecture of the stack model comprises two or more base/learner models along with a meta-model that combines predictions from these base models. The base models, also known as level 0 models, are accompanied by level 1 models, referred to as meta-models. Consequently, the stacking ensemble method incorporates the original training data, primary level models, primary level predictions, secondary level models, and final predictions. Figure 3.6 represents the architecture of Stacking.

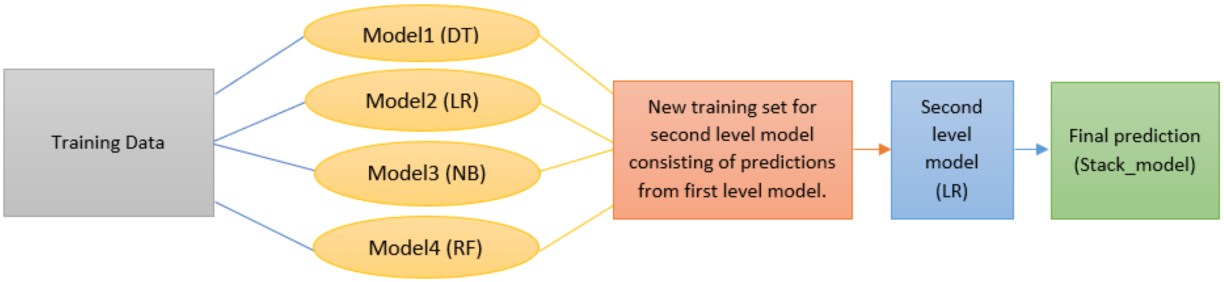

Training Data
Model1 (DT)
Model2 (LR)
Model3 (NB)
Model4 (RF)
New training set for second level model consisting of predictions from first level model.
Second level model (LR)
Final prediction (Stack_model)

Figure 3.6 Stacking Diagram

The original data must be split into n-folds and can also be considered a training dataset or a testing dataset.

Base models: they are also called level-0 models. Training data are used in these models and compiled to yield the outputs of (level 0) predictions.

Level-0 Predictions: based on training data, each base model provides a different prediction, known as a level-0 prediction.

Meta Model: The architecture of the stacking model includes a meta-model (LR) that effectively combines predictions from the base models (DT, NB, and RF). The meta-model, alternatively referred to as a level-1 model, plays a crucial role in this process.

Level-1 Prediction: A meta-model is trained to merge predictions from various base models by utilizing predictions generated from each base model. These predictions are computed using separate data that was not utilized in training the base models. The predictions, along with their corresponding expected outcomes, are used as input and output pairs for training the meta-model. The meta-model is then trained on this dataset to make more precise predictions based on the predictions of the base models.

b) RF

Random Forest (RF) is an ensemble learning method for classifying, predicting, and regressing which involves constructing a multitude of DT's during training of the dataset. RF can be classified as one of the good examples of ensemble ML method, as it combines various DTs to produce a more generalized model. RF creates random subsets of the features.

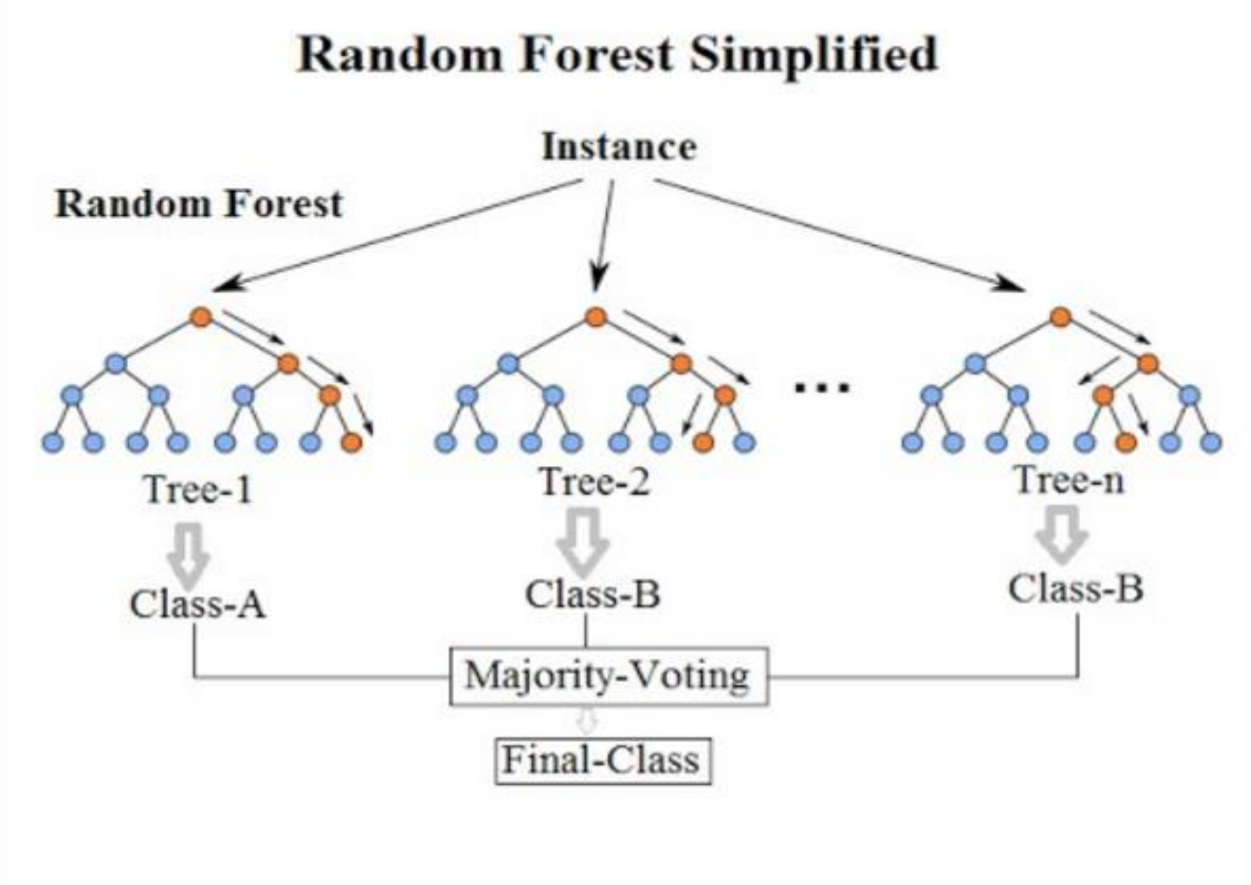


Figure 3.7 Random Forest Diagram *[43]*

3.6.5 PROPOSED SYSTEM ARCHITECTURE

To acquire fresh training data, the DrDoS DNS dataset is partitioned into training and test data. The new training data is subsequently further divided into a training set and a validation set. The blending approach follows a comparable methodology to stacking. Four classification models are used to make these predictions, including LR, DT, RF, and NB. There are two sets of predictions provided by each model, namely validation predictions and test predictions. Consequently, the validation predictions from the four models are combined into a unified validation input. Similarly, the predictions from all four models are merged into a combined test input. After obtaining the validation input, the newly collected data is trained using the LR technique. A final prediction is made using the LR technique, which is compared to actual test data to determine the accuracy of the final prediction.

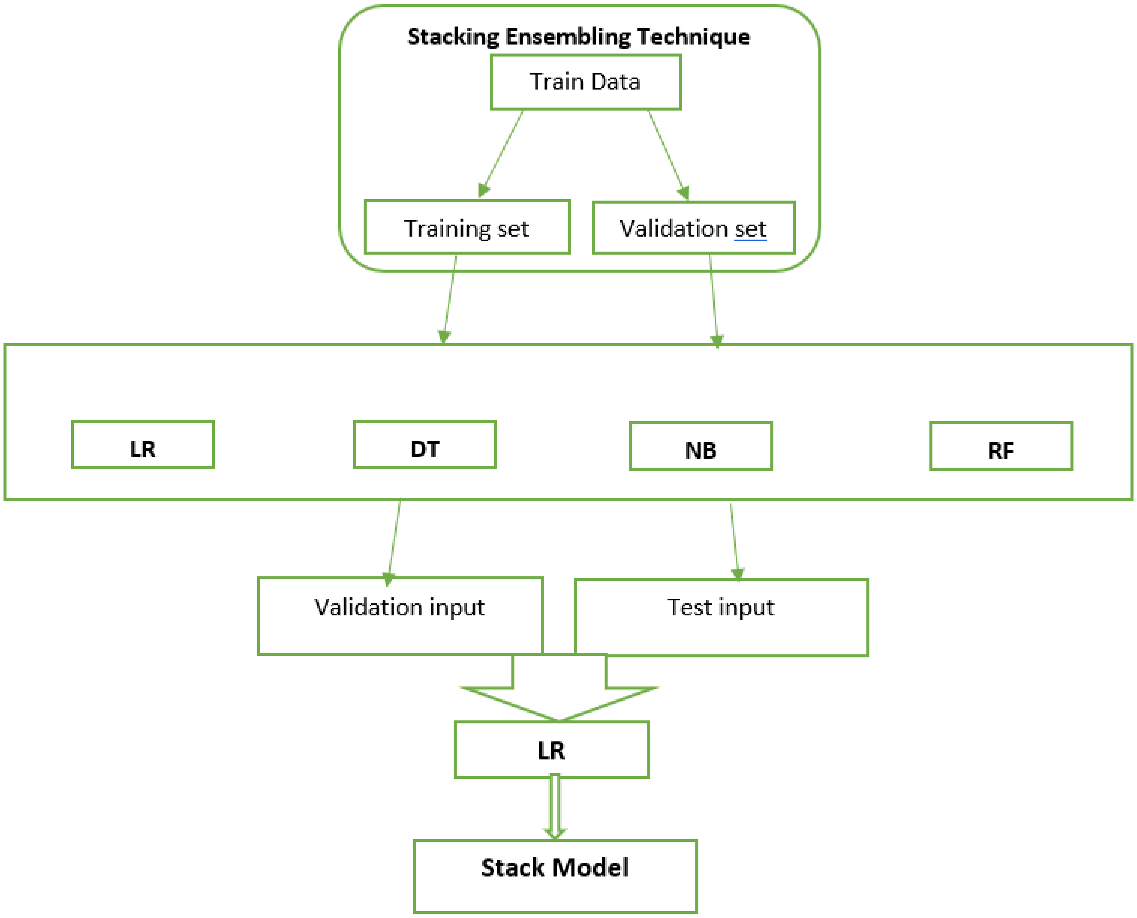


Figure 3.8 Proposed System Architecture *[41]*

3.7 PERFORMANCE METRICS

The suggested solution's performance is evaluated using various metrics, including precision, recall, F1-Measure, Detection Rate (DR), accuracy, Area Under the Receiver Operating Characteristics (AUROC), and Mathew Correlation Coefficient (MCC). These performance measures are represented by TP (True Positive), FP (False Positive), TN (True Negative), and FN (False Negative). Our study gives an overview of the mentioned metrics regarding the model for classifying binary data.

3.7.1 Accuracy

Accuracy is a metric used to evaluate the correct classification of occurrences as either normal or attacks.

$$Accuracy = \frac{TP + TN}{FP + FN + TP + TN} \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots . (i)$$

3.7.2 Precision

Precision, in the context of a classification algorithm, refers to the positive predictive value. It is calculated by dividing the number of actual positive results by the number of positive results predicted by the algorithm:

$$Precision = \frac{TP}{TP + FP} \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots .. (ii)$$

3.7.3 The Detection Rate

Detection Rate represents the proportion of correctly detected attacks from the entire set of attacks included in the dataset.

$$Detection\ Rate\ (DT) = \frac{TP}{TP + FN} \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots (iii)$$

3.7.4 Recall/TPR/Detection

Recall, also known as True Positive Rate (TPR) or Detection Rate (DR), aims to compare the True Positive (TP) items against the False Negative (FN) items that were not classified correctly. The mathematical formula of recall is given in the equation:

$$Recall = TPR\ = Detection = \frac{TP}{TP + FN} \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots (iv)$$

When evaluating the performance of accuracy and recall, it is crucial to consider the trade-off between the two. If one algorithm exhibits low recall but high precision, it may be necessary to explore alternative algorithm techniques and determine the most suitable one to use.

3.7.5 F1-Measure Score

To address this issue, the F1-score is utilized, which provides an average of recall and precision. The F1-score can be calculated using the formula provided.

$$F1 - Measure = 2 \times \frac{Precision \ \times Recall}{Precision \ + Recall} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots . (v)$$

3.7.6 Matthews Correlation Coefficient (MCC)

In general, MCC is considered one of the best metrics for measuring the performance of classification models. This is mostly because, unlike earlier metrics, it takes into account all possible predicted outcomes. This means that if there is an imbalance between classes it will be considered. A MCC is essentially a correlation coefficient between observed and predicted classifications, and it is denoted as follows:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (vi)$$

3.8 FINDINGS INTERPRETATION

For the interpretation of the results, our study made use of probability density function or probability distribution function (PDF), Area Under Receiver Operating Characteristic (AUROC) or AUC, and hypothesis testing.

3.8.1 ROC

A receiver operating characteristic (ROC) curve represents the model's performance at all classification thresholds. Area Under Curve (AUC) measures the entire two-dimensional area under the curve and is therefore a measure of the model's performance across all classification thresholds. As ROC curves plot the accuracy of the model, they are ideal for diagnosing models with unbalanced data.

ROC curve is generated by plotting TP against FP at different thresholds. ROC curves are generated by plotting the cumulative distribution function of TP (y-axis) against the cumulative distribution function of FP (x-axis).

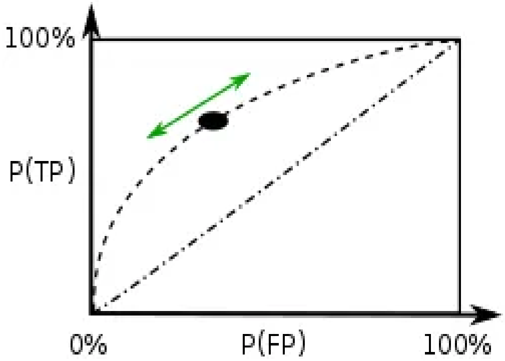


Figure 3.9 AUROC curve1 [52]

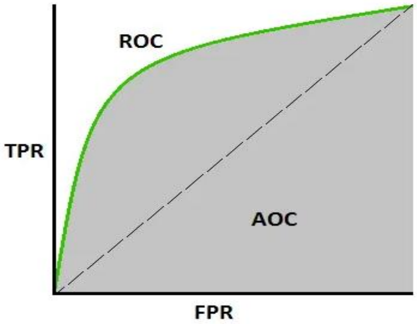


Figure 3.10 AUROC curve2 [52]

For evaluating performance, the area under the ROC curve (AUROC) is used. Generally, the higher the AUC, the better the model is at distinguishing between classes. Generally, an AUC value of 0.5 implies no discrimination, an AUC value between 0.5–0.7 is acceptable and an AUC value above 0.7 indicates a good-to-go model.

3.8.2 PDF

PDF is used to determine the probability of all possible outcomes for a random variable. Depending on the values that a random variable takes, a distribution can either be continuous or discrete distribution. There are different types of probability distribution namely, Gaussian or normal distribution, uniform distribution, and exponential distribution. Our study made use of normal distribution, which is characterized by two parameters, namely:

- Mean (μ) – It shows where the distribution is centred.
- Standard deviation (σ) - Used to measure the spread in a curve.

Normal or Gaussian distribution can be calculated using the following formula.

$$f(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(i)$$

The plot of normal distribution can be seen by properties such as symmetric mean, and the graph design is in the form of a bell-shaped curve. The PDF of normal distribution general view is represented below.

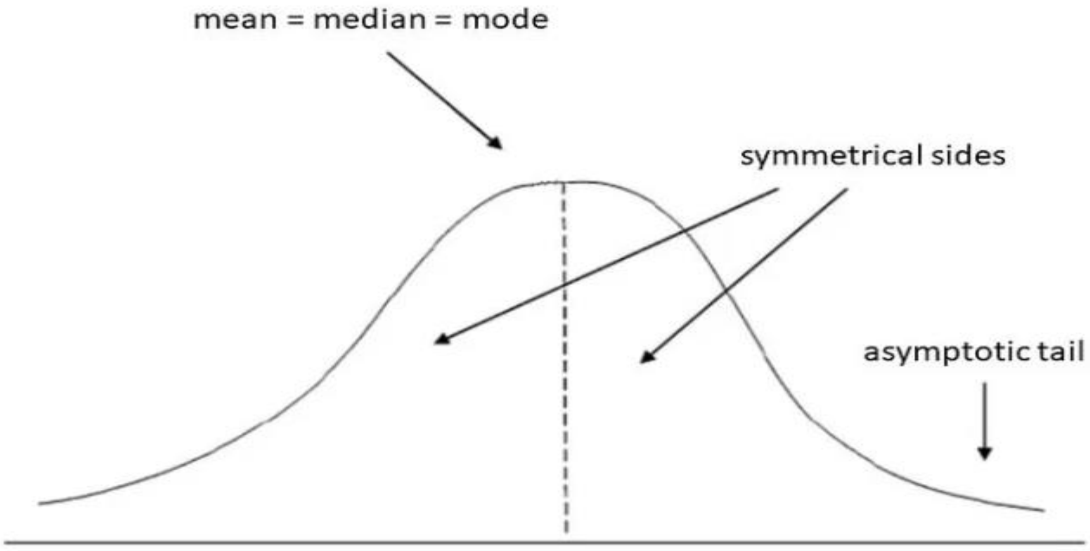


Figure 3.11 PDF of Normal Distribution

3.8.3 Hypothesis Testing

Hypothesis test involves determining if the results of a research study confirm a particular theory about a population. Hypothesis is a premise or a claim that we want to test. Sample data is used to evaluate a hypothesis about a population. Hypothesis testing consists of four steps, each clearly explained below.

Step 1: Identify the hypothesis and verify conditions. State the null hypothesis (H_0), $H_0$is assumed to be true until there is no evidence to the contrary. State the research or alternative hypothesis (H_a), H_a this hypothesis involves the claim to be tested.

Step 2: Decide a significance level (α) to use as a probability cut-off when deciding about the null hypothesis. Our alpha (α) value represents the probability risk of rejecting the null hypothesis incorrectly if we make an incorrect decision. Alpha (α) formula is represented as follows:

$$\alpha = 1 - C \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (i)$$

Where C is confidence interval

Step 3: Compute Z-test statistic (z_t) for population proportion. Obtain sample data, create a test statistic, and compare the result to the parameter value. The test statistic contains a measure of standard error and assumptions (conditions) relating to the sampling distribution and is calculated under the premise that the null hypothesis is true. Z-test statistic (z_t) is denoted as follows:

$$z_t = \frac{\hat{p} - P_0}{\sqrt{\frac{p_0(1-P_0)}{n}}} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots..\dots\dots\dots\dots. (ii)$$

$Where;\ \hat{p}\ is\ the\ sample\ proportion, and\ is\ computed\ as\ follows$

$$\hat{p} = \frac{X}{n} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots.\dots\dots\dots (iii)$$

$Where;\ X\ is\ the\ count\ of\ successes.$

$Where;\ Po\ is\ the\ poulation\ proportion.$

$Where;\ n\ is\ the\ size\ of\ the\ population.$

Step 4: Find the rejection zone or compute p-value (probability value) The test statistic is used to determine a p-value by estimating the likelihood that the sample data will result in such a test statistic or one more extreme. By employing alpha to locate a critical value, the rejection zone may be identified; it is the area that is more severe than the critical value. Decide if the null hypothesis is true. In this phase, we decide on whether to reject the null hypothesis or not. Give a broad conclusion: Once the p-value or rejection region has been determined and the null hypothesis has been statistically determined.

3.9 SUMMARY & CONCLUSION

The results of the simulation for the ML algorithms (DT, LR, NB, KNN, RF, and Stack model) are presented using Microsoft Excel. The results were obtained using Jupyter notebook and the Python programming language. Based on the analysis of the findings, each algorithm's accuracy in detecting DDoS attacks is determined. In addition, we compare the proposed approaches for detecting DDoS attacks. These details are discussed in Chapter 4 of the study.

Several types of DDoS attacks are discussed in this chapter, including those at the network and application layers. It also provides an overview of the dataset simulation tools that were employed to obtain the results. The performance measures of each ML algorithm are thoroughly examined and discussed.

CHAPTER 4: DATA ANALYSIS

4.1 INTRODUCTION

This chapter presents the results obtained during DDoS attacks detection simulations on MEC, and the performance comparison results of various ML algorithms in mitigating DDoS attacks. The chapter further presents comparison results of the implemented hybrid models (RF and Stacking model) against existing ML algorithms (LR, DT, KNN and NB) in detecting DDoS attacks. The performance metrics utilized to check the effectiveness of supervised ML algorithms (Stacking model, RF, LR, DT, NB, and KNN) are accuracy, detection rate, precision, F1-Measure, MCC, and AUROC which are expressly displayed in the form of figures and transformed in chart which is subsequently explained. Metrics utilized to check the performance of the existing ML algorithms are accuracy, detection rate, precision, and F1-measure, whereas for checking the performance of the hybrid models we utilized accuracy, MCC, F1-score, recall, and AUROC.

The findings are used to inform conclusions on which mitigation method performs best in detecting DDoS attacks (DNS flooding attacks) on MEC. The reviewed papers [33] [34] [38] [35] separately concluded that ML algorithms, and the hybrid model, are in their own way the best in detecting DDoS attacks. Our study seeks to determine which method is more optimal/efficient in detecting DDoS attacks between the ML algorithms, and hybrid model.

4.2 RESULTS & DISCUSSION

The proposed DDoS attacks mitigation schemes are supervised ML algorithms which are namely: LR, DT, NB, KNN, and RF, and then the stacking model (hybrid-model). We determined the parameters of each SL and separated the dataset into training and testing at the rate of 0.70 and 0.30 respectively. The dataset was imbalanced (we had 1046567 of DrDoS_DNS attacks, and 2008 BENIGN) which implies that our target variable was significantly imbalanced. The dataset [6] contained a lot of DrDoS_DNS (DDoS attacks) with 99.8% and less benign network activity with 0.2% as shown in Figures 4.1 and 4.2 Label is categorized in two ways, namely: Benign and DrDoS_DNS. Label is our target variable from dataset used to perform our experiment, where 0 represents BENIGN, and 1 represents DrDoS_DNS.

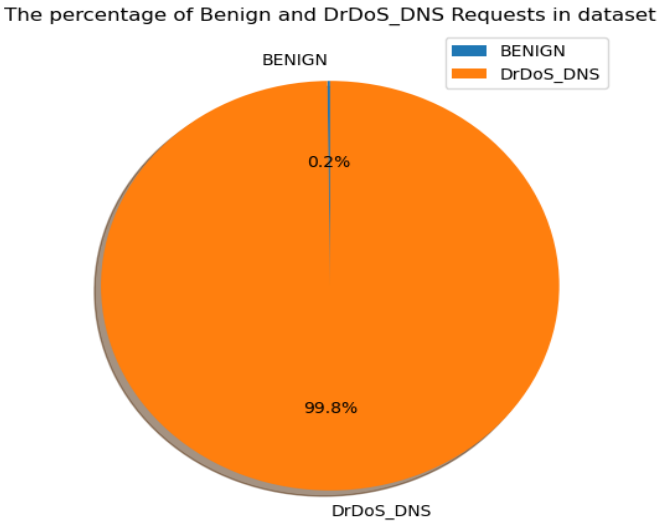


Figure 4.1 Imbalanced dataset 1

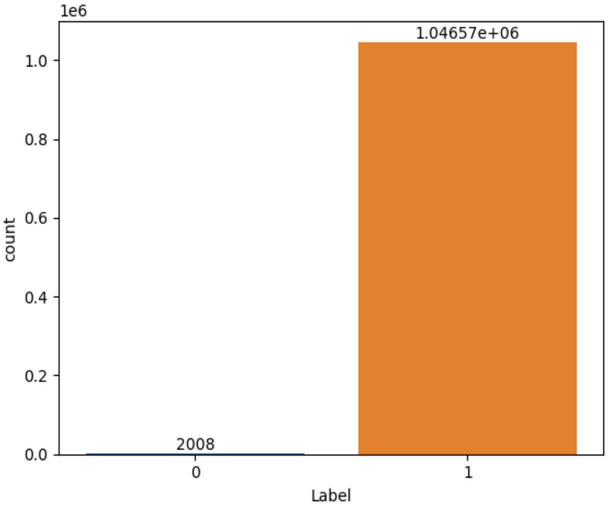


Figure 4.2 Imbalanced dataset2

We had to balance the dataset to make training of the SL techniques easier to detect the DDoS attacks since it also helps the trained model not to be biased based towards one class. Hence, we balanced our dataset using the random under-sampling (RUS) technique. After applying the RUS technique for balancing our dataset [6], the target (Label) variable got balanced, and the resampled dataset then contained equal DrDoS_DNS of 1406 and BENIGN of 1406 network activity, which is 50%**:**50% as shown in the Figures 4.3 and 4.4.

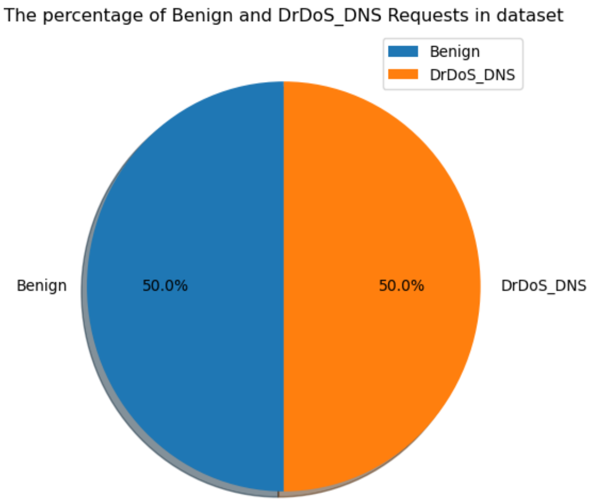


Figure 4.3 Balanced dataset1

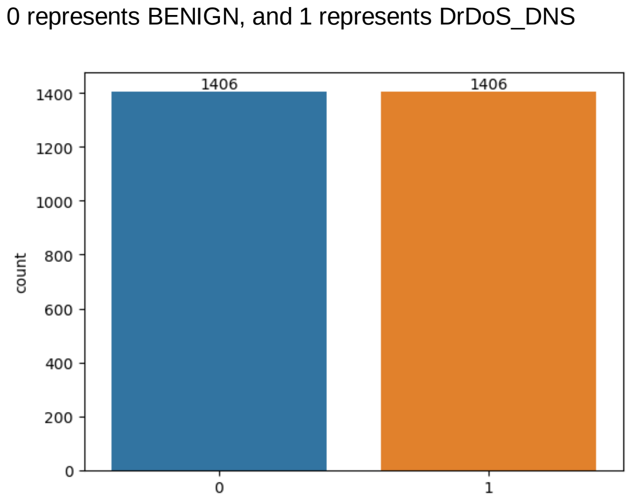
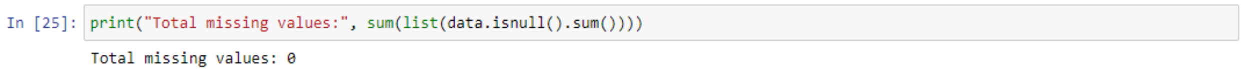


Figure 4.4 Balanced dataset2

Based on the exploratory data analysis (EDA) requirement, when using supervised ML's, there are a few principles to fulfil before training or testing each SL technique. To simplify EDA, we had to perform feature selection in our dataset. The feature selection applied in our study is called data processing or data cleaning a) The data must not have duplicate records: we dropped duplicate records in our dataset. b) We removed unnecessary features in our dataset (see dropped features in Chapter 3). c) The target (Label) variable must be converted to numerical values, and we used label encoding to convert: DrDoS_DNS = 1, and BENIGN = 0. d) The data must not have missing values or null values. Ultimately, our data did not contain any missing/null values.



```
In [25]: print("Total missing values:", sum(list(data.isnull().sum())))

         Total missing values: 0
```

Since we used Sklearn, it was necessary to clean up null values before passing our data to the ML framework. Otherwise, we would have got a long and confusing error message. After performing a check analysis, we found that our dataset did not have any null values as shown in Figure 4.5.

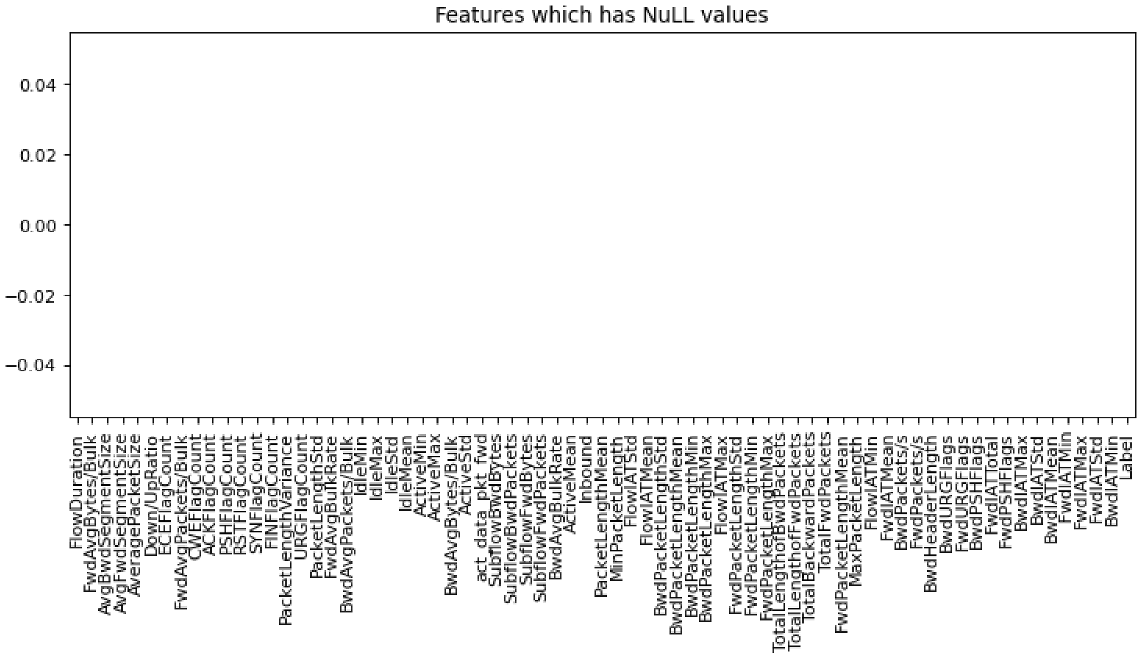


Figure 4.5 Null/Missing Values of feat

Correlation coefficient is a numerical measure of relationships between variables (features). The correlation ranges from -1 to +1. When the correlation is -1.0, it indicates a perfect negative correlation, and when it is +1.0, it indicates a perfect positive correlation. Lastly, if the correlation coefficient is equal to zero, this indicates that there is no relationship between the features. For clear visualization, only a few features were selected.

In Figure 4.6 we represented the correlation coefficient based on a few features selected from the data frame.

In **Figure 4.7** we graphically represented correlation coefficient based on some selected feature. However, both figures clearly show that there was a strong relationship between the selected features.

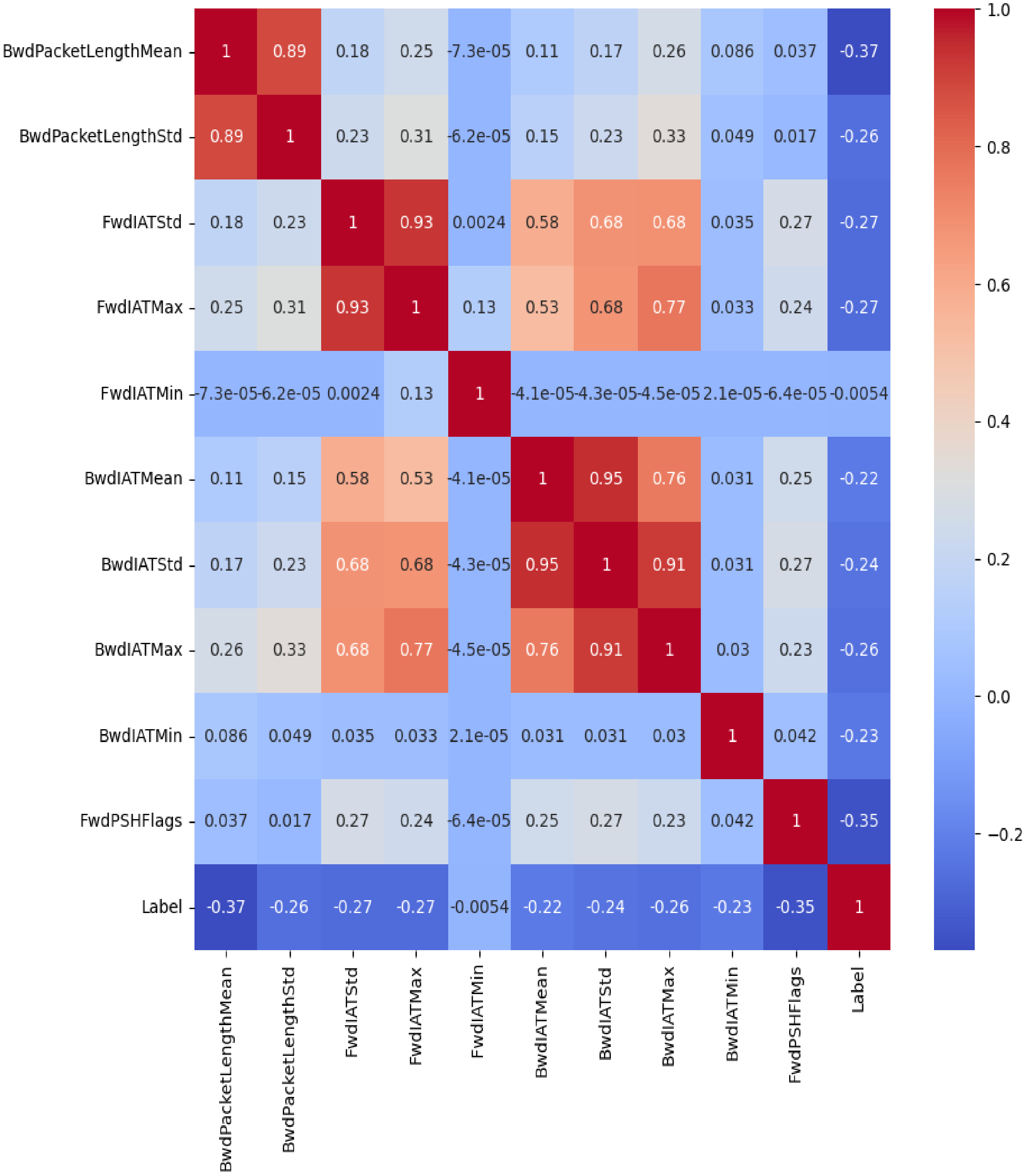


Figure 4.6 Heat map correlation

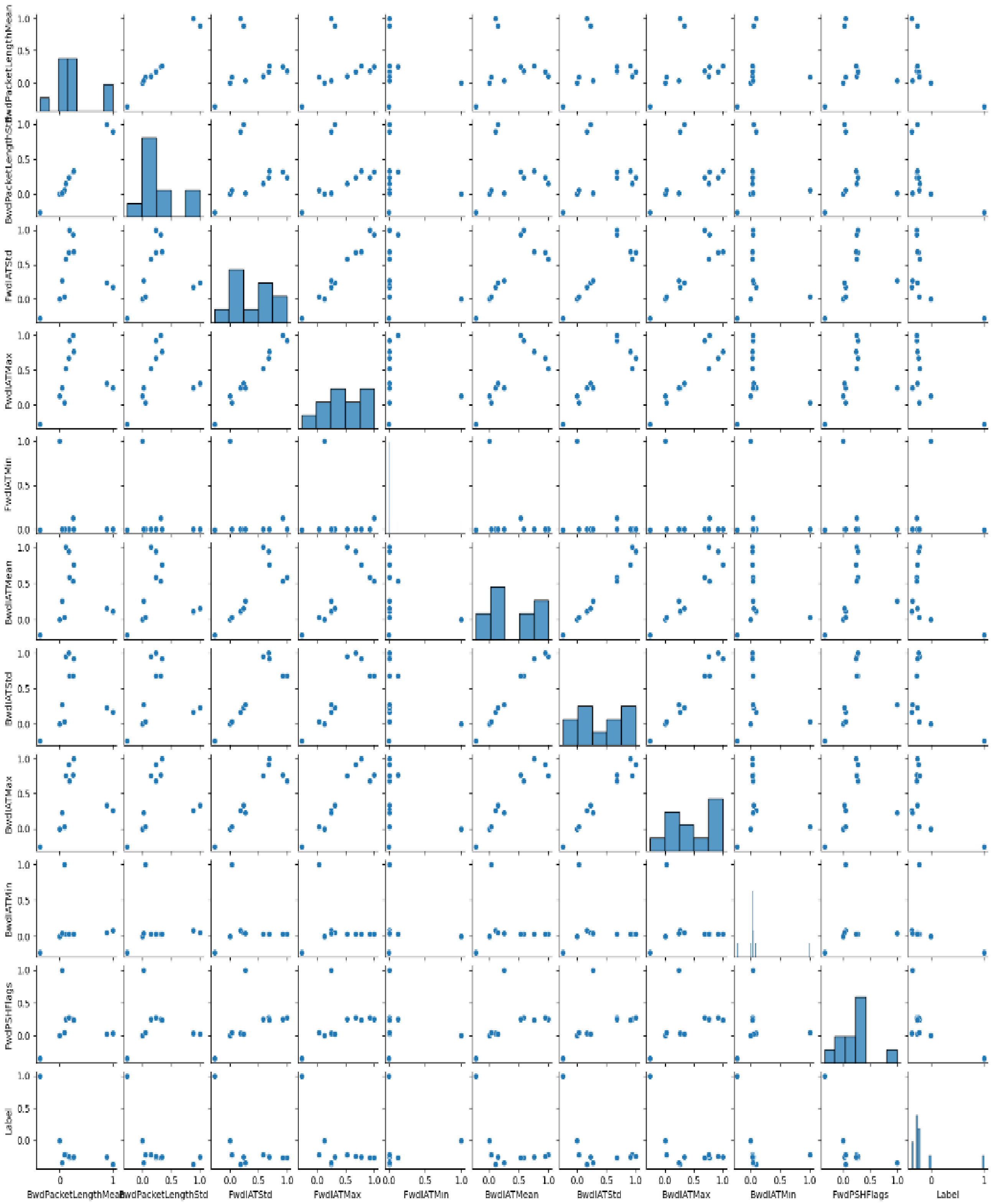


Figure 4.7 Heat map visualization

Based on our literature review, it can be submitted that by using ML algorithms, and hybrid model, systems can learn from data, and they are fed without being programmed, which establishes hidden patterns and leads to better insights, as shown in Figure 4.8.

Figures 4.8 and 4.9 represent the detection methodology process of the DDoS attack, both for the existing ML algorithms as well as the implemented hybrid models. Figure 4.8 presents the ML algorithms (LR, DT, NB, RF, and KNN) that we built from the 1st case of the study. Figure 4.9 presents the ML algorithms (LR, DT, NB and RF) stacked to implement hybrid model, and this is actually the 2nd case of the study.

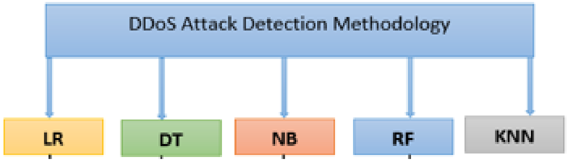


Figure 4.8 The utilized ML Algorithms 1st Case of the Study

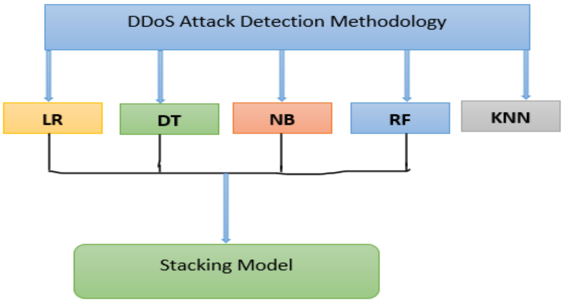


Figure 4.9 The utilized supervised learning 1 for 2nd Case of the Study

Important concepts

- Supervised ML models are also called SL model (Stack model, RF, DT, NB, LR, & KNN).
- ML models (DT, NB, LR, & KNN).
- Hybrid models (RF, and Stack model)

AI Technique: When ML technique is trained or tested with inputs it is called ML model, otherwise called ML algorithm.

4.3 The 1st CASE OF THE STUDY: ML Algorithms

The experimental findings acquired by training, and testing ML techniques on the dataset are presented in this section. The performance metrics are utilized to detail the performance of each algorithm. The findings are presented in Figure: 4.10

The following metrics were used in this study: Precision, Recall, Accuracy, and F1-Measure. The findings show that hybrid model (RF) is more efficient or optimal in dealing with the DDoS attacks compared to the state-of-the-art models LR, DT, NB, and KNN. Below we define the performance metrics utilized to evaluate each supervised ML algorithm (RF, LR, NB, KNN, and DT)

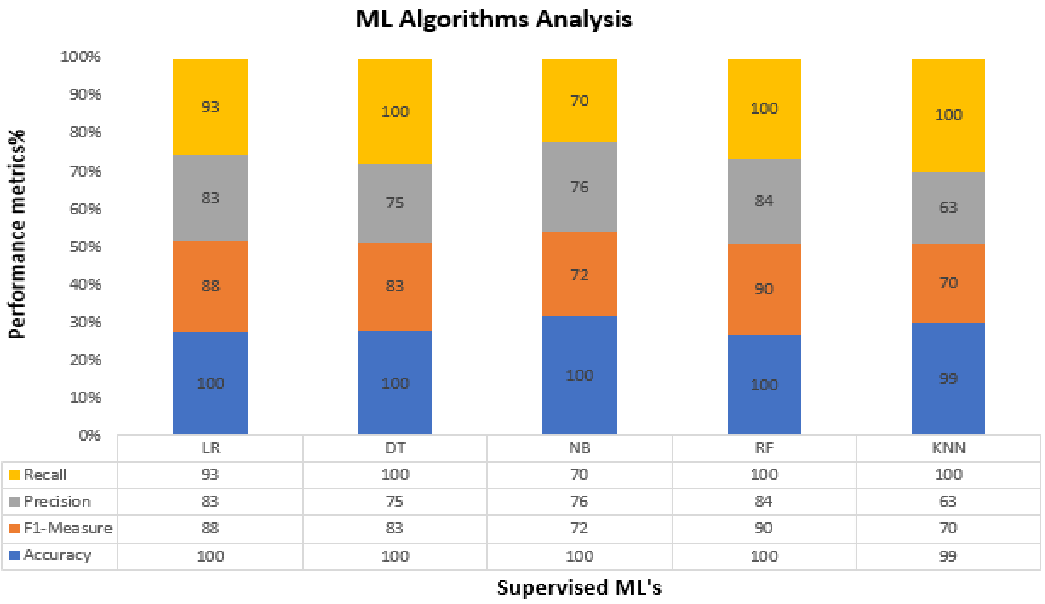
Precision score: This indicates the true positive predictions of the model, correctly identifying the right DNS flooding attacks instead of benign.

Recall score: This is also known as the **Detection score**, and it indicates the predictions of how many DNS flooding attacks correctly identified out of all the DNS flooding attacks.

F1-Measure score: This is a combination of precision and recall score. A perfect model achieves the highest F1 score of at-least 70%-100%.

Accuracy score: This is a metric that describes how well the model performs across all classes.

The figure below displays the findings of the 1st case of the study. We evaluated five supervised ML algorithms (LR, DT, ND, RF, and KNN) using the performance metrics described above.



	LR	DT	NB	RF	KNN
🟨 Recall	93	100	70	100	100
▨ Precision	83	75	76	84	63
🟧 F1-Measure	88	83	72	90	70
🟦 Accuracy	100	100	100	100	99

Figure 4.10 ML Model Findings

Figure 4.10 displays the computed metric results for each ML algorithms. We trained and tested each ML through the use of the collected datasets [6]. Based on our study, the model with the highest accuracy score is more capable of detecting DDoS attacks than the other models. In our findings all the ML models (LR, DT, RF, and NB) achieve the highest accurate score of 100% except KNN which is 99%. Hence, we can conclude that all ML models can detect DNS flooding attacks in MEC environment with precision, except for KNN.

If the ML model has low recall and high precision rate, this implies bias towards DDoS attacks detection and therefore F1-measure must be compared to reach a final decision based on which one is the best model. From this perspective, high F1-measure rate indicates that the model performs better compared to the others.

When comparing the models from best optimal to least optimal model, it is evident that NB achieves a low recall score of 70% with high precision score of 76%. This then implies that NB model is biased in detecting DNS flooding attacks. It can be observed from Figure 4.10 that KNN is not biased in detecting DDoS attacks, but the mere fact that it achieved the lowest F1-measure score of 70% suggests that the model is outperformed by NB.

LR, DT, and RF are not biased in terms of detecting DNS flooding attacks (DDoS attacks). Since LR and DT achieved an accurate score of 100%, the implication is that we have to compare F1-measure score. Hence, LR is the second best since it has the highest F1-score of 88% whereas DT have F1-measure score of 83%.

We can conclude that in the first case of the study RF performed better compared to other models since it achieved the highest recall score of 100%, highest precision score of 84% as well as highest F1-measure score of 100%.

Based on this analysis of our computed findings, we discovered that to check the range of best optimal to least optimal model respectively, we could use F1-measure scores of each ML model.

4.4 The 2nd CASE OF THE STUDY: Hybrid model

As discussed, in Chapter 3, As discussed, in Chapter 3, stack ensembling technique is used to build a new model and improve its performance. The purpose of implementing a hybrid-model in our study was to optimize the accurate detection of DDoS attacks. Hence, the figure below presents the process of implementing hybrid model (Stack model) [42].

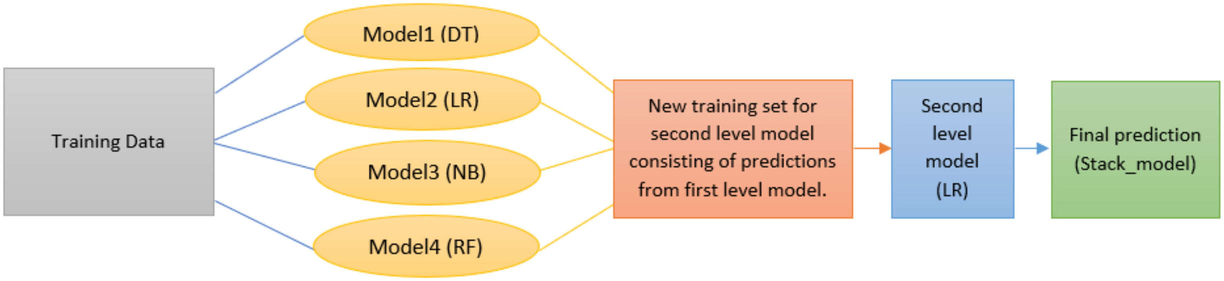
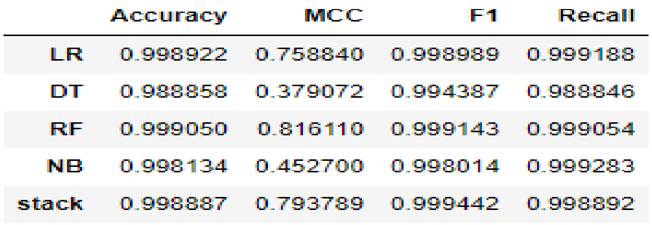

Training Data
Model1 (DT)
Model2 (LR)
Model3 (NB)
Model4 (RF)
New training set for second level model consisting of predictions from first level model.
Second level model (LR)
Final prediction (Stack_model)

Figure 3.6 Stacking Diagram 1

In the process of implementing a hybrid model, we integrated the best four optimal models namely LR, DT, NB, and RF. The experimental findings acquired by training, and testing SL technique on the dataset using performance metrics are detailed in the following graphical representation, Figure 4.11, 4.12, 4.13, 4.14, 4.15, 4.16, 4.17, 4.18, 4.19 and Figure 4.20, as well as Table 4.1.

We used the same metrics that were applied in section 4.3 to check the performance of supervised ML algorithms in detecting DDoS attacks. In addition, the following statistical performance metrics MCC and AUROC were used as they are ideal in dealing with supervised ML models tested or trained on dataset that was imbalanced and got balanced for ease of training and testing SL techniques.

The base models DT, LR, NB, and RF are used to train our target variable in order to learn how to detect the DDoS attacks. LR was used as our meta model (level-1 model) to implement the stacking hybrid model. It is crucial to remember that RF is a combination of various DT’s; hence it is regarded as a hybrid model as well. The findings of the base models (ML models) are called level 0 predictions, whereas the stack model is called level 1 predictions. The findings from our implemented hybrid models (RF, and Stacking hybrid model), and our ML models (DT, LR, and NB) are represented in the table below:



	Accuracy	MCC	F1	Recall
LR	0.998922	0.758840	0.998989	0.999188
DT	0.988858	0.379072	0.994387	0.988846
RF	0.999050	0.816110	0.999143	0.999054
NB	0.998134	0.452700	0.998014	0.999283
stack	0.998887	0.793789	0.999442	0.998892

Table 4.1 Supervised ML findings

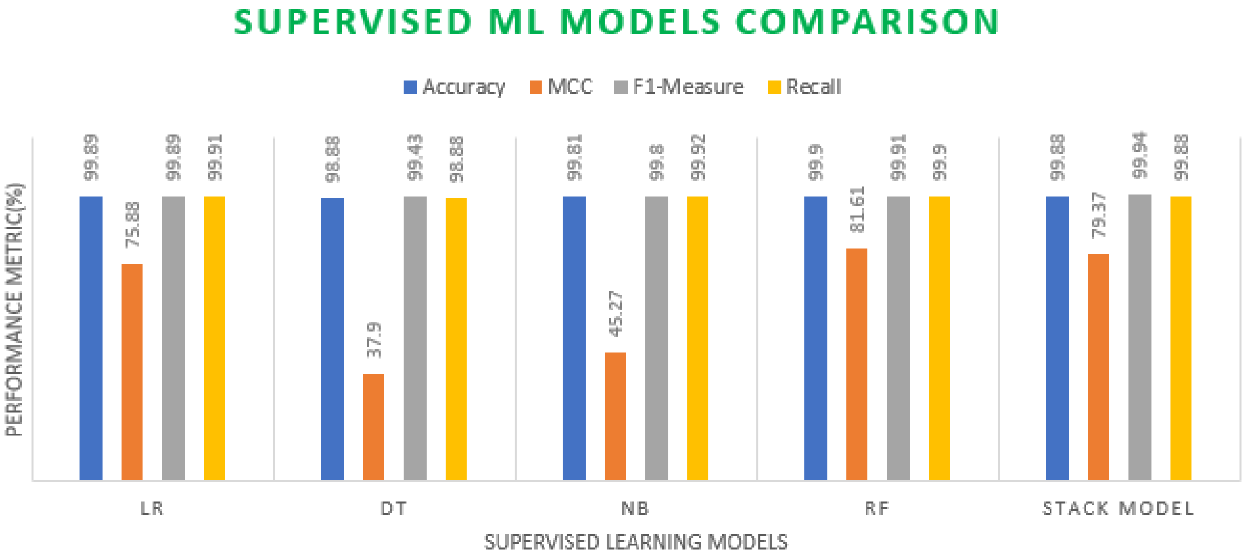


Figure 4.11 Supervised Learning Models

As can be observed in the **Table 4.1**, and Figure **4.11,** both hybrid models and ML models are quite successful in detecting attack traffic of DNS flooding attack in MEC environment. It is also evident that they all achieved almost the same results for computed metrics: accuracy, F1-measure, and recall. Nonetheless, there is a slight difference with F1-measure scores. Analytically, it ca be seen in the 1st case of the study that F1-measure scores can be utilized to range the performance of ML models, respectively. Similarly, in this case of the study the effectiveness of each supervised ML model is measured by using the F1-measure metric.

Hybrid models outperformed the ML models by achieving the highest F1-measure scores compared to ML models. This suggests that they are the best optimal or efficient models in mitigating the DDoS attacks in MEC:

- ✓ **Hybrid models**: 1st - Stack model = **99.94%**, and 2nd - RF = **99.91%**.
- ✓ **ML models**: 3rd - LR = **99.89%**, 4th - NB = **99.80%**, and 5th - DT = **99.43%**

When comparing MCC scores, there is a competition of hybrid models, as RF outperforms stack model by 2.24%; other than that, the range of effective models in detecting DDoS attacks remains the same. According to these findings, we can conclude that hybrid models are most effective in detecting the DDoS attacks, whereas ML models are least effective. We derived the conclusion from evidence on which hybrid model is most accurate by computing the average for each hybrid model using F1-measure, and MCC score. Hence, RF outperformed the Stack model by 1.1% as shown in the figure below:

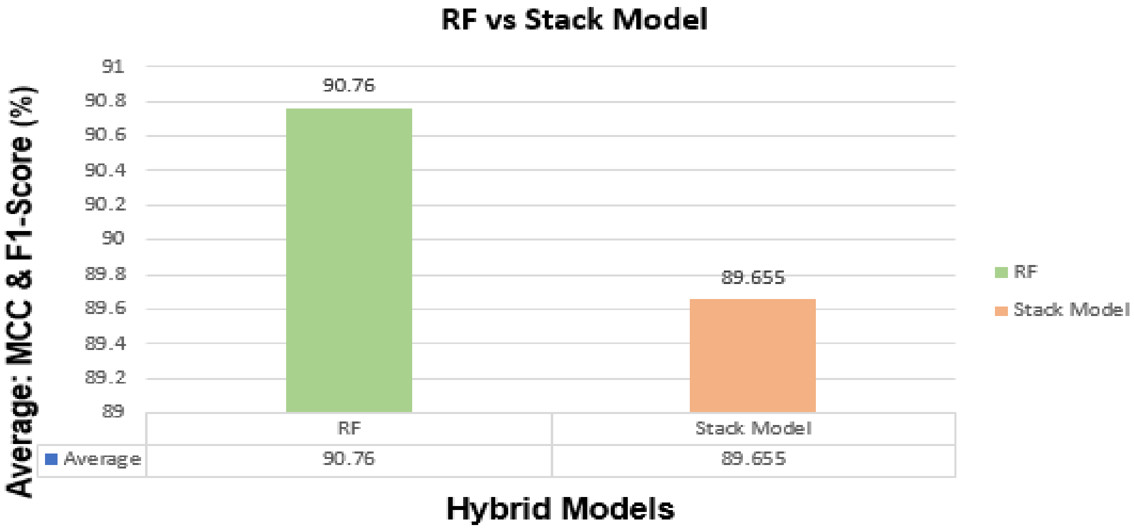


Figure 4.12 Average of Hybrid Models

4.5 ROC Curve for Supervised ML Techniques

The synthetic dataset was generated for the purpose of confusing the supervised ML models by adding noisy features to make the problem more difficult to resolve. We then separated the data into training and testing at the rate of 0.80 and 0.20 respectively. The classification models (DT-ML model, KNN-ML model, RF-hybrid model, Stacking-model-hybrid model) were built. ROC and AUROC score were computed for each classification models for the purpose of evaluating each model. The findings are represented by the plots of ROC curve, as displayed below.

We use AUROC metric to further evaluate each SL technique performance because ROC metric is best suited to evaluate supervised ML models trained and tested on imbalanced dataset [6], and it is also ideal in diagnosing models with unbalanced data. If the AUROC score is greater than or equal to 0.50, then it implies that the model is a good-to-go model, but if the AUROC score is under the random prediction which is always equal to 0.50, then this indicates that the model is not optimal in performance. Hence, the supervised ML model with a ROC curve line lying under the diagonal line reflects that the performance of diagnostic test completely failed. Based on a rough classification [44], in general, AUC is interpreted as follows: 90% -100% = excellent;

80% - 90% = good; 70% - 80% = fair; 60% - 70% = poor; 50% - 60% = fail.

Figure 4.13 demonstrates ML algorithms comparison with the use of performance metric AUROC. The more the AUROC curve line gets closer to the true positive rate or the higher the AUROC score, the more ML algorithm is efficient in dealing with DDoS attacks.

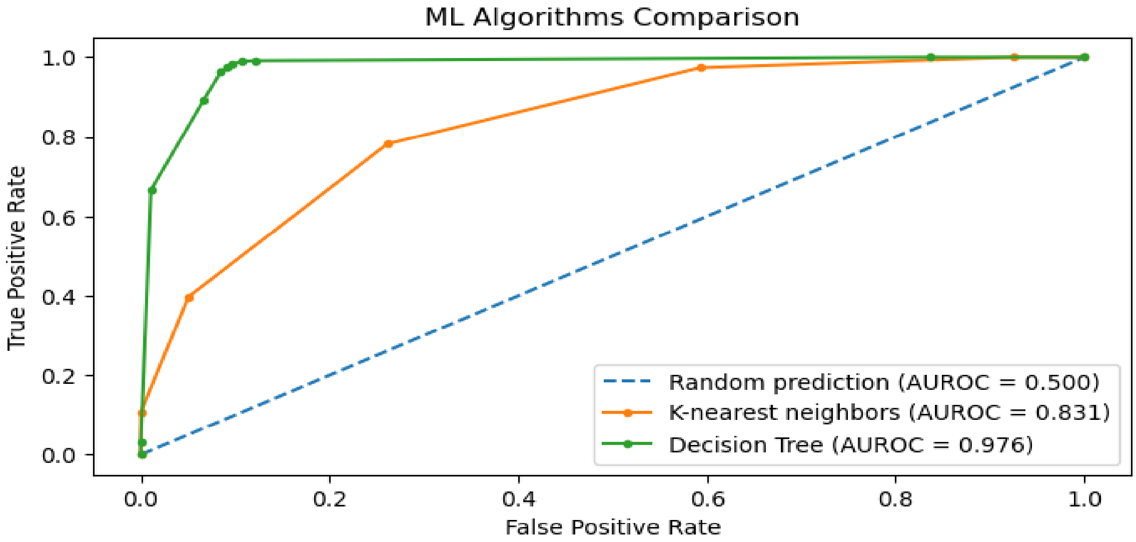


Figure 4.13 AUROC ML Models Comparison

Based on Figure 4.13, ML models are compared wherein DT achieved 97.6% which implies that the model performed well, and KNN achieved 83.10% which implies that the model performance is good. This indicates that DT outperforms KNN by 14.50%.

Figure 4.14 demonstrates Hybrid model comparison with the use of performance metric AUROC. The more AUROC curve line gets closer to the true positive rate or the higher the AUROC score, the more hybrid model is efficient in dealing with DDoS attacks.

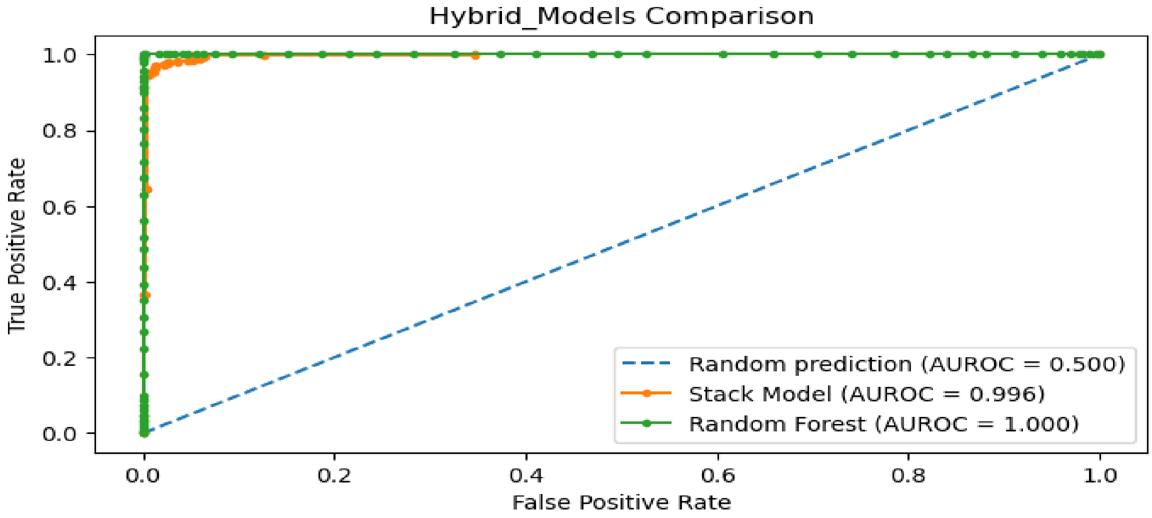


Figure 4.14 AUROC Hybrid Models

Based on Figure 4.14, when comparing the hybrid models, the ROC curve shows that both hybrid models achieved excellent results wherein RF achieved 100%, and Stack model achieved 99.6%. However, RF outperformed the stack model by 0.004%.

Figure 4.15 demonstrates ML algorithms against hybrid model comparison with the use of performance metric AUROC. The more AUROC curve line gets closer to the true positive rate or the higher the AUROC score the more ML algorithm or hybrid model is efficient in dealing with DDoS attacks.

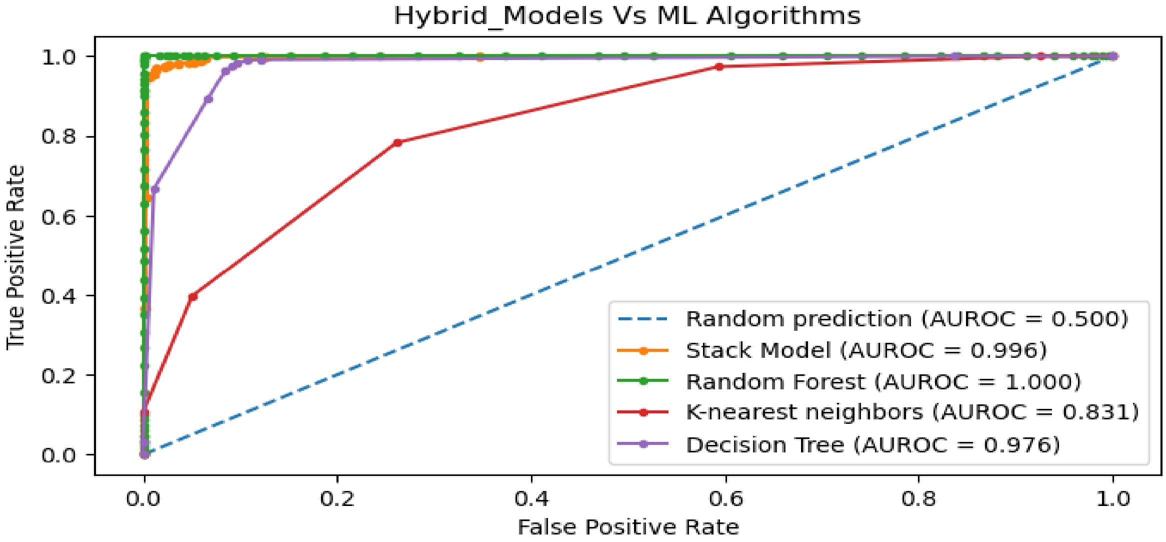


Figure 4.15 AUROC Hybrid vs ML Models

Based on Figure 4.15, when comparing the SL techniques, the evaluated findings show that the hybrid models (RF, and Stack model) outperformed the ML models (DT, and KNN).

4.6 PDF of Normal Distribution for Supervised ML Techniques.

PDF describes probabilities for continuous random variables. A PDF for a normal distribution calculates the probability for a range of values rather than a single value because in continuous random variables, there is no probability of a single value. This implies that the probability of certain single values in a continuous random variable is always equal to zero $(P(X = x) = 0)$.

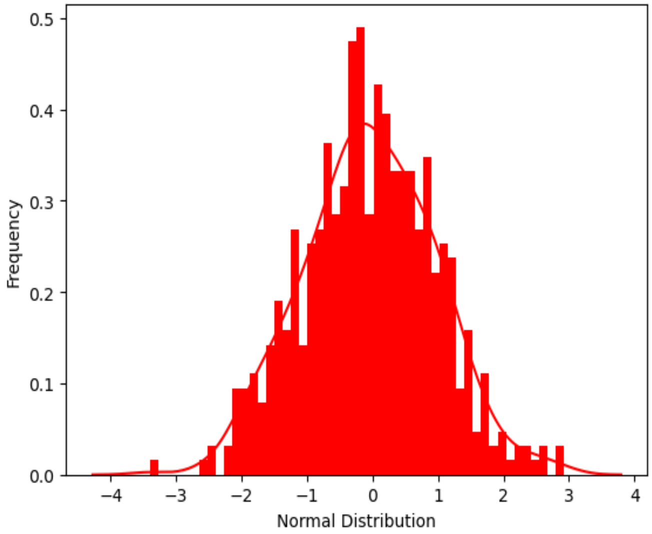


Figure 4.16 PDF of Normal Distribution 1 (RVS=500)

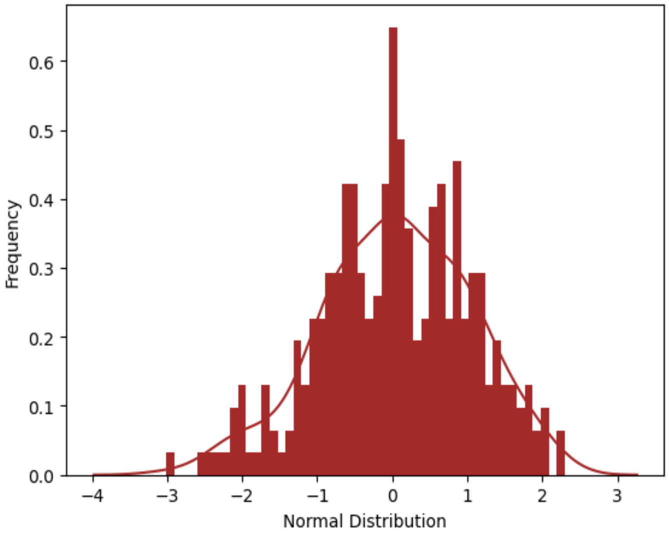


Figure 4.17 PDF of Normal Distribution 1 (RVS=290)

PDF at a given point shows the probability density or value on the y-axis, not the probability at that point. Due to this, the probability of X taking a single discrete value is zero for a continuous distribution.

The figures above show PDF of normal distribution computed based on target variables (Benign, and DNS DDoS attacks), and these PDFs are calculated at different random variable samples (RVS). Both PDFs are normally distributed; they are not skewed either to the left or right since the dataset was balanced by RUS technique. The implication is that the supervised ML techniques were built under fair (or unbiased) inputs. As observed from both graphs, the possible outcomes or the probability of an algorithm to be randomly trained or tested with equal network activities (DNS attacks and benign) inputs are almost balanced.

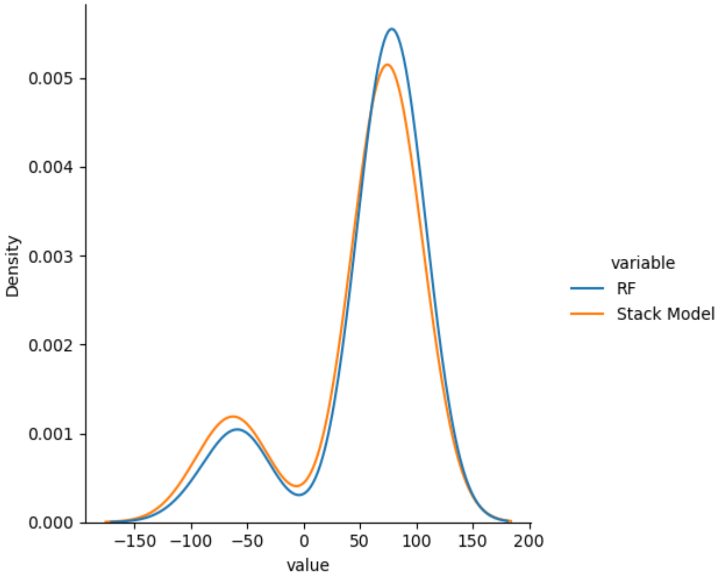


Figure 4.18 PDF of Hybrid Models

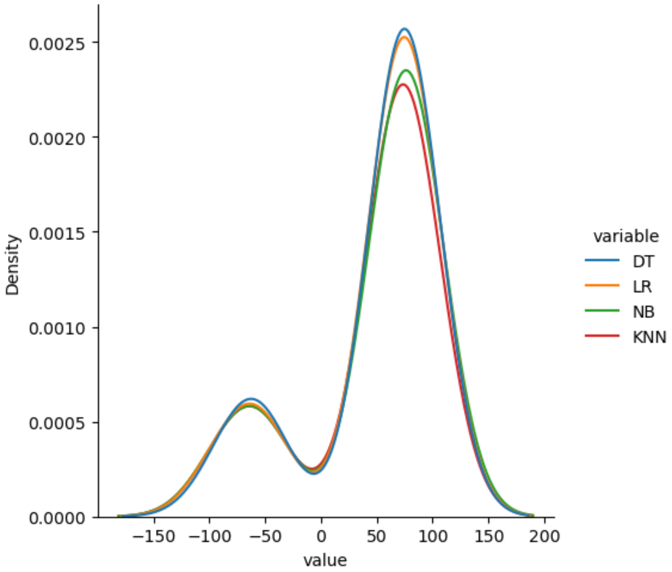


Figure 4.19 PDF of ML Models

PDF of Normal distribution below are based on F1-measure scores for both evaluated supervised ML techniques (Stack model, RF, DT, LR, NB, KNN). F1-measure scores data-frames were created for each supervised ML technique. Based on the computed findings for Figure 4.12, F1-measure score for stack model is 89.66%. However, the generated random variables or the created data-frame did not contain any value exceeding stack model F1-measure score average, similar to all data-frame of each supervised ML techniques.

Figure 4.18 shows the PDF of normal distribution for hybrid models. It should be noted that the generated random variables for computing this PDF was generated based on the 2nd case of the study findings (Figure 4.12). The generated random variables for stack model were close to 89.66%, and random variables for RF were close to F1-score of 90.76%. Hence, we can conclude that the probability detection for stack model is

$(P(X = x) = 0.005)$, and for RF is $(P(X = x) > 0.005)$. As observed from the computed probability detections of RF and Stack model, the more random variables are close to 100%, the more possible outcomes that the model probability detections would be high. The figure for PDF of normal distribution for Hybrid Models also concurs the same.

Figure 4.19 shows the PDF of normal distribution for ML models. It should be noted that the generated random variables for computing this PDF was generated based on the 1st case of the study findings (figure 4.10). An overview of the generated random variables for ML models is presented as follows: Random variables for: LR ≤ 88, DT ≤ 83, NB ≤ 72, and KNN ≤ 70. ML models probability detection based on the PDF are respectively as follows: LR: $(P(X = x) \geq 0.0025\,)$, DT: $(P(X = x) > 0.0025)$, NB: $(P(X = x) \leq 0.0023)$, and KNN: $(P(X = x) > 0.005)$. However, we can conclude based on the PDF findings that the more random variables are close to maximum limit of each ML random variable, the more possible outcomes of the probability detections would be high.

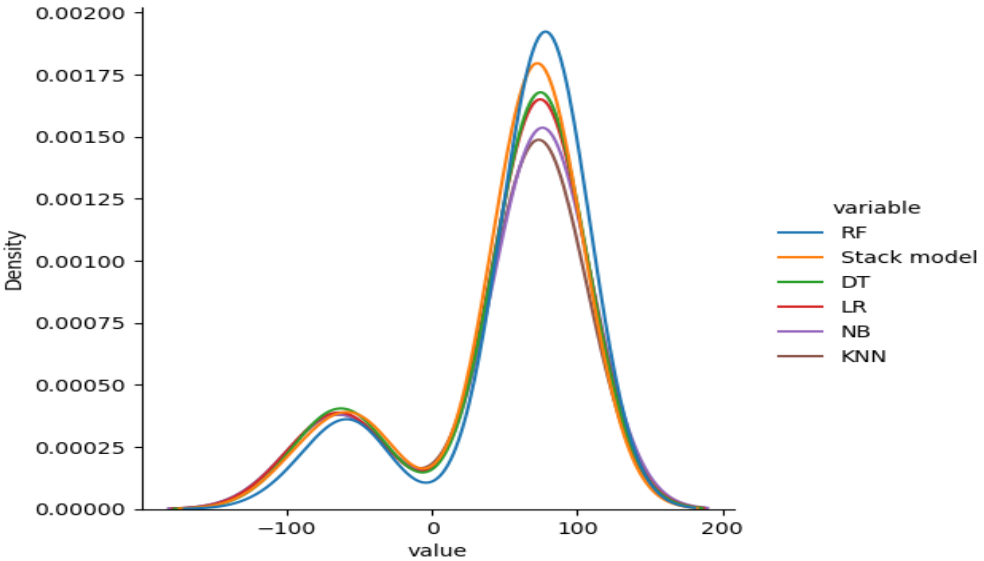


Figure 4.20 PDF of Hybrid Vs ML Models

Figure 4.20 shows the PDF of normal distribution for hybrid against ML models. The above PDF was computed to present the comparison between hybrid and ML models. Based on the evaluated findings of both **Figure 4.18, and Figure 4.19,** we can conclude that the supervised ML technique that reaches high probability detection implies high possible outcomes of effective models in detecting DNS DDoS attack on MEC environment. These PDF of normal distributions findings strongly support our evaluated results from the 1st case of the study, and 2nd case of the study.

The regenerated dataset was complex enough to confuse the trained and tested SMLs. Both SMLs models seems to be precisely performing well in simple dataset. However, with generated dataset containing wide range of patterns and noisy features, ML-based hybrid models seem to suffer highly sensitive nuances present in the dataset which leads to achieving different precise accurate detections results. This can be contested when comparing the huge difference of ML-based hybrid models for PDF of normal distribution (figure 4.18) and AUROC scores (figure 4.14). Therefore, it's recommended to use well clean balanced dataset to both train and test ML-based hybrid models.

4.7 Hypothesis Testing for Conclusion

This study consolidates the evaluated findings with the use of hypothesis testing. We formulated a research hypothesis question based on the obtained results. There are a few principles to follow in hypothesis testing such that, if the sample size is greater than or equal to 30, Z-test statistic (z_t) for population proportion must be used which follows that normal distribution must be used as well. The formulated question is as follows:

Author 1 claims that the ML-based hybrid models (Stack model, and RF) achieve at-least 90% of F1-measure score whereas author 2 claims that Stack model and RF achieve at-most 90% of F1-measure score. 1406 out of 3812 network activities (benign and DNS DDoS attacks) were randomly sampled to test at 95% confidence interval.

Step 1: Our null hypothesis is assumed to be true, until proven contrary.

$\boldsymbol{H_0}$: F1-measure score ≤ 90% (The hybrid models achieve F1-Measure score of less than or equal to 90%).

$\boldsymbol{Ha}$: F1-measure score > 90% (The hybrid models achieve F1-Measure score greater than 90%).

Step 2: Computing significance level.

$$\alpha = 1 - C = 1 - 0.95 = 0.05$$

Step 3: The test statistic for this problem is sample proportion which is the number of network activities that are correctly classified by ML-based hybrid model divided by the total number of network activities sampled. Let P_0 be the true proportion of correctly classified network activities by the ML-based hybrid model.

Computing sample proportion mean:

$$\hat{p} = \frac{X}{n} = \hat{p} = \frac{1406}{3812} = 0.37$$

Computing Z-test statistic (z_t) for population proportion

$$z_t = \frac{\hat{p} - P_0}{\sqrt{\frac{p_0(1 - P_0)}{n}}} = \frac{0.37 - 0.90}{\sqrt{\frac{0.90(1 - 0.90)}{3812}}} = -109.08$$

Step 4: The rejection zone is the range of test statistic values for which we reject the null hypothesis. Given that this is a one-tailed test with a significance level of 0.05, we reject the null hypothesis if the test statistic is smaller than the critical value zα = -1.645. Since our test statistic $Z_t = -109.08$ is smaller than the critical value $z_a = -1.645$, we reject the null hypothesis. Therefore, we have sufficient evidence to support the claim made by Author 1 that the ML-based hybrid models achieve at-least 90% of F1-measure score.

4.8 CONCLUSION

In this chapter, the benign, together with the attack traffic from the dataset [6] were attained. Data was classified using supervised ML techniques (DT, NB, LR, KNN, RF and Stack model). The customised MEC-based dataset contained the benign, and the DDoS attacks (specifically DNS flooding attacks).

The dataset has statistical features such as source IP addresses, destination IP addresses, forwarded packets, backward packets, and Label. The mentioned features define the source and target machines' IP addresses where packets containing either the attack traffic and benign was distributed, the label (our dependent variable) feature entails those benign and DNS flooding attacks which we encoded into categorical values.

To achieve accurate classification, ML models and a hybrid model were utilized. By examining 79 network features, 71 features were identified as effective and used as input for supervised ML algorithms. After pre-processing and feature selection, DT, NB, LR, KNN, RF, and Stack models were employed to classify over a thousand records.

According to the investigated findings of the 1st case of the study, and 2nd case of the study, RF achieved the best optimal accurate rate in dealing with DDoS attacks compared to all supervised ML models (Stack model, DT, NB, LR, and KNN). When comparing the SL techniques, the hybrid models (RF, and Stack model) outperformed ML models (DT, NB, KNN, and LR). The ultimate and final chapter deals with recommendations and provides conclusions derived from the study.

CHAPTER 5 SUMMARY, RECOMMENDATIONS & CONCLUSION

5.1 INTRODUCTION

A summary of our findings is presented in this chapter. We discuss the conclusion, recommendations, and contribution, as well as the limitations of the study. This entails the conclusions drawn from the assessed outcomes of implementing ML models, including the hybrid model. Lastly, the chapter summarises our concluding remarks.

5.2 SUMMARY OF THE RESEARCH FINDINGS

The findings are categorized in two ways, namely: the 1st case and 2nd case of the study. Our findings are based on ML models and hybrid models implemented to detect DDoS (DNS flooding) attacks on the MEC network environment.

We proposed ML algorithms to detect DDoS attacks based on the existing dataset from the New Brunswick Canadian Institute for Cybersecurity. For evaluating our results, the computed performance metrics for each ML algorithm (LR, RF, DT, NB, and KNN) were utilized. Each ML model is compared based on each metric score, and thereafter the performance of each ML model is ranked from the most optimal to the least optimal, respectively.

Based on the evaluated findings we confirmed that the higher the F1-measure score the better the detection rate. NB was verified to be biased in detecting the DDoS attacks, whereas KNN was not biased but achieved the lowest F1-measure score of 70% which makes it outperformed by NB. In tandem, LR, DT, and RF were both found not to be biased, and they are ranked based on their F1-measure scores from most effective to least effective (RF, LR, DT) respectively. Based on the overall analysis of accurate score results, it can be concluded that all ML models successfully identified the attack traffic associated with DNS flooding attacks in MEC environments.

We implemented hybrid models (stack model, and RF). For implementing the stack model, we used DT, LR, NB, and RF, whereas RF uses various DT's. Performance measures such as accuracy, recall, MCC, and F1-measure score were calculated for each model to assess and compare the performance of hybrid and ML models.

The accuracy and recall scores for all supervised ML models (DT, LR, NB, stack model and RF) were greater than 90%. As experimentally proven from the 1st case of the study, the higher the F1-measure score, the better the detection rate. Supervised ML models F1-measure scores performances ranged from most effective to least effective respectively as follows: **Hybrid models**: 1st - Stack model = **99.94%**, and 2nd - RF = **99.91%**. **ML models**: 3rd - LR = **99.89%**, 4th - NB = **99.80%**, and 5th - DT = **99.43%**

Based on this F1-measure, hybrid models outperformed the ML models. When comparing MCC scores, the supervised ML models are respectively ranked from most effective to least effective as follows: **Hybrid models**: 1st - RF = **81.61%**, and 2nd – Stack model = **79.37%**. **ML models**: 3rd - LR = **7.88%**, 4th - NB = **45.27%**, and 5th - DT = **37.9**

Based on MCC scores, hybrid models still outperformed ML models. Based on F1-measure scores and MCC scores, both ML models can still be ranked in the same order. However, when comparing the F1-measure, the stack model outperformed RF but the converse was true when comparing the MCC scores. Hence, to find the most effective model between hybrid models, we computed the average based on MCC, and F1-measure scores where RF outperformed the stack model by 1.1%.

For computing the AUROC score, we generated a new dataset using a synthetic technique. Both supervised ML achieved an AUROC score greater than the random prediction score, implying that both models are good-to-go models. We implemented another stack model using DT, RF and KNN. ML algorithms were compared where DT achieved 97.6% and outperformed KNN by 14.5%. Hybrid models were compared where RF achieved 100% and outperformed the stack model by 0.4%. Lastly, hybrid models and ML algorithms were also compared where Hybrid models outperformed ML algorithms.

5.3 SUMMARY OF PDF & HYPOTHESIS TESTING.

We employed statistical methods such as the normal distribution's probability density function (PDF) and hypothesis testing to effectively substantiate our findings in both the first and second cases of the study. Based on the evaluation of both PDFs and hypothesis tests, our conclusions were evidently supported.

5.4 CONCLUSION

Our study demonstrated that supervised ML techniques (LR, NB, KNN, DT, RF, and Stack model) are successful in dealing with DDoS attacks. The experimental results show that hybrid models are significantly accurate as compared to ML algorithms. Even in a situation where models are confused by adding noisy features to make the problem more difficult to deal with, hybrid models are still optima in dealing with DDoS attacks. Based on the overall findings, RF generates the most reliable and accurate results.

Based on the evaluated results of both the 1st case and 2nd cases of the study, the supervised ML techniques were able to detect the attack traffic of DNS flooding attacks in MEC. However, Hybrid models outperformed ML models in all cases. Lastly, based on the overall findings, RF is the most effective SL technique in dealing with DDoS attacks. When comparing ML models, LR outperforms other ML models (NB, KNN, and DT).

5.5 RECOMMENDATIONS AND LIMITATIONS

Our study was motivated to generate scientific knowledge that ensures that hackers are unable to create traffic to the internet servers, preventing edge users from utilising cellular networks or the Internet. Our study focused on supervised ML techniques (hybrid models, and ML models). Future work could examine security complexities and compromises using various ML such as semi-supervised, unsupervised, and reinforcement learning to confirm and further investigate the utilization of diverse traffic datasets and assess the effectiveness of our proposed ML model on additional datasets.

The limitations of the suggested system involve the utilization of supervised ML techniques only, as well as the use of a single dataset. A model that uses embeddings rather than a pre-trained stack could be an alternative approach. In this study, we classified input traffic into normal (benign) and DNS flooding attack categories using a binary classification system.

5.6 FINAL CONCLUSION

DDoS attacks pose significant complications and difficulties for many aspects of our lives such as the MEC environment. Thus, there was a need to develop a comprehensive intrusion detection system in order to reduce the number of attacks such as DNS flooding attacks. Without proper handling, these attacks can cause complete disruption, as they become more complex and can bypass many traditional protection methods. In this study, supervised ML techniques are implemented in MEC to address network security concerns.

Our study examined six supervised ML algorithms: LR, DT, NB, KNN, RF, and the Stack model. Several measurements were used in the evaluation, including accuracy, precision, recall, MCC, AUROC, true-positive rate, false-positive rate, and F1-measure. When comparing ML models in all cases of the study, LR outperforms other algorithms (NB, DT, and KNN). Based on the evaluated findings, hybrid models outperform ML models. According to the experiment, the hybrid (Stack and RF) model has the best accuracy score of 99.88% and 99.90%, respectively.

Based on the 2nd case of the study there was competition and optimism between the hybrid model where the Stack model achieved the highest F1 measure score of 99.94 whereas RF achieved the highest MCC score of 81.61%. Hence, the average based on MCC, and F1-measure scores were calculated, where RF outperformed the Stack model by 1.1%. In that case, RF was confirmed as the most optimal or effective SL technique. The results show that ML models are quite effective at detecting DDoS attack traffic. The goal of our study was to contribute to interdisciplinary research in IT. This study's application can be used in our real-life actual systems across a variety of IoT sectors.

REFERENCES

[1] D. M. P. Bangalore, “LANDSCAPE ON 5G TECHNOLOGY ATechnology AND SERVICE PROVIDERS PERSPECTIVE”.

[2] H. Rutvij, J. P. Jhaveri and C. J. Devesh, “DoS Attacks in Mobile Ad Hoc Networks: A Survey,” pp. pp. 535-541, 7 Januaey 2012.

[3] Osanaiye and A. Opeyemi, “DDoS defence for service availability in cloud computing,” 2016.

[4] C. Lai, L. Rongxing, Z. Dong and S. Xuemin, “Security and privacy challenges in 5G-enabled vehicular networks,” *IEEE Network, 34(2),* pp. pp.37-45.

[5] C. I. f. Cybersecurity, “DDoS Evaluation Dataset (CIC-DDoS2019),” 2019. [Online].

[6] Canadian Institute for Cybersecurity, “DDoS Evaluation Dataset (CIC-DDoS2019),” 2019.

[7] K. Bhardwaj, J. C. Miranda and G. Ada, “Towards {IoT-DDoS} Prevention Using Edge Computing,” *In USENIX Workshop on Hot Topics in Edge Computing (HotEdge 18),* 2018.

[8] X. Xu , S. Yongqiang and H. Zunguo, “Defending DDoS attacks using hidden Markov models and cooperative reinforcement learning,” *In Pacific-Asia Workshop on Intelligence and Security Informatics,* pp. pp.196-207, 2007.

[9] N. Bindra and S. Manu, “Detecting DDoS attacks using machine learning techniques and contemporary intrusion detection dataset,” *Automatic Control and Computer Sciences 53, no. 5 ,* pp. pp.419-428, 2019.

[10] Mamolar, S. Ana, P. Zeeshan, M. Jose , C. Alcaraz and M. K. Asad, “Towards the transversal detection of DDoS network attacks in 5G multi-tenant overlay networks,” *Computers & Security,* vol. 79, pp. pp.132-147, 2018.

[11] Mishra, Anupama, B. Brij, Gupta, P. Dragan, J. G. P. Francisco and H. Ching-Hsien, “Classification Based Machine Learning for Detection of DDoS attack in Cloud Computing.,” in *In 2021 IEEE International Conference on Consumer Electronics (ICCE), pp. 1-4. IEEE*, 2021 January.

[12] D. Chaudhary, B. Kriti , Brij B and Gupta, “Survey on DDoS attacks and defense mechanisms in cloud and fog computing,” *International Journal of E-Services and Mobile Applications (IJESMA), 10(3),* pp. pp.61-83, 2018.

[13] An, Xingshuo, S. Jingtao, L. Xing and L. Fuhong, “Hypergraph clustering model-based association analysis of DDOS attacks in fog computing intrusion detection system,” *EURASIP Journal on Wireless Communications and Networking,* pp. pp.1-9, 2018(1).

[14] A. Agarwal, K. Manju and S. Rajiv , “Detection of DDOS attack using deep learning model in cloud storage application.,” *Wireless Personal Communications,* pp. pp.1-21, 2021.

[15] Singh, Shivangi, K. Khushboo, G. Shashank, D. Amit and K. Neeraj, “Detecting Different Attack Instances of DDoS Vulnerabilities on Edge Network of Fog Computing using Gaussian Naive Bayesian Classifier,” in *In 2020 IEEE International Conference on Communications Workshops (ICC Workshops), pp. 1-6. IEEE*, 2020.

[16] Zhao, Yingqi, L. Yajie, L. Jun, L. Mingzhe, N. Yifan, Z. Yongli and Z. Jie, “Traffic Scheduling Strategy for Mitigating DDoS Attack in Edge Computing-enabled TWDM-PON.,” in *In 2020 Opto-Electronics and Communications Conference (OECC) (pp. 1-4). IEEE*, 2020.

[17] Liu, Jianhua, W. Xin, S. Shigen, Y. Guangxue, Y. Shui and L. Minglu, “A Bayesian Q-Learning Game for Dependable Task Offloading Against DDoS Attacks in Sensor Edge Cloud,” *IEEE Internet of Things Journal,* vol. 8, no. 9, pp. pp.7546-7561, 2020.

[18] Xiao, Liang, W. Xiaoyue , D. Canhuang, D. Xiaojiang, C. Xiang and G. Mohsen, “Security in mobile edge caching with reinforcement learning,” *IEEE Wireless Communications,* vol. 25, no. 3, pp. pp.116-122, 2018.

[19] Bi, Xiaoming, T. Wenan and X. Ruohui, “A DDoS-oriented Distributed defense framework based on edge router feedbacks in Autonomous Systems,” *International Multi-symposiums on Computer and Computational Sciences,* pp. pp. 132-135, 18 October 2018.

[20] Dao, Nhu-Ngoc, V. Duc-Nghia, L. Yunseong, P. Minho and C. Sungrae, “MAEC-X: DDoS prevention leveraging multi-access edge computing,” in *International Conference on Information Networking (ICOIN) (pp. 245-248). IEEE*, 2018.

[21] Lawal, A. Muhammad, A. S. Riaz and R. H. Syed, “ A DDoS Attack Mitigation Framework for IoT Networks using Fog Computing,” *Procedia Computer Science,* vol. 182, pp. pp.13-20, 2021 Jan 1.

[22] Bhushan and Kriti, “DDoS attack defense framework for cloud using fog computing,” in *IEEE international conference on recent trends in electronics, information & communication technology (RTEICT) (pp. 534-538). IEEE.*, 2017.

[23] Ahmad and Iftakhar, “A Survey on DDoS Attacks in Edge Servers (Doctoral dissertation, The University of Texas at Arlington),” 2020.

[24] He, Qiang, W. Cheng, C. Guangming, L. Bo, Z. Rui, Z. Qingguo, X. Yang, J. Hai and Y. Yun, “A game-theoretical approach for mitigatingedge ddos attack.,” *IEEE Transactions on Dependable and Secure Computing.,* 2021 Jan 29.

[25] Dao, Nhu-Ngoc, V. Trung , S. Umar, K. Joongheon, B. Thomas, D. Din-Thaun and C. Sungrae, “Securing heterogeneous iot with intelligent ddos attack behavior learning,” *IEEE Systems Journal ,* 2021 Jun 10.

[26] Mamolar, S. Ana, S. Pablo, C.-P. Enrique, P. Zeeshan, M. C. Jose and W. Qi, “Autonomic protection of multi-tenant 5G mobile networks against UDP flooding DDoS attacks.,” *Journal of Network and Computer Applications,* vol. 145, p. p.102416, 2019 Nov 1.

[27] Aryal, Bhulok, A. Robert, B. Lain and Collings, “SDN Enabled DDoS Attack Detection and Mitigation for 5G networks,” *Journal of Communications, 16(7),* 2021.

[28] M. Aamir and M. A. Z. Syed, “Clustering based semi-supervised machine learning for DDoS attack classification,” *Journal of King Saud University-Computer and Information Sciences 33(4),* pp. pp.436-446, 2021.

[29] H. Polat, P. Onur and C. Aydin, “Detecting DDoS attacks in software-defined networks through feature selection methods and machine learning models,” *Sustainability, 12(3),* p. p.1035.

[30] M. Aamir and M. A. Z. Syed, “DDoS attack detection with feature engineering and machine learning: the framework and performance evaluation,” *International Journal of Information Security, 18(6),* pp. pp.761-785.

[31] D. Banitalebi, Afsaneh, S. MohammadReza and Z. B. Farsad , “The DDoS attacks detection through machine learning and statistical methods in SDN,” *The Journal of Supercomputing, 77(3),* pp. pp.2383-2415, 2021.

[32] A. E. Cil, Y. Kazim and B. Ali, “Detection of DDoS attacks with feed forward based deep neural network model.,” *Expert Systems with Applications 169,* p. p.114520, 2021.

[33] M. E. Ahmed, K. Hyoungshick and P. Moosung, “Mitigating DNS query-based DDoS attacks with machine learning on software-defined networking,” *In MILCOM 2017-2017 IEEE Military Communications Conference (MILCOM),* pp. pp. 11-16, 2017.

[34] KASIM and Ömer, " A Robust DNS Flood Attack Detection with a Hybrid Deeper Learning Model," *Computers and Electrical Engineering,* vol. 100, p. p.107883, 2022.

[35] Q. Li, M. Linhai, Z. Yuan and Y. Jinyao, "DDoS attacks detection using machine learning algorithms," *In International Forum on Digital TV and Wireless Multimedia Communications,* no. Springer, Singapore, pp. pp. 205-216, September 2018.

[36] Ahuja, Nisha, S. Gaurav, M. Debajyoti and K. Neeraj, "Automated DDOS attack detection in software defined networking," *Journal of Network and Computer Applications 187,* p. p.103108, 2021.

[37] Alghazzawi, Daniyal, B. Omaimah, U. Hayat and Z. A. Muhammad , "Efficient detection of DDoS attacks using a hybrid deep learning model with improved feature selection," *Applied Sciences 11,* vol. No. 24, p. p.11634, 2021.

[38] L. Chen, Z. Yuedong, Z. Qi, G. Guanggang and Y. ZhiWei, "Detection of dns ddos attacks with random forest algorithm on spark," *Procedia computer science,* vol. 134, pp. pp. 310-315, 2018.

[39] Wong, FuiFui and X. T. Cheng, "A survey of trends in massive DDoS attacks and cloud-based mitigations," *International Journal of Network Security & Its Applications,* vol. 6(3), p. p.57, 2014.

[40] Xu, Rui, M. Wen-li and Z. Wen-ling, "Defending against UDP flooding by negative selection algorithm based on eigenvalue sets," *In 2009 Fifth International Conference on Information Assurance and Security,* vol. Vol. 2, no. IEEE, pp. pp. 342 - 345, 2009.

[41] D. A. Varma, A. Ravi , S. S. V. Venkata , B. Venkatesh and R. Kannadasan, "Detection of DDoS attacks using machine learning techniques: a hybrid approach," *In ICT Systems and Sustainability: Proceedings of ICT4SD,* vol. Volume 1, no. Springer Singapore, pp. 439-446, 2020.

[42] Wilson, H. Vincent, P. N. Arun, S. Aswin, K. Sushant and R. A. John , "Ranking of supplier performance using machine learning algorithm of random forest," *International Journal of Advanced Research in Engineering and Technology (IJARET),* p. 11(5), 2020.

[43] Wikipedia contributors, "Random forest," *In Wikipedia, The Free Encyclopedia,* p. from https://en.wikipedia.org/w/index.php?title=Random_forest&oldid=1134454815, 18 January 2023.

[44] Safari, Saeed, N. Ahmed, B. Alireza and E. Mohamed, “Evidence based emergency medicine; part 5 receiver operating curve and area under the curve,” *Emergency 4(2),* p. 111, 2016.

[45] Jia, Weijia and Z. Wanlei, “Distributed network systems: from concepts to implementations,” vol. (Vol. 15), 2004.

[46] Ye, Jin, C. Xiangyang, Z. Jian, F. Luting and S. Ling, “A DDoS attack detection method based on SVM in software defined network,” in *Security and Communication Networks*, 2018.

[47] Paharia, Bhumika and B. Kriti, “DDoS Detection and Mitigation in cloud via FogFiter: a defence mechanism,” in *In 2018 9th international conference on computing, communication and networking technologies (ICCCNT) (pp. 1-7). IEEE.*, 2018.

[48] Osanaiye and A. Opeyemi, “DDoS defence for service availability in cloud computing,” 2016.

[49] Lai, Chengzhe, L. Rongxing, Z. Dong and S. Xuemin, “Security and privacy challenges in 5G-enabled vehicular networks,” *IEEE Network, 34(2),* pp. pp..37-45.

[50] H. Li, Y. Chang, W. Liming, A. Nirwan, T. Ding, H. Xuenqing, X. Zhen and H. Dan, “A Cooperative Defense Framework against Application-level DDoS Attacks on Mobile Edge Computing Services,” *IEEE Transactions on Mobile Computing,* 03 June 2021.

[51] C. Lai, L. Rongxing, Z. Dong and S. Xuemin, “Security and privacy challenges in 5G-enabled vehicular networks,” *IEEE Network,* vol. 34(2), pp. pp.37-45, 2020.

[52] D. Chicco and J. Giuseppe, “The Matthews correlation coefficient (MCC) should replace the ROC AUC as the standard metric for assessing binary classification,” *BioData Mining,* vol. 16(1), pp. pp. 1-23, 2023.