Analyzing Security Threats And Mitigation Techniques For Fog Computing In IoT-Enabled Smart Cities

by

Sheikh Sadi Emon
20301349
Marium Malek
20301151
Md. Iftekhar Hossain Turja
24341087
Khalid Redwan Sun
20301281
Md. Atiqur Rahman
20301406

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
June 2024

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Sheikh Sadi Emon
20301349

Marium Malek
20301151

Md. Iftekhar Hossain Turja
24341087

Khalid Redwan Sun
20301281

Md. Atiqur Rahman
20301406

Approval

The thesis/project titled "Analysis On The Security Threats Of Implementing Fog Computing In IoT Based Smart Cities And Their Mitigation Strategies." submitted by

1. Sheikh Sadi Emon (20301349)
2. Marium Malek (20301151)
3. Md. Iftekhar Hossain Turja (24341087)
4. Khalid Redwan Sun (20301281)
5. Md. Atiqur Rahman (20301406)

Of Spring, 2024 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on June 5, 2024.

Examining Committee:

Supervisor:
(Member)

Dr. Muhammad Iqbal Hossain
Associate Professor
Department of Computer Science and Engineering
BRAC University

Co-Supervisor:
(Member)

Dr. Jannatun Noor Mukta
Assistant Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Dr. Sadia Hamid Kazi
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

In the midst of rising urbanization and advancements in technology, Smart Cities have developed as a crucial response to the complexities associated with urban living. A significant contribution to the development of automated smart cities has come from the fusion between IoT fog-based technologies. This integration facilitates instantaneous data processing, optimization of resources, and improved citizen services. Nevertheless, as these systems get more complex and interconnected, they become subject to many security threats. This thesis examines the major security issues introduced by Fog Computing in Smart Cities. Fog Computing is an extension of cloud capabilities to the network infrastructure edge, resulting in enhanced data privacy, decreases latency, and brings new risks that will require additional investigation and mitigation. This study begins by providing a thorough explanation of the fundamental principles underlying Smart Cities, the use of IoT and Fog Computing in building said smart cities, thereby facilitating a comprehensive comprehension of their interconnectedness. Subsequently, this paper investigates the security vulnerabilities that threatens the use of Fog Computing with IoT based Smart Cities, with a particular motif on the potential vulnerabilities pertaining to infringements of data privacy, unauthorized access, network congestion, and associated apprehensions such as Distributed Denial of Service (DDoS) attacks, the dissemination of malware, and physical manipulation. These strategies encompass resilient authentication mechanisms, encryption protocols, intrusion detection systems, and blockchain technology. Finally, this thesis analyzes Fog Computing security vulnerabilities in IoT-based Smart Cities and emphasizes the solutions for proactive approaches to protect these revolutionary urban ecosystems' integrity, privacy, and resilience.

Keywords: Fog Computing, IoT, Smart Cities, Cloud, Network edge, Data Privacy, Security Vulnerabilities

Table of Contents

List of Figures

Chapter 1

Introduction

The combined application of the Internet of Things (IoT) and fog computing technologies has significantly contributed to the advancement of the Smart Cities idea. Fog Computing reduces latency and enhances data privacy by extending cloud capabilities to the network edge. In this particular instance, fog computing is implemented to facilitate rapid data processing due to the inherent limitations in processing capabilities of IoT devices. The core of the research issue revolves around the potential for data privacy breaches, wherein sensitive information becomes vulnerable to unauthorized access and manipulation throughout its transmission between edge devices, Fog nodes, and central cloud servers. When Internet of Things (IoT) devices are integrated into Smart Cities, a broad variety of security holes are created. Hackers might exploit the security holes to have unauthorized access to sensitive information which will disrupt essential services such as public safety, healthcare, transportation as well as overall quality of life in Smart Cities. Investigating the application of Fog Computing technologies in IoT-based Smart Cities is a multifaceted and diverse task. This study primarily focuses on examining the notable security risks detection and mitigation while incorporating Fog Computing into the complicated structure of Smart Cities. In addition to this, it is necessary to have a detailed grasp of the consequences that arise as a result of these security flaws.

1.1 Research Problem

When integrating Fog Computing technology within the framework of Internet of Things (IoT)-based Smart Cities, the first step in the investigation of the study comprises completing a comprehensive evaluation of the complex security risks that develop as a result of this integration. Some of the vulnerabilities such as data breaches, unauthorized access risks in network systems, and the possibility of physical tampering. These are some of the concerns raised. These vulnerabilities must be identified and investigated to fully understand the research subject.
In addition, the current research investigates the existing network architecture of fog computing as well as the vulnerabilities that exist inside the distributed network environments. Besides, malevolent entities by the hacker on the vulnerabilities might cause the disruption of service with manipulated data. The security of communication channels and the integrity of data at each processing point would face challenges due to the distributed nature of Fog Computing and those vulnerabilities must be analyzed and solved with efficient security strategies.

Within the physical security framework, the investigation is being continued on the substantial security challenges surrounding the possibility of tampering or illegal access to fog nodes and edge devices that are located within smart cities. So, it becomes necessary to secure both the physical infrastructure and digital resources for the overall operations of smart city systems. However, to ensure the integrity of both components, it brings out the complexity of this research topic which explores innovative strategies to fulfill both cyber and physical infrastructure security demands.
The study does not only work on the implications and identifications but also incorporates the problems and issues [4], while combining IoT with fog in smart cities. Unauthorized access to computer systems can reveal the private information of the people of smart cities which would cause the invasion of privacy with significant financial damages. The unauthorized access to the system would interrupt beneficial services which could put people's safety, health, transportation, and overall livelihood at risk. The research topic is further emphasized by considering the economic consequences of security breaches, which encompass the costs associated with recovery efforts to ensure safety.
The research subject at hand necessitates a comprehensive comprehension of the difficulties involved, while also highlighting the pressing need for proactive actions to safeguard and fortify Smart Cities in the age of technological advancement. In summary, our research aims to uncover the hidden dangers that the IoT devices may face in smart cities, particularly when fog computing is also involved.

1.2 Research Objective

The primary purpose of this research is to investigate and develop solutions for the security concerns raised by the application of fog computing along with the Internet of Things enabled in automated smart cities. This study aims to investigate the objectives within the context of the dynamic urban landscape, where the use of IoT and fog-based technologies holds significant importance. The objectives are:

1. To analyze Fog Computing's fundamentals and its incorporation into IoT-based Smart Cities.

2. To identify and analyze the security risks that come from Fog Computing in urban areas.

3. To Review security steps like authentication, encryption, and intrusion detection to mitigate such threats.

4. To evaluate the effectiveness of proposed security solutions.

5. To give practical suggestions for improving the proposed solution to the security of IoT-based Smart Cities.

1.3 Research Contribution

Our thesis facilitates the following contributions:

- We analyze critical factors affecting the security of Fog computing environments in IoT-enabled smart cities, identifying various cyber threats and their detrimental impacts on security measures.
- We develop a novel detection system along with comprehensive mitigation strategies. This system is capable of identifying different types of malicious attacks in real time and instantly mitigating them by dropping the malicious packets.
- We create a new dataset for our model by extracting identical features from a generated dataset based on the flow of simulation, similar to the CICIOT2023 dataset. This facilitates effective training and testing of our detection system.
- We design an innovative algorithm that can detect multiple types of attacks in real time. The algorithm incorporates an instant mitigation strategy by dropping malicious packets to prevent further damage.
- To implement our framework in a practical Fog computing environment, we develop a virtual architecture using Mininet and integrate it with the RYU controller. This setup allows us to simulate and evaluate the performance of our detection and mitigation strategies in a controlled environment.

1.4 Thesis Structure

The thesis is being outlined for the enhancement of security measures in smart cities through employing fog computing along with IoT technology. To begin with, chapter 1 portrays the introduction of the research depicting the necessity of highlighting the security threats in the smart city environment. Afterward, chapter 2 represents the literature review and explores the basic concepts of IoT, fog computing, and their use cases in smart cities. It also provides an Idea of existing security threats and their potential mitigation techniques. Furthermore, chapter 3, illustrates the proposed model along with the analysis, identification, and mitigation of security threats in addition to data collection, preprocessing, and decision-making based on the accuracy of the model grounded on developed algorithm.Moreover, Chapter 4 shows the details of the experimental setup along with the configuration of the Mininet network emulator and Ryu controller for analyzing, monitoring, and mitigating the security vulnerabilities against suspicious attacks and it also highlights experimental results and findings comprising on 34 attack classes.Then, chapter 5 demonstrates CIA Maintenance of proposed model and make a comparison with existing studies as well as states the limitation. Finally, Chapter 6 concludes the thesis with an overall summarization of the findings and future research opportunities emphasizing the importance of the integrity and privacy of IoT-enabled smart cities against rising security threats.

Chapter 2

Literature Review

Fog computing works like a media or filter between cloud and IoT devices. According to [46], "A part of cloud computing, fog computing introduces data storage, calculations, applications, and data analysis closer to IoT devices." Another study [4] found that, "The major capabilities in the framework of fog computing include working close to the local nodes. The necessity of data transportation to cloud servers has significantly diminished due to the prevalence of local computation for most tasks. According to the study conducted by [46], "it is necessary to consider several variables such as confidential information, cloud reachability, its application latency, data preservation, transfer, and credibility in communication while dealing with fog computing in this domain." As found in [46], "in a range of applications, fog computing gives up previously unthinkable possibilities, such as real-time navigation, e-health services, manufacturing, and critical infrastructure control." Fog computing minimizes the need to send almost all the information to a single, centralized cloud-based server by allowing real-time handling and evaluation of this information at the edge according to [41] and this might result in more rapid choices and more rapid reactions. According to [46], there are several network and security concerns with fog computing, including "data processing, resource constraints, trust, authentication, privacy, and many more."

2.1 Background

2.1.1 IoT Devices:

The IoT (Internet of Things) enables widespread device connectivity, sharing of data, and open communication. The proliferation of IoT devices across a range of sectors, from smart homes and healthcare to transportation and industrial automation, has enabled unprecedented connection and convenience by a survey [48]. IoT gadgets present unprecedented levels of connectedness and convenience, but they also pose serious security threats. Securing these devices and the networks they use is crucial for defending against online threats. Significant security issues, such as unauthorized access, data breaches, and malicious attacks, have been brought up by the IoT devices' quick expansion. In order to safeguard private data and maintain the integrity of IoT networks, it is now vital that IoT devices be secured [49].

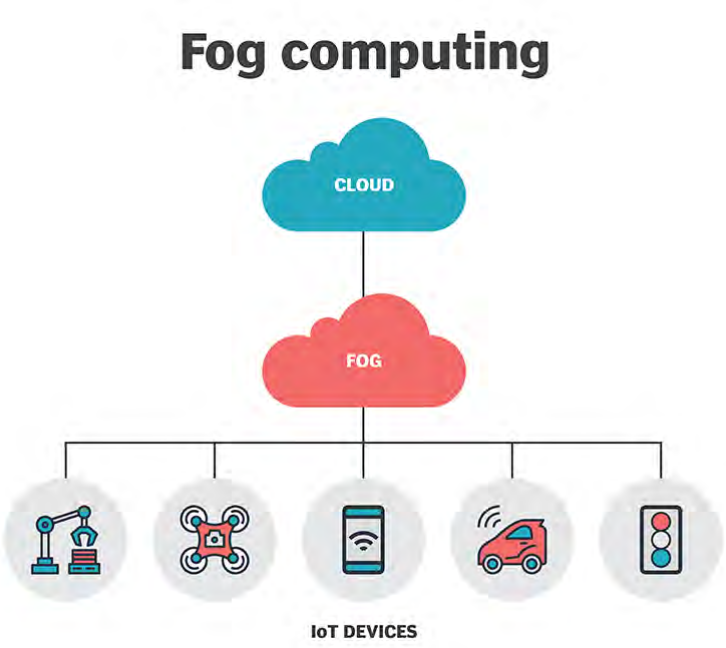


Figure 2.1: Fog Computing

2.1.2 Smart City :

It is anticipated that intelligent cities would improve livability, encourage environmentally conscious growth, and boost the effectiveness of government operations. According to [13], "A smart city's infrastructure is interconnected with billions of devices that, through a variety of applications, including smart homes, smart surroundings, smart environments, and healthcare, can be mutually beneficial for citizens." The most crucial issue arises when operating a huge amount of Internet of Things (IoT) facilities in a smart city environment. To deliver innovative services, thousands of smart things, cars, phones, and people connect with one another, from a communication and information context, the architecture of fog computing can be quite beneficial. The communication network's latency must be addressed in a coordinated manner because it can cause the network to perform poorly. According to the findings in [36], "it is evident that fog computing holds promise as an essential factor in facilitating low-latency access for Internet of Things (IoT) applications." As stated in the literature [4], "a fog computing-enabled architecture serves as a simplistic computational layer that bridges the gap between the cloud and Internet of Things (IoT) layers, effectively tackling the challenges associated with this integration." Moreover, according to the findings of [38], it has been observed that, "computationally intensive data processing tasks can be moved from the cloud-based layer to the fog-based layer of a system through the establishment of a functional layer with reduced complexity." Additionally, the fog layer can also function as a gateway to other higher-level layers.

2.1.3 Security Threats:

The Fog platform's functionality introduces a new point of vulnerability between end users and cloud services, which might possibly be used for malicious activities

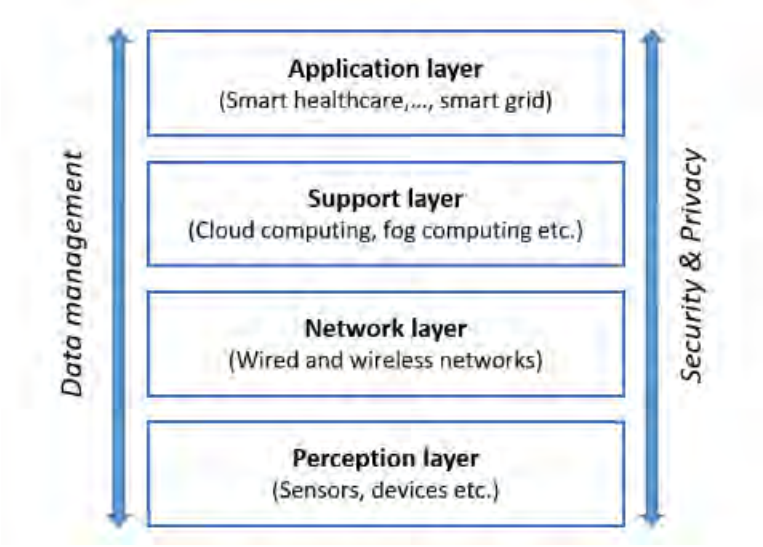


Figure 2.2: IoT-based architecture for a smart city

[8]. As seen in [13], "by using contemporary attacks like background knowledge attacks, collusion attacks, eavesdropping attacks, likability attacks, spam attacks, Sybil attacks, inside curious attacks, outside forgery attacks, and identity attacks it is possible to hack nearly all smart applications that are connected to fog."
The most recent challenges caused by the rapidly changing smart applications, in addition to these issues, are the ones listed below found by various studies.

1. **IoT Botnet Activities:** The recently created IoT botnets pose a severe threat to the IoT network. A good instance of this is the Mirai bot-net, which has the ability to infect a broad range of heterogeneous IoT devices, including printers, routers, DVRs, printers, IP cameras, and webcams, spread infection, and then conduct DDoS attacks against selected servers.

2. **Concerns about privacy in virtual reality:** Virtual reality, or VR technology, has been used by a large number of institutions and entities in digital smart cities, including municipal planning agencies, medical professionals, and the manufacturing sector. The usage of accessible without encryption communications in virtual reality and the sharing of confidential information with third parties, however, both increase the potential for privacy issues.

3. **Threats Posed by AI:** Artificial intelligence (AI) technologies are essential for a wide range of intelligent applications, including the autonomous management of trading platforms, household appliances, and pacemakers. A research done in [13] made the following suggestion: "For instance, internet service providers and device manufacturers might use data mining technologies to thoroughly analyze personal data and to collect sensitive data beyond what is necessary to achieve the fundamental objectives of the related services."

4. **MITM (Man in The Middle):** This sort of attack can quickly propagate throughout a fog computing system. In this type of attack, legitimate gateways that also serve as Fog devices can get exploited or spoofed. Customers of KFC or Star Bar are two examples who have connected to illegal access points that provide false service set identifiers as public trustworthy ones. Once the

attackers seize control of the gateways, victims' private communications will be hacked, as proposed by another study [14].

5. **Mirai:** It is an attacking process which turns into an attacking botnet from a malware to infect IoT devices by launching a severe DDoS attack. The malware finds vulnerabilities of IoT devices, and afterwards, it exploits weak credentials to get the control of the device executing the attacks [27] Largest DDoS attack is possible to generate due to the Mirai even though on well-defended targets. Moreover, it is becoming a growing concern that the rapid growth of vulnerable IoT devices is going to face the potential harm by Mirai [7]. That's why a combination of technical interventions is necessary, which is stated in this study.

6. **DNS spoofing:**

   It is an attack which works through a resolution process by manipulating the Domain Name System diverting the web requests to unintended destinations. Then compromising the DNS infrastructure attackers redirect the request to malicious websites or steal confidential information or launch a malicious virus bypassing security measures [45].

7. **Recon:** This attack gathers detailed information about the target system by exploiting network and data breaches without even directly engaging with the target system. It uses the techniques of scanning, and footprinting with enumeration of the vulnerabilities of the victim systems as well as devices [56]. Different automated tools like FireCompass are intended to expose cloud data, databases, and ports to the attackers. This study evolves in detecting Reckon attacks and continuous monitoring to mitigate the security risks with new edge techniques.

2.2 Related Works

Cloud computing has been a very popular topic amongst the consumers, and the number of businesses utilizing this technology has increased significantly. Yet, the users have faced some new problems related to this novel technology. For example, the latency and bandwidth of cloud computing are a lot more demanding than newer and more efficient models like fog computing. By serving as an intermediate node between the end user and the cloud, this new technology in [21] reduces task latency by bringing compute capacity closer to the user. As fog computing is an even newer topic so very little research has been done to assess its security. But still, there are some noteworthy papers that have made significant progress in this field. Amongst which [18] carried out a survey that successfully identified important security issues in the architectures used in fog computing, further reviewed these issues, and provided solutions as well as direction for future research regarding the topic.

The implementation of fog computing in modern healthcare is immense, various IoT devices are already being used in the field and to integrate them, the fog computing approach is highly considered. One of the most crucial responsibilities in the healthcare industry is keeping a tab on the patient's vital signs. A lot of models are available for this purpose and one such method is that proposed in [10] which uses an inexpensive fog based monitoring system and using this technology, "The Internet of Things (IoT) devices may quickly gather different patient health data, deliver it to the appropriate parties, and provide an automated review for the convenience of the doctor." Another method that tries to mitigate any security concerns related to these IoT devices and the previous model is a security protocol from [3], "It adds a layer of fog-based software between the cloud and the IoT device and makes use of a cloud access security broker to bolster edge node security." On the other hand [5] suggested, "a novel fog computing-based architecture that can manage time-sensitive healthcare data. This design makes use of a sizable geographically scattered system to guarantee data accuracy, consistency, and low latency."

In smart cities utilizing IoT devices, to mitigate against software-based network attacks, [22] has come up with "a new intrusion detection method using decision trees and CNN's". This method [22] has proven to be able to detect abnormal traffic through the IoT network in the smart cities, particularly in the smart traffic management system. The utilization of voice data in smart cities is another use of fog computing. The work done by [6] is able to use speech data collected from the user's smartwatch and send it over to the cloud via a low powered fog-based system. By collecting and processing most of the data in the fog system as shown in [10], the data can be very easily stored in the cloud to carry out the user's request. As smartwatches are relatively low powered devices, the main purpose of the architecture in [6] is to make it easy for the user to issue commands and tasks to the IoT devices by the cloud.

Further improving the work done by [3], the research done by [15] aims to improve the role of IoT devices in smart healthcare systems implemented in smart cities. Also, by using the data collected from the patients as suggested in [17] they are able to assess the time by which the patient will check out and prepare the payment gateway beforehand so that it becomes less of a hassle for the patient. In order to reduce the security issues in fog computing [25] has done an in-depth research on

human behavior and created a decoy based system that generates fake data that seems reliable enough to pass along as real data using fake documents. This data may be generated at a moment's notice if any attacker is able to infiltrate the users network and download any data, then this fake data will be sent and the attacker will download them believing them to be real. This is only possible due to the behavior predicting technology, that maintains a close relationship with the real users data, and thereby helps to keep malicious actors from harming the user. Additionally, tracking packages may be added to the fake dummy data so that the attacker may be tracked and further attacks prevented. Furthermore, [1] proposed the use of logs to handle the security threats that arise in some applications of fog computing in smart cities. By gathering and analyzing different logs of users about how they tend to use the devices gives us the required insight as to where the possible vulnerabilities may be and how to overcome them.

A recent development is the use of machine learning in fog-based systems. For that reason, it is common practice to apply ML in smart cities. So, researchers [19] have done their survey in this field. They tried to determine the capabilities of ML in fog computing and IoT. According to their argument, fog-based ML apps may offer strong end-user and high-service layer capabilities, resulting in more perceptive and intelligent replies to the necessary tasks. Resource allocation, correctness, and reliability constitute the core elements of computational fog, and this article presents a thorough summary of the most recent developments in ML approaches. The study also covers additional ML-related viewpoints, including different software assisting models, methodological frameworks, and datasets. Using these datasets, the IoT devices are able to improve themselves constantly, and the system becomes more reliable. A lot of researchers have been involved in the development of countermeasures against the security threats involving fog computing. A research done in [14] shows, "the implication of fog computing in IoT based smart cities and identified some crucial areas like traffic, smart grid, automated delivery and such that require special attention." They classified their approach based on these security threats and proposed a lot of innovative ideas and furthered the research. Other studies done by [9] also did various research in this field and have contributed to opening new viewpoints to approaching the security threats. Other academics, including [2], have shifted their focus to look for novel threats in the fog computing sector. To ensure that any new threats are able to be resolved quickly, journals like [20] also opened the door to new comparisons between fog and cloud computing, diversifying the field of fog computing even more [21].

The study [43] proposed a model implementing a hybrid detection and mitigation process of IoT botnet attacks like Mirai, Satori, and ZeroAccess using HIDPS and NIDS structure of federated learning. However, it is a hybrid approach of Intrusion Detection System (IDS) Integrating Network Based Intrusion Detection(NID) and Host Based Intrusion Detection System (HIDS). It can identify encountered zero-day attacks in an IoT environment through federated learning as well as enhance security by mitigating attacks with a fog computing orchestrator. This model Minimized single point errors by reducing training overload and prioritizing privacy. Furthermore, this model shows efficiency due to its decentralized training architecture utilizing the federated learning model of 1D-CNN.

In the paper [12], the author implemented Intrusion Detection System (IDS) and Intrusion Prevention System (IPS) in terms of preventing man in the middle attack

in fog computing. Deploying the IDS nodes, The research focuses on monitoring and analyzing the behavior of fog nodes inside distributed fog networks. The proposed model aims to detect and prevent various MitM (Man in the Middle) attacks such as wormhole attacks, packet modification, etc by utilizing lightweight encryption techniques. Also, this study focuses on energy sufficient security measures with lower latency due to the limited resources in fog devices inside fog computing environments.

The research [31] stated a framework for the network based solution using DNS proxy server for the detection and prevention against DNS-based attacks such as DNS spoofing, data exfiltration, DNS tunneling using machine learning based detection as well as enabling real-time analysis for attack detection capabilities. By integrating the visualization process, the Machine learning algorithm classifies and analyzes the attack in real time and further provides a robust defense against the DNS attack. In terms of prevention, the system analyzes the DNS queries, further, It blocks the malicious request setting the domain IP address redirecting to 0.0.0.0. By caching and redirecting domains, the system provides a proactive prevention mechanism mitigating the risks imposed by DNS attacks.

The study [37] emphasizes a defensive deception strategy against reconnaissance attacks in the cloud as well as the IoT environment using Deep Reinforcement Learning (DRL). Their goal was to provide an efficient as well as impactful deception strategy that would outface reconnaissance, increasing attack costs, and hide system assets. By formulating utility functions, They customized DRL agents so that they could generate an efficient deception procedure to find the common vulnerability within operating systems associated with clouds. Also, They were able to hide cloud assets up to 20.58% and increased attacking costs up to 40.40%. The study highlighted Potential defensive deception against reconnaissance attacks integrating artificial intelligence techniques for enhancing cloud security for IoT devices.

The study [16] further highlighted the current challenges and feasible solutions of utilizing fog computing in IoT infrastructure. There are some challenges like safeguarding data against the integrity of fog nodes with malicious IoT devices, further identifying authentication, and building decentralized, secure infrastructure associated with securing fog computing in IoT. There are some approaches to handle those threats including data integrity protection protocol along with anomalous intrusion detection systems with homomorphic encryption techniques. By ensuring data integrity, the decentralized Internet of Things safeguards user privacy by identifying possible dangers. To secure privacy of fog computing. The research [24] proposed a framework for enhancing privacy in Fog layer by discharging security processing tasks by developing secure communication channels with encryption mechanisms. The framework ensures data protection of user credentials by utilizing fog as an intermediary between IoT devices and the cloud. Moreover, the framework follows advanced encryption techniques for user controlled access and safeguards user information against potential threats.

The authors of [42] discussed a number of solutions related to the real-time security challenges in Fog based IoT networks. They propose authentication techniques that include identity based or anomaly based authentication, next they talk about authorization solutions that use trust based and role based access control to limit access to trusted individuals only. These are highly effective solutions that work great in real-time situations with low to no latency. Similar to the method proposed

in [43], this paper also uses a host based IDS to stop malicious traffic. A secure cryptographic key management system is used to mitigate Sybil attacks in the case of large scale data analysis scenarios. Now, looking into the Industrial IoT environment where very large scale data is processed every moment, a novel approach is introduced in [54] to combat the security threats in an efficient manner. This model is called the Efficient Execution of Offloaded IIoT Trusted Tasks, or EEOIT for short. EEOIT works by offering a trust management system using Fog nodes in these large environments. This system is able to differentiate between malicious and regular tasks by using a mixed trust based evaluation technique. The trust value, work size, and deadline are used as features to determine the priority of the task using the technique for Order Preference by Similarity to the Ideal Solution (TOPSIS) algorithm, thus ensuring that high priority tasks get completed first. By increasing job execution on fog nodes, this method enhances overall performance and keeps malevolent nodes from compromising system integrity. To combat the risk of unauthorized access to the IoT system, the study done in [35] introduces a Data Security Management Model (DSMM). By using an extensible authentication protocol and density control weighted election, the proposed model improves data security and privacy in data transfer across IoT devices. DSMM mitigates security risks and ensures that only authorized organizations can access the data by effectively clustering data and implementing authentication processes. This lowers the possibility of unauthorized access and attacks.

The method described in article[11] defends the IoT system from DDoS attacks by employing a two-tiered security layer. In the first layer, a Virtual Private Server is used to ensure secure communication with IoT devices, after which a challenge-response authentication system is used to prevent ant malicious traffic from entering. This integrated approach ensures the security of the IoT system, thus preventing any potential DDoS attacks. Now, evolving from the two layered architecture, [23] proposes a faster and more accurate three level design that helps to deal with DDoS attacks in IoT environments. Like the previous approach, in this model, the first layer also acts as a firewall that filters out any malicious data by comparing it with known malicious signatures. To make this process work in real-time, the firewall is placed closer to the end devices in the Fog node. After that, the malicious data is blocked and sent to the Virtual Servers where they use virtualized network functions to further analyze the data and update the malicious signature database if required. Based on predefined patterns, these servers analyze the incoming data and identify possible DDoS attacks. Finally, data from several local servers are coordinated and merged by the central cloud server. The cloud server analyzes the information to detect subtle attack patterns and improve the overall detection accuracy. In paper, [30] the proposed fog computing DDoS mitigation framework includes an anomaly-based intrusion detection approach as well as a database. The framework detects DDoS attacks using a classification method called k-Nearest Neighbors (KNN). By deploying the mitigation framework, the fog computing resources are used to address the resource limitations of IoT devices. Network traffic is scanned by the framework against attack signatures recorded in a database. The administrator receives an alert and the flow is stopped if an attack is found. In the event that the traffic is normal, it is provided using a classifier that makes the distinction between suspicious and regular traffic, the regular traffic is allowed to go to the end device and the suspicious data is recorded and the database signatures are updated using them.

The model demonstrated in [55] uses a technique called FASA to detect TCP-SYN Flood DDoS attacks. This method has a detection phase and a prevention phase. In the detection stage, an adaptive neuro-fuzzy inference system (ANFIS) is used to detect the malicious traffic, while in the prevention stage, the SDN controller drops the malicious packets and blocks the hosts port to stop further attacks. This prevents the attacker from doing more harm.
The use of Deep Learning models in IoT environments has been an efficient way to detect security threats. This has been the case for [44] where IDS is used along with Deep Learning models to track and analyze the malicious traffic. Analyzing these helps the IDS to strengthen the security of the IoT system and stop policy breaches. The IoT ecosystem's overall security posture is improved by the detection of numerous attack types, including replay, sinkholes, and denial of service assaults, made possible by IDS integration with IoT systems. Furthermore, according to the research in [51], there are a number of methods to ensure security in the Internet of Things. These include using feature engineering and optimization to improve security attributes, rule-based systems to make intelligent decisions based on extracted rules, classification and regression techniques to predict and categorize security incidents, and deep neural network learning-based approaches like MLP, CNN, and RNN to build sophisticated security models. Another model proposed in [57] acts in the host level for the purpose of detecting intrusions in end devices; the suggested Multi-Stage Intrusion Detection System (MS-IDS) uses Machine Learning (ML) models in addition to kernel-level and end device data to identify different kinds of attacks. Here nine layers of security measures are taken to ensure the integrity of the network. The study conducted in [29] also employs Deep Learning models such as SVM, MLP, and CNN for intrusion detection purposes in IoT environments. Random Forest is used in the case of large amounts of data and CNN is used to reduce the information attributes, helping to quickly and effectively detect suspicious traffic and resolve them. The research in [47] further works on the previous research and uses Deep Learning models such as LSTM and CNN's to identify any intrusion in the system. In particular, the model uses Gated Recurrent Units (GRU), Long Short-Term Memory (LSTM), and Convolutional Neural Networks (CNN) for this purpose. After testing this model in IoT environments, the researchers observed that the detection capability of these models was far better compared to other machine Learning models.
This paper [26] provides the insight of securing IoT devices in smart city environments to enhance living through traffic management, public health, and resource management by IoT technologies. The author's goal was to address Security issues of IoT devices which have been growing in numbers exponentially. So, they stated their concern about the vulnerability of those devices. Against cyber attacks and also analyzed various security breaches in the IoT environment such as Cyberwarfare and DDoS attacks on the internet. Moreover, they also presented preventive responses against those threats.
In the Study of [52], the authors proposed a model which incorporates multilevel authentication infrastructure for the attack detection and mitigation in the IoT environment. They stated that traditional ways of prevention against cyber attacks are not being useful anymore. That's why they introduced a multi-level authentication process which includes a verification process by hashing, Chebyshev polynomial To enhance the security of the IoT environment in smart cities.

The research of [53] analyzed the vulnerability of IoT systems and implemented advanced security measures using NodeMCU-ESP8266 for a real-time reaction against cyber threats. It also offers hardware based security models such as Hardware Security Modules (HSMs) along with Zero Trust Architecture (ZTA). It emphasized its goal of providing cyber threat detection and also took preventive measures for protecting sensitive data in the IoT environment.
The author of the Research paper [34] discussed challenges that occurred by DDoS attacks which affect critical services within the network by raising traffic from numerous compromised devices. The author has utilized two datasets of SDN and CICDDoS2019 in order to train and test eight individual models integrating Deep Neural Network (DNN) to detect DDoS attacks over the system to enhance security measures on the SDN network.
The proposed model of this paper [33] focused on intrusion detection as well as mitigation framework for SDN-controlled IoT networks consisting of three major components such as IDS sensors, IDS manager to evaluate network traffic, and SDN controller to enhance safety standards. By leveraging the dynamic reconfigurability characteristics of SDN, this model is highlighted in detecting cyber threats by ensuring secure communication among IoT applications.
The paper [39] proposed a model of a machine learning algorithm for Intrusion Detection and Prevention Systems (IDPS). Moreover, it analyzes the vulnerabilities of the current IoT environment and focuses on developing effective defense mechanisms, especially against Mirai and Bashlite botnets. The research also highlights robust security measures like firmware updating, and network control to prevent attacks for developing a safer IoT ecosystem.
In the study of [32], the study proposed a model which integrates Deep Learning(DL) along with Software-Defined Networking (SDN) and developed a real-time middleware solution to detect cyber attacks and further goes for prevention in smart homes. It used Raspberry Pi, and Zodiac-Fx SDN switch and also utilized deep learning classifier to detect and mitigate attacks like DDoS in smart home's IoT environment. Its goal was to build up a low cost effective security framework which accurately identifies and prevents cyber threats and enhances the reliability of IoT networks in smart cities.
The authors of [40] proposed a framework integrating with deep learning which generates a cyber kill chain model to detect multi step attacks using four components IoTDSCreator(dataset generator), IoTEDEF (anomaly detection of multi agent), for predicting attack as well as management IoTPredictor and IoTAtM. Its goal was to build a self-evolving system which detects early stage attacks and mitigates them.
The paper [28], highlighted a framework of Intrusion Detection System (IDS) to enhance fog layer security using machine learning models. It worked on the KDD Cup'99 dataset utilizing the models of K-Means, Decision Trees, and Random Forest algorithms to detect attacks. It focused on the limitations of cloud-based IDS and further leveraged fog computing by reducing latency while improving detection efficiency and it found K-Means as the most impactful algorithm with higher accuracy and fastest computing time in IoT environments.

Chapter 3

Methodology

3.1 Proposed Model

The major goal of this research is to use machine learning to identify potential security threats and identify security weaknesses in Smart Cities. Analyzing the security threats will help the Fog Computing of IoT devices to take necessary steps to mitigate the security threats. The data collection process is the most vital task in this work. As we aim to detect Security Threats related to the use of fog technology in IoT based smart cities using different activities of user interaction and data will help us to analyze the activity within the IoT devices. After data collection, processing needs to be ensured where we will sanitize, scale, and standardize the collected data. We can now apply qualitative and quantitative data analysis to process and interpret the collected data. After data analysis, we will be able to identify the common threads, and patterns which will specify the security vulnerabilities in fog computing. Based on the analysis we can figure out the mitigation strategies to defer the vulnerabilities. After the data has been preprocessed to make it appropriate for training using a security model, the model will go through the training and testing phases, and the results will be received. The performance accuracy of our model will be determined based on the results we obtain. If the accuracy is high the trained model will be considered as a successful model to detect security threats. On the contrary, if the accuracy is low, the data will be sent to the processing stage again.

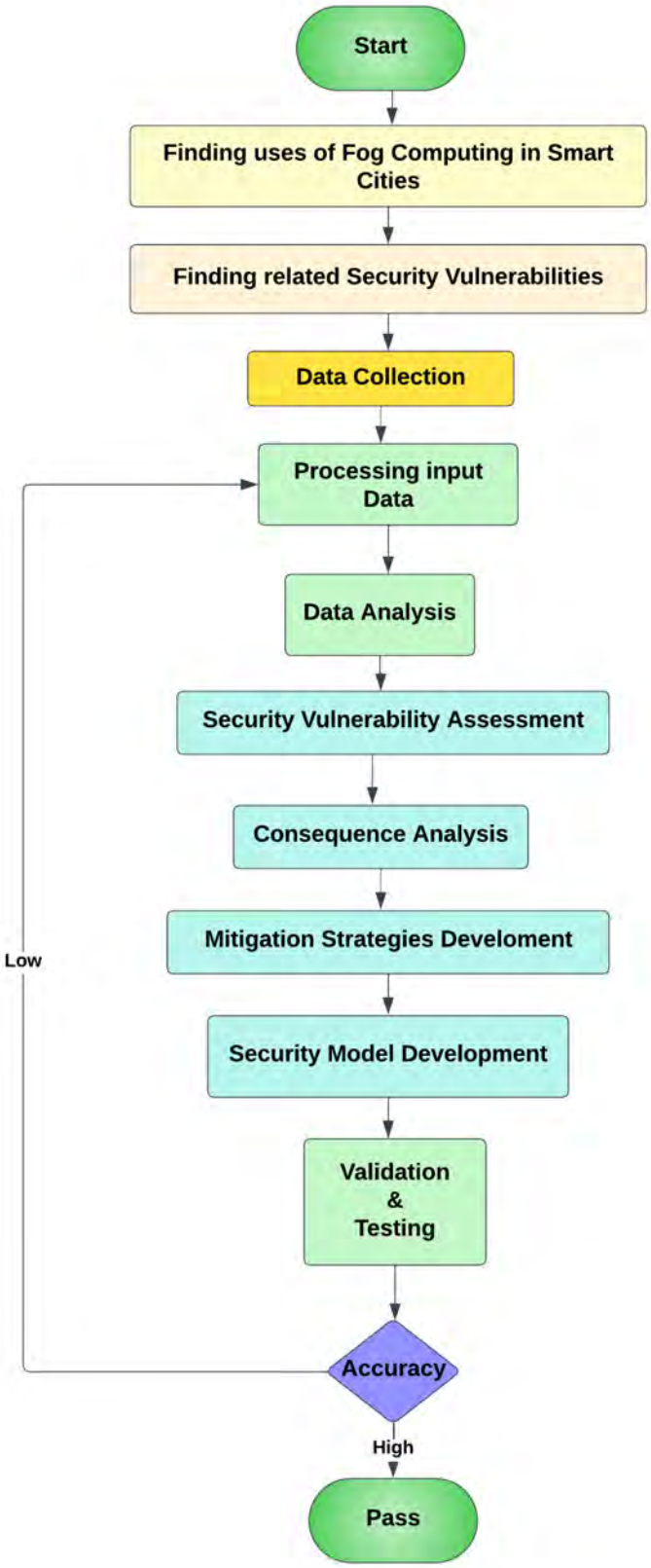


Figure 3.1: Proposed Model

Finding Uses of Fog Computing in Smart Cities: Finding and documenting the applications of fog computing within smart cities, Research analyzes how fog computing works with IoT devices to enhance overall services and infrastructure. This can include traffic management, smart grids, hospital management, and environmental monitoring.
Finding Related Security Vulnerabilities Finding out potential security threats associated with the uses of fog computing. Review the literature and conduct risk assessments to compile some vulnerabilities. Consider threats like data unauthorised access, breaches, and denial of service attacks, Mirai attack and more.
Data Collection Accumulation of relevant data from various sources, user interactions with IoT devices, network traffic, sensor outputs, Medical data. Ensure extensive data coverage to support the data analysis. Research methods include surveys, logs, and databases.
Processing Input Data Sort the collected data by removing errors, duplicates, and irrelevant information. For ensuring consistency, scale the data to normalise it. Organised the data to follow a certain format and structure and make it suitable for the data analysis. This step confirms data usability and efficiency.
Data Analysis Apply both qualitative and quantitative analysis methods to interpret the processed data.By using different tools like, statistical tools, machine learning algorithms, and other techniques.
Security Vulnerability Assessment Identify and detect common vulnerabilities and security weaknesses among Iot devices. By using the analysed data to find out specific areas of concern within the particular environment. Generate a detailed report of potential security threats and their connection.
Consequence Analysis Check the possible impact of the security threats. Consider the seriousness and probability of threats exploiting these vulnerabilities. Analyse how these security threats could affect smart city operations and services.
Decision: Accuracy Evaluate the accuracy of the security model developed in those mentioned steps. If Low Accuracy: Mitigation Strategies Development: Create strategies to detect and mitigate vulnerabilities. This includes updating security measures. Loop back to Processing Input Data to process and improve the data, and analyse it again. If High Accuracy: Proceed to the next step.
Mitigation Strategies Development Establishment strategies and measures to mitigate the particular threats and attacks. This could require implementing new security protocols, shifting to a new software, increasing encryption methods, or improving access controls.
Security Model Development Generate a security model to detect and counter some particular security threats. Train the model using the processed and analysed data.Also ensure the model is competent in threat detection and mitigation within the particular environment.
Validation & Testing Validate and test the developed security model. Run the model on test datasets to assess its performance, accuracy, and authenticity. Make necessary adaptations to improve its effectiveness.
Pass If the security model achieves high accuracy, it is considered successful. The model is ready to detect and mitigate security threats in fog computing environments within smart cities.

3.2 Dataset Description

Our collected dataset is being originated from the CIC network of IoT which is available combining in both pcap and csv format capturing original data from various scenarios [50]. Here, Pcap file accommodates all packets of the feature extraction detail whereas csv file contains the simplified version of features. This dataset comprehends attacks, necessary data generation, and extraction and labeling it for the further ML assessment. Various tools are being used to capture the attack on real time IoT scenarios of network traffic and further, the result is being captured in a pcap file. Later, TCPDUMP and DPKT are being used for preprocessing with feature extraction and conversion to store in the dataset. CSV datasets show the feature combinations and facilitates the attack detection and classifications based on machine learning models. This dataset contains large data of around 13 GB which needs to be further encountered for comprehensive processing including data cleaning to discard incomplete packets. In addition, this CICIoT2023 dataset presents a valuable foundation for IoT security utilizing ML models stating the relevance of characteristics, mean values, and statistical measures for future research.

	mean	std	min	25%	50%	75%	max
flow$_d uration$	5.707423	272.545466	0	0	0	0.105109	218108.739996
Header_Length	76685.740945	460930.919192	0	54	54	286.2	9907147.75
Protocol Type	9.066693	8.944772	0	6	6	14.44	47
Duration	66.351042	14.019758	0	64	64	64	255
Rate	9063.656307	99518.349428	0	2.093149	15.768890	117.738154	8388608
Srate	9063.656307	99518.349428	0	2.093149	15.768890	117.738154	8388608
Drate	0.000006	0.008684	0	0	0	0	29.715225
fin_flag_number	0.086606	0.281257	0	0	0	0	1
syn_flag_number	0.207280	0.405358	0	0	0	0	1
rst_flag_number	0.090533	0.286944	0	0	0	0	1
psh_flag_number	0.087694	0.282850	0	0	0	0	1
ack_flag_number	0.123378	0.328871	0	0	0	0	1
ece$_f lag_number$	0.000001	0.001184	0	0	0	0	1
cwr_flag_number	0.000001	0.000941	0	0	0	0	1
ack_count	0.090559	0.286432	0	0	0	0	7.4
syn_count	0.330221	0.663398	0	0	0	0.06	12.87
fin_count	0.099085	0.326948	0	0	0	0	248.32
urg_count	6.243520	71.843823	0	0	0	0	4312.5
rst_count	38.488735	325.448905	0	0	0	0.01	9613
HTTP	0.048204	0.214198	0	0	0	0	1
HTTPS	0.055139	0.228252	0	0	0	0	1
DNS	0.000130	0.011406	0	0	0	0	1
Telnet	0	0.000192	0	0	0	0	1
SMTP	0	0.000272	0	0	0	0	1
SSH	0.000042	0.006463	0	0	0	0	1
IRC	0	0.000384	0	0	0	0	1
TCP	0.573770	0.494528	0	0	1	1	1
UDP	0.212020	0.408739	0	0	0	0	1
DHCP	0.000002	0.001345	0	0	0	0	1
ARP	0.000066	0.008136	0	0	0	0	1
ICMP	0.163673	0.369979	0	0	0	0	1
IPv	0.999887	0.010634	0	1	1	1	1
LLC	0.999887	0.010634	0	1	1	1	1
Tot sum	1308.633248	2615.082056	42	525	567	567.54	116053.4
Min	91.608917	139.720449	42	50	54	54	5858
Max	182.047585	524.388465	42	50	54	55.26	41814
AVG	124.690882	241.122562	42	50	54	54.049618	11600.474325
Std	33.356670	160.454938	0	0	0	0.371910	10996.260915
Tot size	124.718095	241.730018	42	50	54	54.06	13098
IAT	83186023.3826	17048700.777	0	83071565.7	83124522.02	83343908.46	167639436.04
Number	9.498647	0.819275	1	9.5	9.5	9.5	15
Magnitue	13.122379	8.630164	9.17	10	10.392305	10.396715	145.390447
Radius	47.139697	226.936791	0	0	0	0.505921	15551.061321
Covariance	30767.167317	324891.301223	0	0	0	1.344216	143542736.693839
Variance	0.096447	0.233044	0	0	0	0.08	1
Weight	141.516561	21.070656	1	141.55	141.55	141.55	244.6

Table 3.1: Dataset Description

The flow_duration feature is essential for differentiating between malicious connections that end quickly and extended user communications because it shows the complete amount of time a network flow is active. The variable *Header_Length* measures the overall length of all packet headers, suggesting the possibility of packet fragmentation or extra options that may be typical of specific attacks. The term

Protocol Type refers to the type of communication protocol being used, such as ICMP, TCP, or UDP, each of which has unique usage patterns and vulnerabilities. Rate specifies the quantity of packets transmitted per second and helps in the detection of floods where packet rates are unusually high. Duration indicates the active length of the flow in seconds and provides a metric for session analysis. The terms Srate and Drate refer to the number of packets transmitted per second from the source to the destination, respectively between which, abnormally high rates suggest the possibility of DDoS attacks. Specific functions are served by TCP flags such as fin, syn, rst, psh, ack, ece, and cwr, as well as their corresponding counts ($ack\_count, syn\_count, fin\_count, urg\_count, rst\_count$), which are altered in a variety of attack scenarios. Application-level protocols such as HTTP, HTTPS, DNS, and others have binary signals that indicate how the protocol is used inside the flow. These signals are essential for detecting certain application-layer attacks. TCP, UDP, DHCP, ARP, ICMP, and other network and transport layer protocol indicators are crucial to different network behaviors and vulnerabilities. In order to identify network anomalies, statistical indicators such as Tot sum, Min, Max, AVG, and Std are crucial as they offer an overview of the intensity and variability of the flow. By measuring the intervals between packets in a flow, IAT (Inter-Arrival Time) reveals unusual patterns that are frequently observed in network scanning or attacks. Number detects flooding attacks and large-scale transfers by counting all of the packets in a flow. For a multidimensional study of flow properties, advanced statistical measures including Magnitude, Radius, Covariance, Variance, and Weight are utilized.

In this research, We intend to detect several attacks on IoT devices and that's why we are working on the related data that contains various attacks being faced by IoT devices in a particular network. Our collected dataset is being constructed by examining 105 devices against 33 attacks on an IoT topology. Moreover, those 33 attacks (DDoS-ICMP_Flood, DDoS-UDP_Flood, DDoS-TCP_Flood, DDoS-SYN_Flood, DDoS-PSHACK_Flood, DDoS-RSTFINFlood, DDoS-SynonymousIP_Flood, DoS-UDP_Flood, DoS-TCP_Flood, DoS-SYN_Flood, Mirai-greeth_flood, Mirai-udpplain, Mirai-greip_flood, DDoS-ICMP_Fragmentation, MITM-ArpSpoofing, DDoS-UDP_Fragmentation, DDoS-ACK_Fragmentation, DNS_Spoofing, Recon-Host-Discovery, Recon-OSScan, Recon-PortScan, DoS-HTTP_Flood, VulnerabilityScan, DDoS-HTTP_Flood, DDoS-SlowLoris, DictionaryBruteForce, BrowserHijacking, Sql-Injection, CommandInjection, XSS, Backdoor_Malware, Recon-PingSweep, Uploading_Attack) are being distinguished into 7 classes such as Mirai, spoofing, Web-based attack, brute forcing, DoS, and DDoS.

This dataset contains 27 million rows with 47 columns which would be a data heavy evaluation We tested our models to detect the attack using a machine learning algorithm splitting the dataset of 80% for training the models and the remaining 20% are utilized for testing.

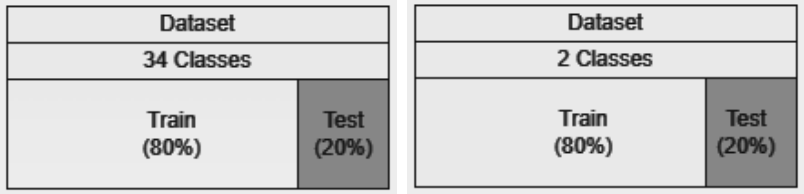


Figure 3.2: Dataset Prepossessing

Then We further categorized the data in 2 classes as well as 34 classes in 3.2 to make the comparison more precise to detect whether it is an attack or benign. Moreover, before running the model, we dumped the null values and converted categorical values to binary according to the requirement of the model.

3.3 Dataset Analysis

The system's architectural development and its ability to predict the suspicious attacks upon the security breaches define the Precision ability of the system against the unusual occurrences. It should have to be discussed the ratio between the pre-prediction of the attack (True positive Count as TPC) and the exact outcome after prediction [TPC + FPC(False Positive Count)]. It can be described mathematically by this equation given below,

$$PrecisionCount = \frac{TPC}{TPC + FPC} \tag{3.1}$$

Recall Detection: It defines the capability of the network to recognize the attack on the currently active network. It can be expressed mathematically using True Negative Count (TNC) and False Negative count (FNC) by this equation,

$$RecallDetection = \frac{TNC}{TNC + FNC} \tag{3.2}$$

Accuracy: it refers to the framework's capacity to accurately identify a normal packet as a "normal packet" and a malicious packet as an "attack packet." It describes the proportion of successful forecasts across the entire set of samples. It is represented logically by Equation (3):

$$Accuracy = \frac{(TPC) + (TNC)}{TPC + FPC + TNC + FNC} \tag{3.3}$$

F1 - score: is equivalent to the precision count and Recall detection's harmonic mean. For standard and malicious traffic, it indicates the percentage of precise forecasts in the test data set. Equation (4) provides a logical definition for it.

$$F1Score = 2 * \frac{PrecisionCount * RecallDetection}{PrecisionCount + RecallDetection} \tag{3.4}$$

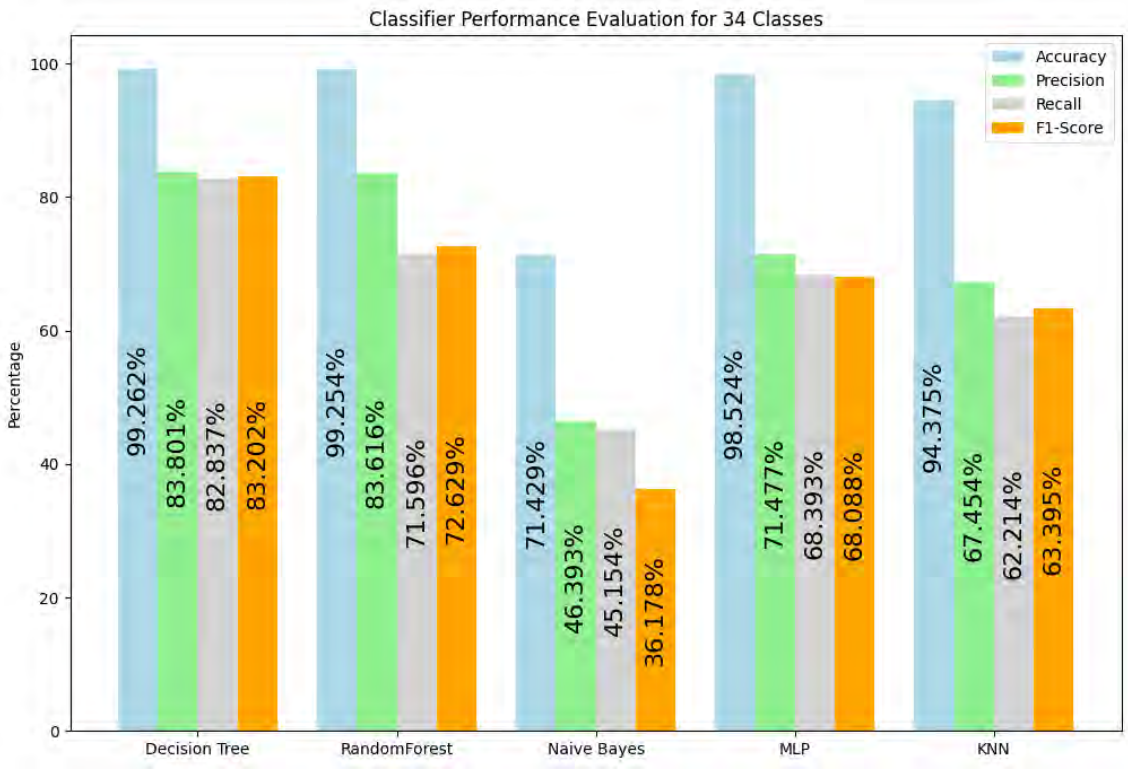


Figure 3.3: Performance comparison of different models for 34 classes

By analyzing the dataset with the label of 34 classes, the performance of different models is shown using the figure 3.3. The Gaussian Naive Bayes model provides us with the lowest result for the accuracy which is 71.43% and opposite to that, the Decision tree and Random Forest model provide us with a good accuracy of 99.26% and 99.25% respectively.

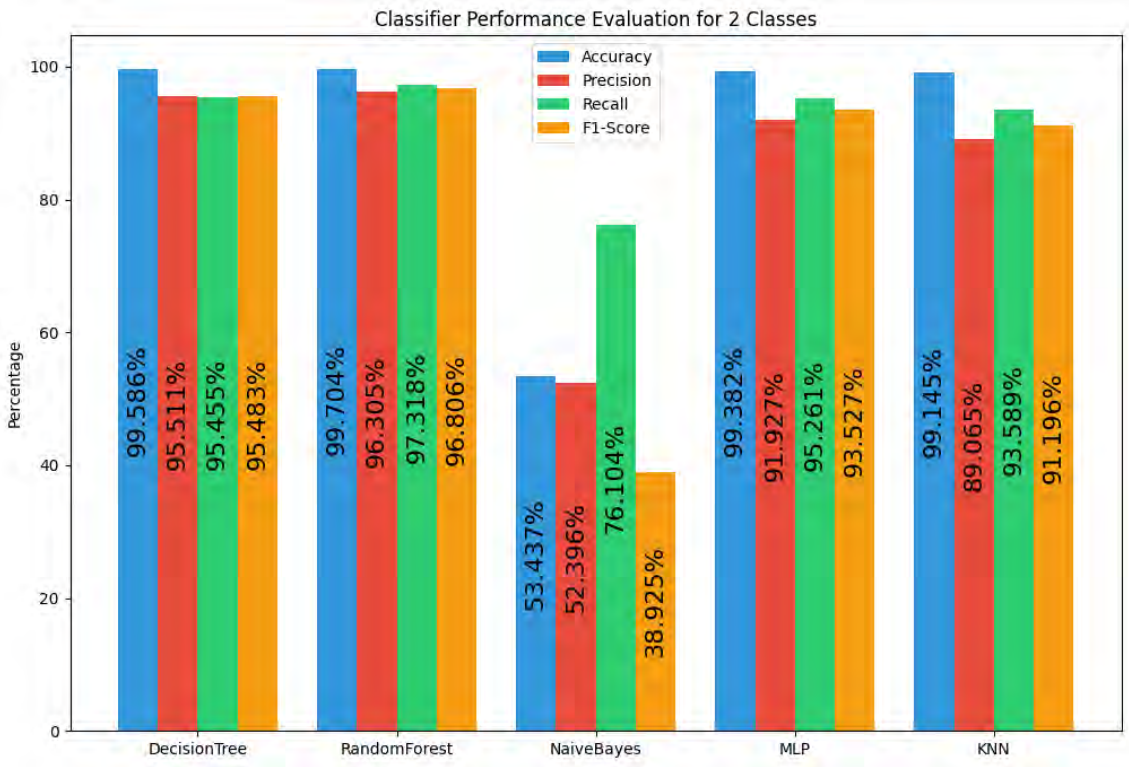


Figure 3.4: Performance comparison of different models for 2 classes

Furthermore, the dataset with label of 2 classes, the performance of different model is shown using the figure 3.4. We can see that the Gaussian Naive Bayes model is not suitable for this dataset. We know that, the Gaussian Naive Bayes model does not perform well when the features are highly correlated. Though, there is remarkable progress on other models.

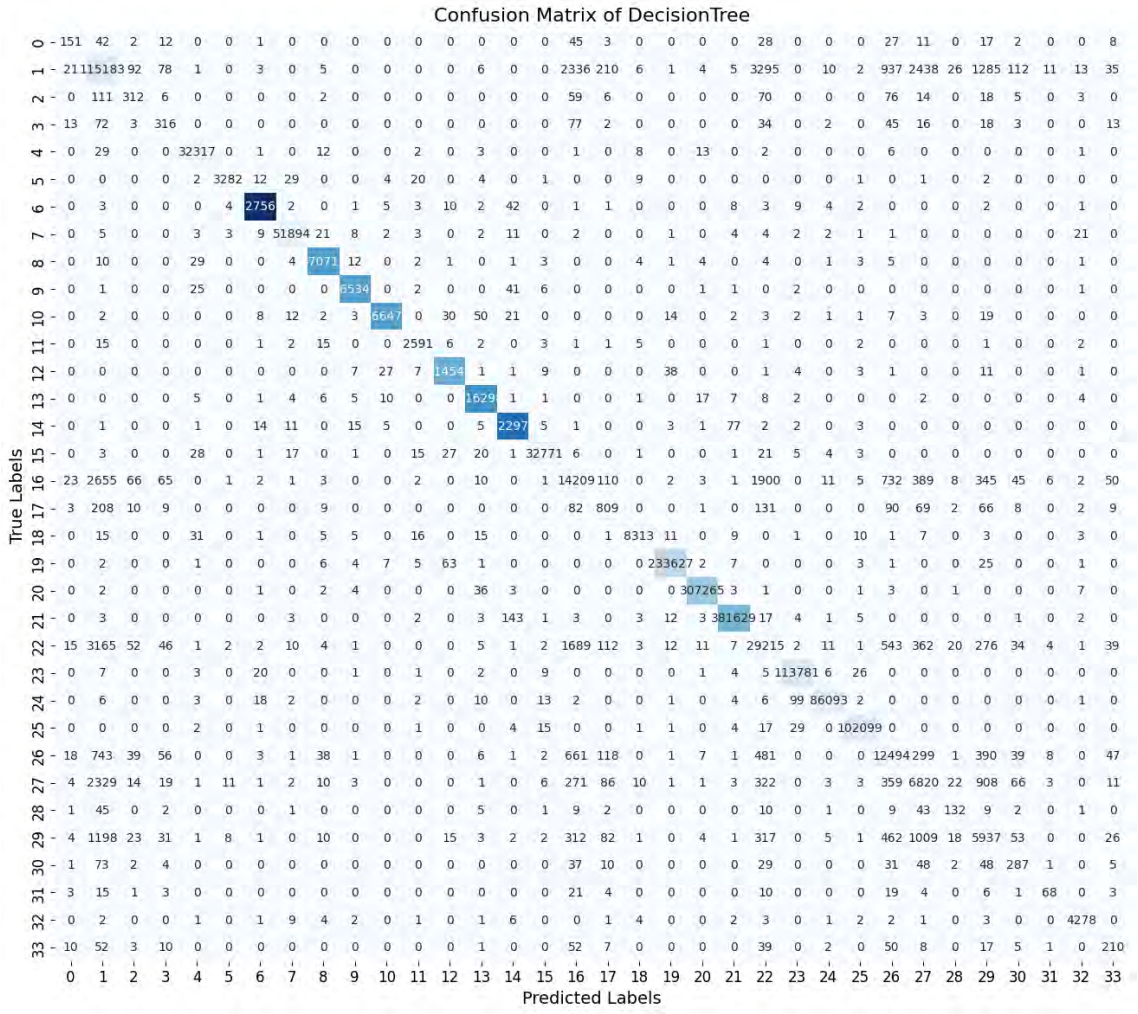


Figure 3.5: Confusion matrix of Decision Tree for 34 classes

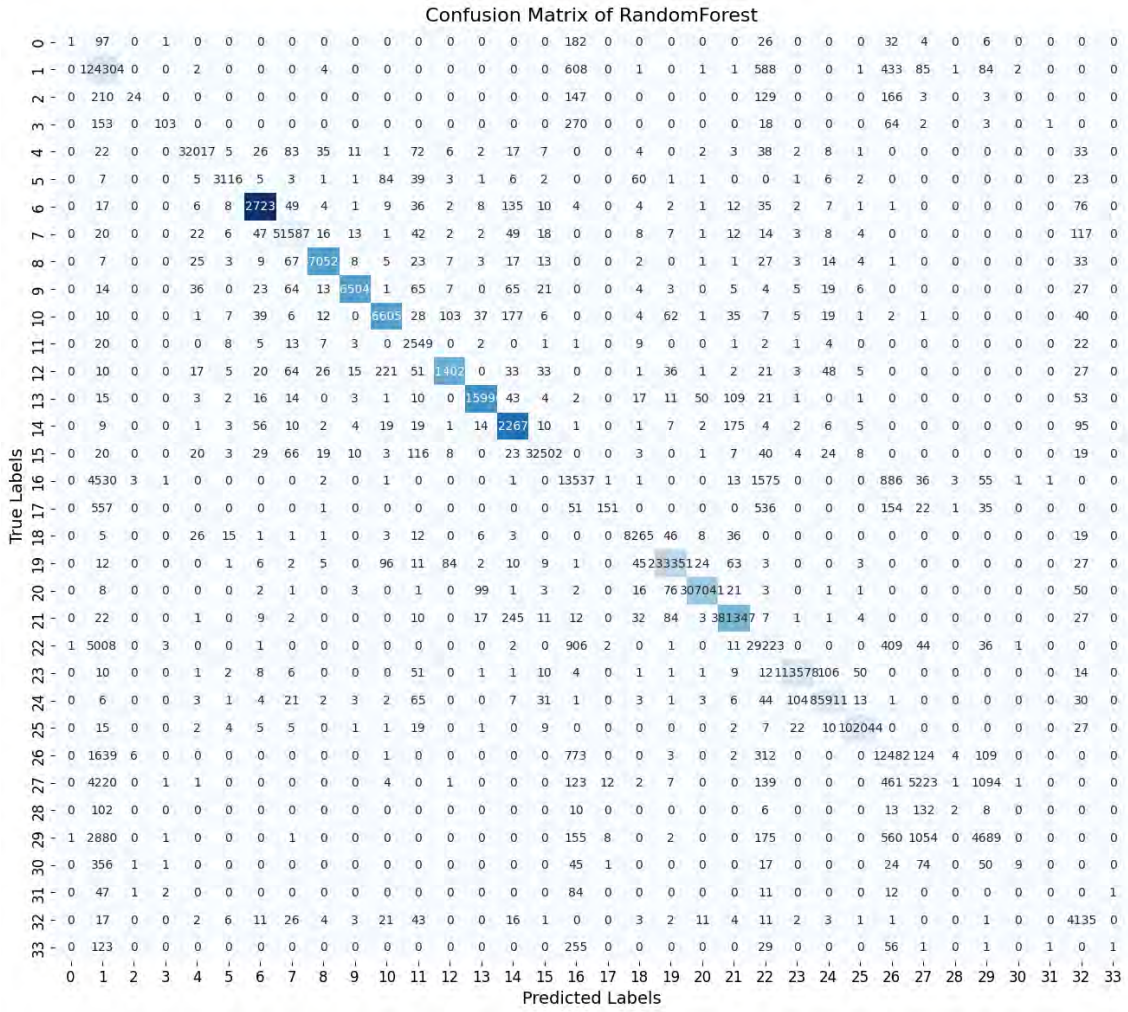


Figure 3.6: Confusion matrix of RandomForest for 34 classes

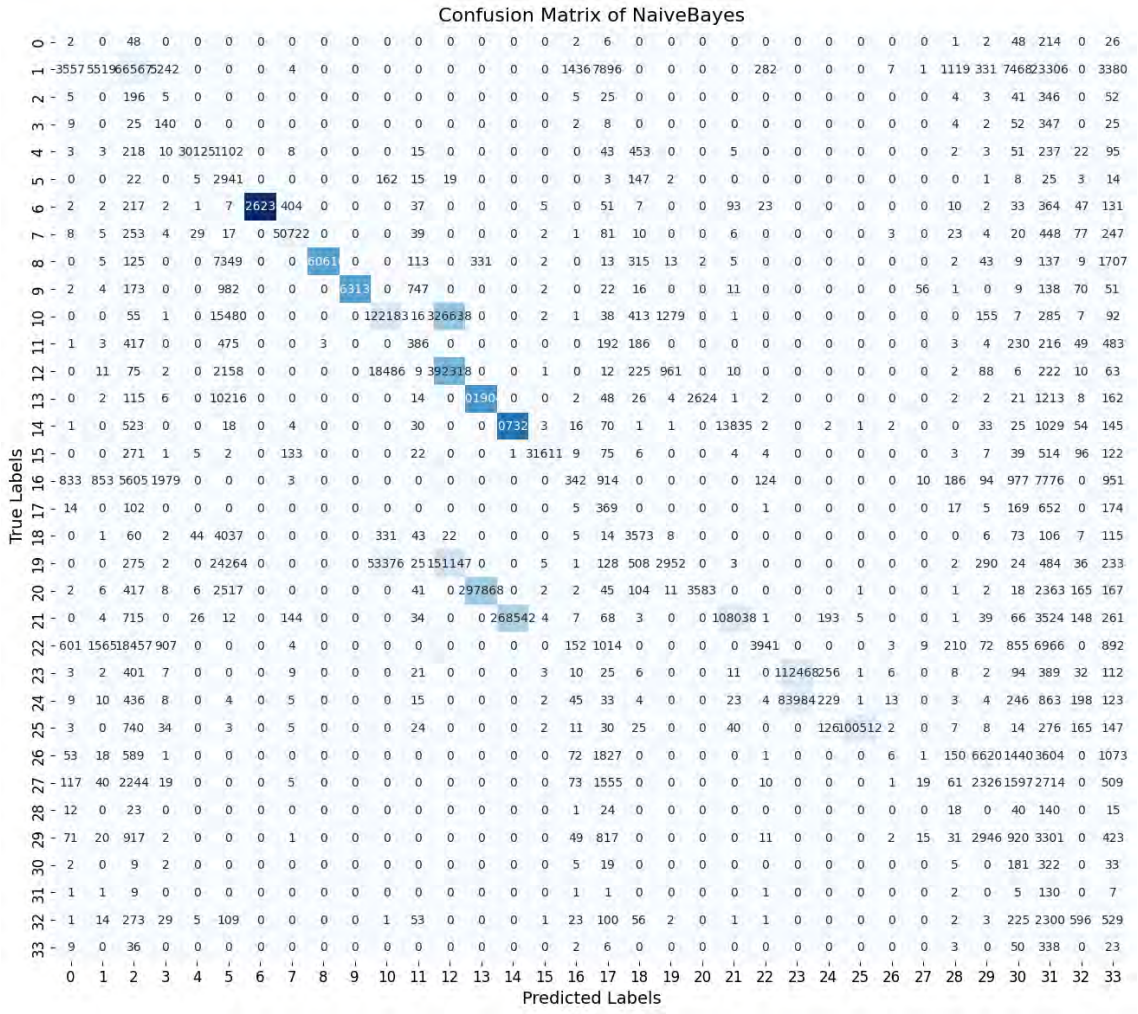


Figure 3.7: Confusion matrix of Naive bayes for 34 classes

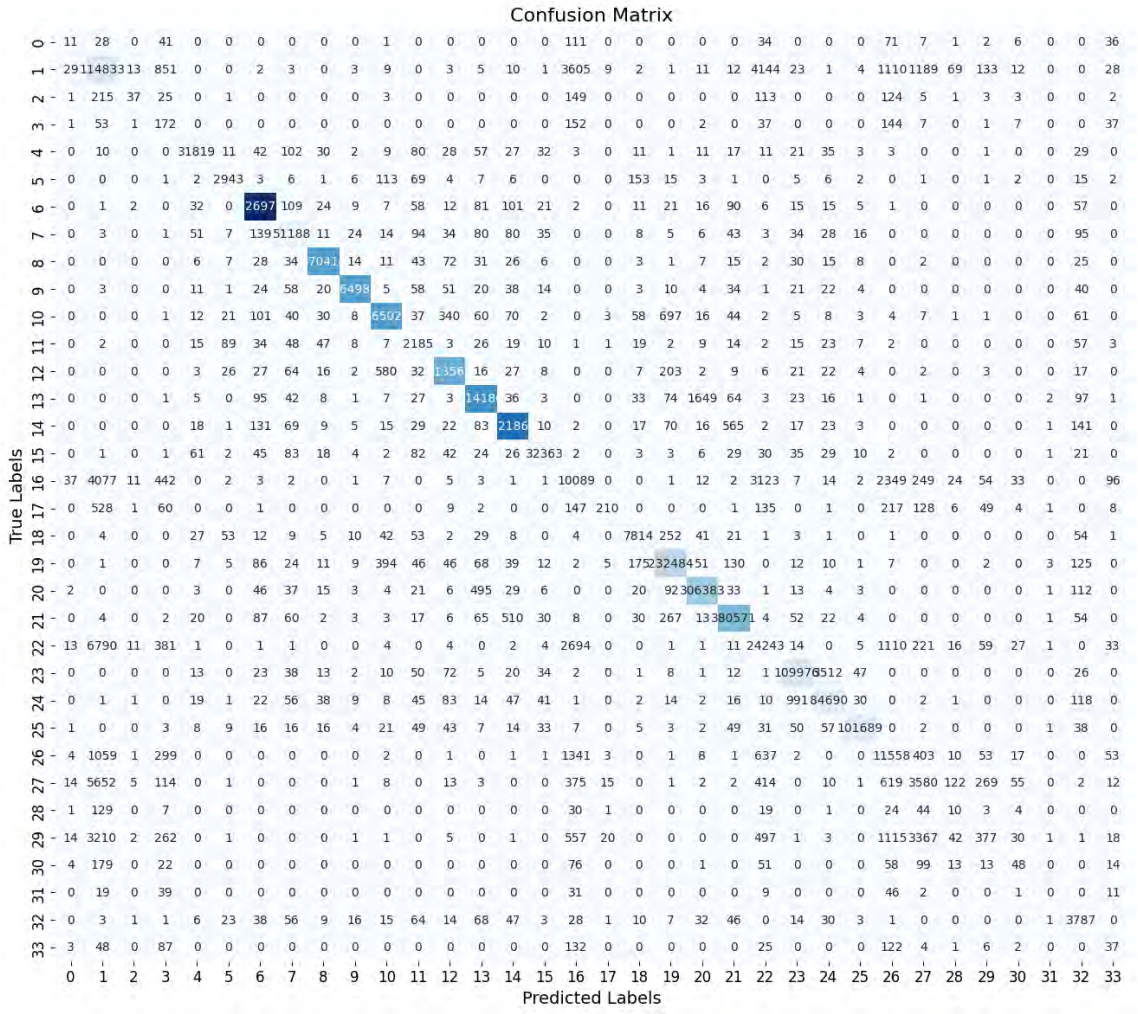


Figure 3.8: Confusion matrix of MLP for 34 classes

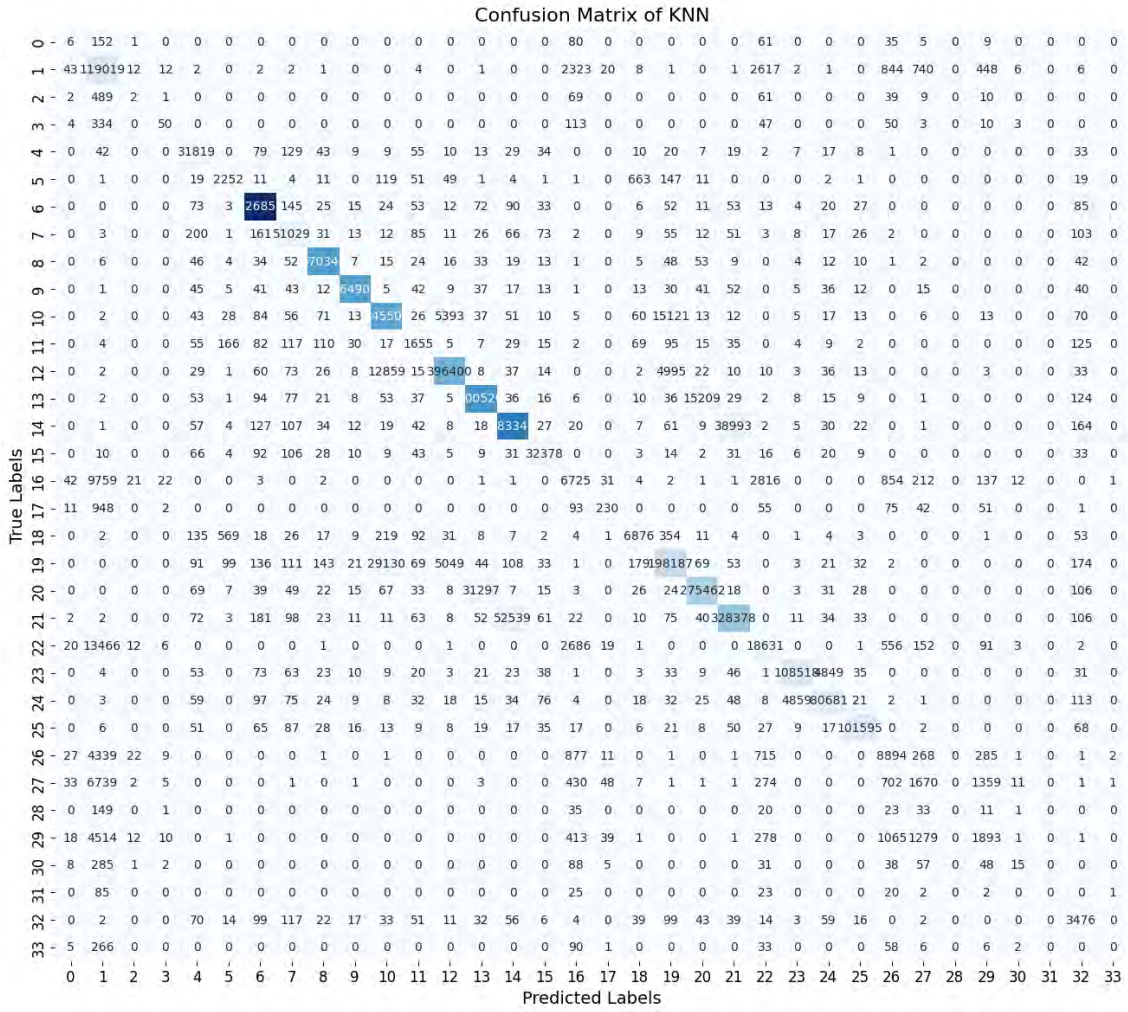
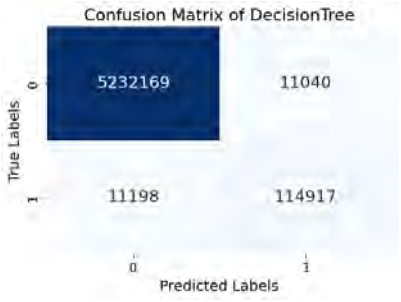
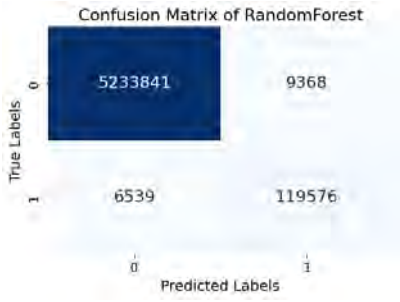


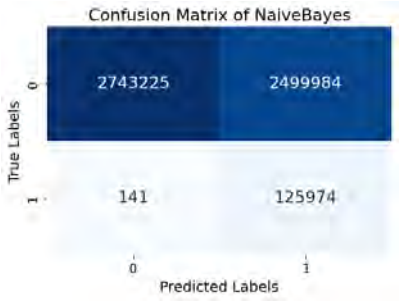
Figure 3.9: Confusion matrix of KNN for 34 classes



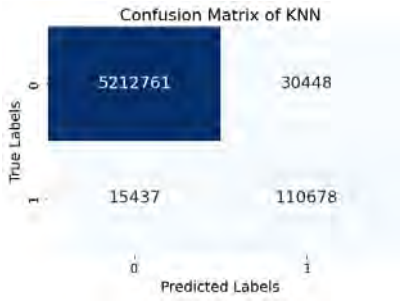
(a) Decision Tree



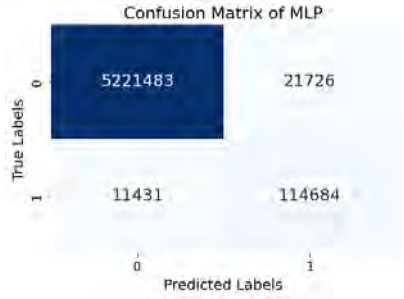
(b) Random Forest



(c) Gaussian Naive Bayes



(d) KNN



(e) MLP

Figure 3.10: Confusion matrix of different models for 2 classes

The confusion matrix derived from our classification models for 34 classes figure 3.5, 3.6, 3.7 3.8, 3.9 and 2 classes figure 3.10. The number of true positive count (TPC), true negative count (TNC), false positive count (FPC), and false negative count (FNC) cases is represented by each cell in the matrix.
With respect to 2 classes, the accuracy of the models such as KNN, Decision tree, Random forest, MLP, and Naive Bayes are respectively 94.38%, 99.26%, 99.25%, 98.52%, 71.43% .
Likewise, In respect to the 34 classes, the accuracy for those models are respectively 99.1%, 99.59%, 99.70%, 99.38%, 53.44%. To compare the outcome based on 34 and 2 classes, the accuracy for each model is nearly the same. Also, it indicates that we can detect those attacks precisely whether it is on the parameter of 2 or 34 classes which would completely depend on the architecture or complexity of the network but the detection result would be very close to each other.

3.4 Algorithm

We have implemented the algorithm to simulate attack detection and mitigation techniques specifically designed for Fog Computing in IoT-enabled smart cities. Our method creates a realistic network environment where different kinds of network attacks are produced and monitored using mininet and ryu controller. Our system can efficiently detect and mitigate malicious activities by utilizing machine learning classifiers, which improves the security and resilience of Internet of Things devices with limited computing capacity.

Algorithm 1 Topology

DEFINE class MyTopo(`Topo`):
  METHOD build():
    ADD switches $s1$ to $s6$ with OpenFlow 1.3 support
    ADD hosts $h1$ to $h18$ with specific MAC addresses, IP addresses, and CPU allocation
    ADD links between:
      Each host and its corresponding switch
      Switches in a linear topology
DEFINE function startNetwork():
  CREATE instance of MyTopo
  DEFINE remote controller $c0$ with IP '_._._._' and port
  START the network
  OPEN Mininet CLI for user interaction
  STOP the network

Algorithm 2 Controller

Require: Network traffic flows, trained machine learning models
Ensure: Detection and mitigation of network attacks

Initialization:
Initialize classifiers:
Start monitoring thread
Datapath State Management:
function STATE_CHANGE_HANDLER(ev)
    datapath ← *ev.datapath*
    if *ev.state* == `MAIN_DISPATCHER` **then**
        Register *datapath* in `datapaths`
    else
        Unregister *datapath* from `datapaths`
    end if
end function
Network Monitoring:
while True **do**
    for each *dp* in `datapaths.values()` **do**
        REQUEST_STATS(*dp*)
    end for
end while
Request Flow Statistics:
function REQUEST_STATS(datapath)
    Send flow stats request to *datapath*
end function
Process Flow Statistics:
function FLOW_STATS_REPLY_HANDLER(ev)
    for each *stat* in *ev.msg.body* **do**
        Calculate flow features
        Append features to `stats_df`
        FLOW_PREDICT([$flow\_features$], ev.msg.datapath)
    end for
end function
Train Machine Learning Model:
function FLOW_TRAINING
    for each classifier **do**
        Train and evaluate accuracy
        Update best model
    end for
end function
Predict and Mitigate Attacks:
function FLOW_PREDICT(flow_features, datapath)
    Predict traffic type using the best model
    if malicious flow detected **then**
        INSTALL_DROP_FLOW(datapath, match)
    end if
end function
Install Drop Flow:
function INSTALL_DROP_FLOW(datapath, match)
    Create flow mod message to drop packets
    Send flow mod message to *datapath*
end function

Chapter 4

Experimental Evaluation

4.1 Experimental Settings

4.1.1 Ryu Controller

The included Ryu controller script incorporates an enhanced Ryu program that extends the functionality of a basic switch to include monitoring of networks and detection of attack features. Using the OpenFlow protocol, this Ryu program, called Monitoring42, monitors network flows dynamically, gathers information, and applies security measures depending on traffic behavior. The script initially defines a collection of classifiers for machine learning and creates a data frame to gather and analyze network information. It has a monitoring feature that periodically asks network switches for flow statistics. These are used afterward to gather detailed information on network traffic, such as the number of bytes and packets transported, the length of the flow, and different TCP segment flags. The script also computes certain parameters that are essential for differentiating between normal and abnormal behaviors, such as protocol types and service-specific traffic (HTTP, HTTPS, DNS, etc.). After the data is gathered, methods for preprocessing like scaling and imputation are used to get it ready for machine learning analysis. Labeled data is used to train the machine learning component to identify patterns linked to various kinds of network assaults. The script takes preventive measures by dropping malicious traffic when it detects an attack based on the patterns of traffic and the predictions made by the machine learning models. This involves generating an OpenFlow rule that aligns with the attributes of the malicious flow, such as IP addresses and TCP flags. Other OpenFlow rules can also be applied, targeting various parameters like Ethernet type, VLAN ID, IP protocol, and ports for TCP/UDP, along with ICMP, ARP details, and more. Possible actions range from directing packets to a designated port, assigning queue IDs, altering packet attributes, to managing VLAN and MPLS tags, and modifying TTL values. The install_drop_flow function creates a flow modification command without any specified actions, effectively instructing the switch to block packets that meet the identified criteria. This command is then transmitted to the pertinent switch, which implements the rule in its flow table.

4.1.2 Mininet

Mininet is a network emulator that uses Linux namespaces for process virtualization to generate a realistic virtual network on a single system, enabling fast simulation of a large number of network nodes [58]. It is very advantageous for researchers and developers to create and test network applications and protocols in a controlled environment. Using Python script in Mininet we were able to create a custom network topology class which is derived from the Topo class.

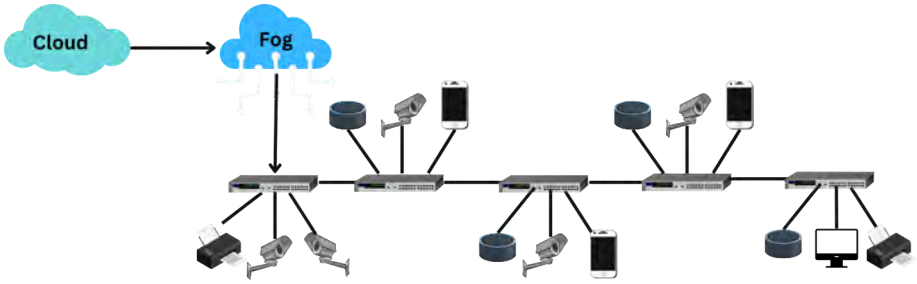


Figure 4.1: Topology

The topology 4.1 consists of numerous hosts and switches, with each host allotted a portion of CPU resources, a unique MAC address, and an IP address. For advanced network management and control, the switches are set up by using the OpenFlow1.3 protocol. The code develops network topology by defining the network connections between these nodes using a series of links between hosts and switches. The program is started by establishing a connection to a remote controller and instantiating the custom topology using the Mininet class. After that, the network is turned on and users can communicate with it using the Mininet CLI. This enables real-time configuration, testing, and evaluation of network behaviors as well as the effects of various protocols and applications. It offers a practical experience that emphasizes experimentation and learning in a dynamically controllable setting.

4.2 Experimental Results

We used customized programs and special scripts to generate different kinds of network traffic, including normal traffic and attacks like DDoS, Mirai botnet, and ARP spoofing. This traffic data was gathered by the Ryu controller, which also obtained relevant metrics like as packet counts, byte counts, and flow time. Preprocessing was done on the gathered data, followed by feature extraction to find important indicators of network behavior, normalization to standardize feature values, and treatment of missing values by imputation or elimination. After the dataset was analyzed, it was used for testing, and the CICIoT2023 trainset was used. A number of machine learning models, such as multilayer perceptrons (MLP) and decision trees, were trained and tested.

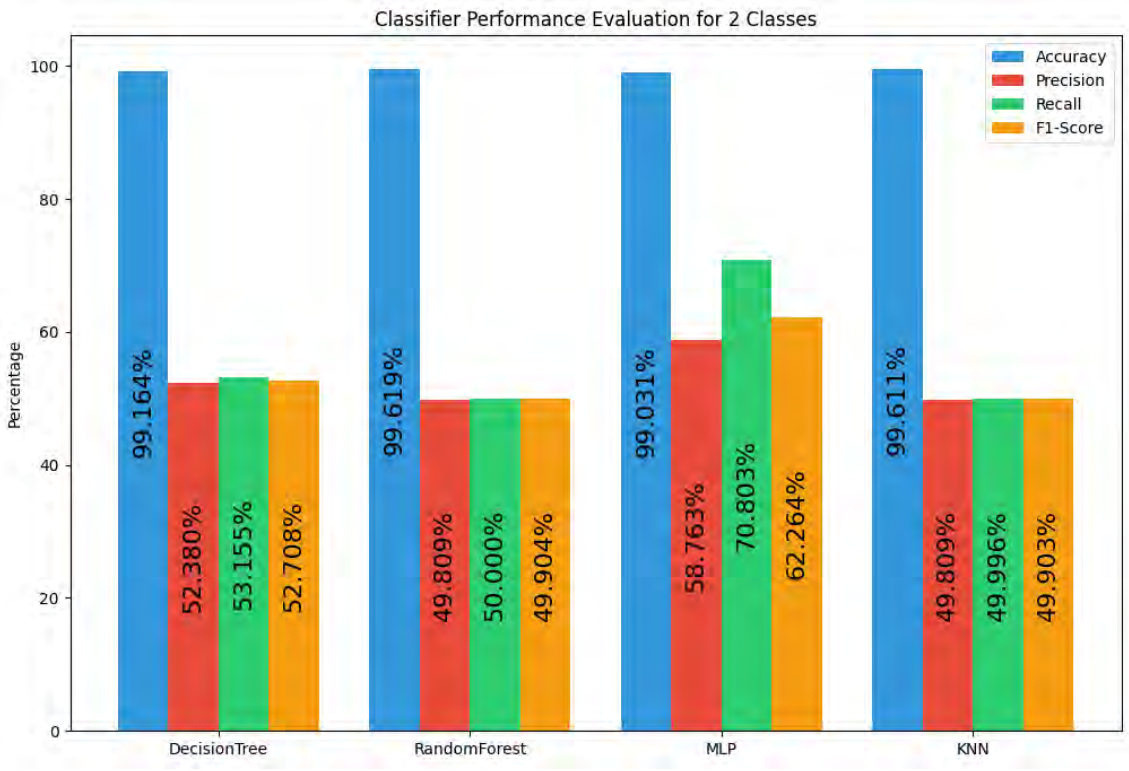
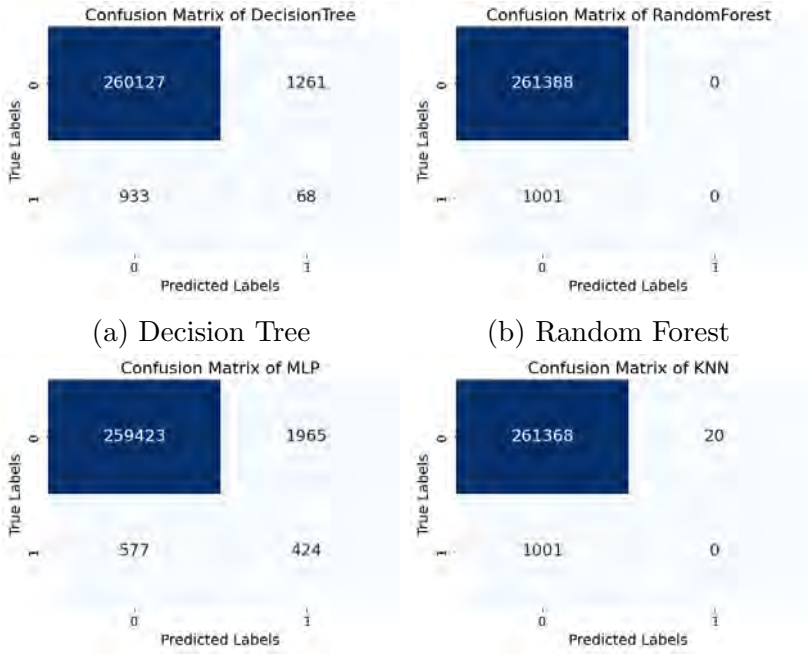


Figure 4.2: Performance comparison of different models of result



(a) Decision Tree (b) Random Forest

(c) MLP (d) KNN

Figure 4.3: Confusion matrix of different models result

The outcome of the machine learning models figure 4.2 which are used to check network traffic data collected with a Ryu controller propose some important changes along with gaps regarding how the system is now set up to identify network issues.

The accuracy, precision, recall, and F1 scores of each model—Random Forest, Decision Tree, MLP, and KNN—alter. These metrics are crucial for evaluating the way that classification models perform in security-sensitive situations.
First of all, the models developed to be mostly precise at identifying the data into the appropriate categories (attack vs. benign) based on the high accuracy scores detected throughout most models, especially Random Forest and KNN. True positive rates indicate that these models are only so good at correctly recognizing actual attacks as random guessing, which is represented by the precision scores being near 0.5. This indicates that although the models successfully classify entities in general, they are not accurate in precisely identifying individual network attack cases.
The recall scores provide an additional level of understanding. The MLP model has a greater recall rate, indicating a comparatively higher level of effectiveness in recognizing the relevant instances of threats (also known as true positives). However as its lower precision score indicates, this occurs at the expense of more false positives. The F1 score is significantly greater for MLP, indicating that even with its limitations, it could continue to be the more successful model of all the models tested for verifying suspicions of network attacks in a mixed data environment such as the one gathered by the Ryu controller.

4.3 Experimental Findings

These results could be influenced by several factors:
Class Imbalance: In network security, this is a common issue where the amount of benign traffic is significantly greater than the amount of attack instances. This type of data may cause models to become biased in favor of the majority class, which would hinder the ability to recognize attacks against the minority class.
Feature Selection: It is possible that the selected features do not have the discriminating ability needed to distinguish between malicious and benign traffic. This problem could result in fail to detect small but significant changes in network activity that point to intrusions.
Overfitting of the Model: When a model performs well on training data but is unable to generalize to new, unknown data, it may be overfitted. This is shown by high accuracy combined with low precision and recall. This is especially troublesome in network security, because attack paths and behaviors change rapidity.
MLP Model Strengths: Higher recall and F1 scores for the Multilayer Perceptron (MLP) model demonstrated its capacity to identify more true positive assault cases. Although it generated more false positives due to its poorer precision, this trade-off is frequently justified in security conditions where failing to detect an attack might have a greater negative impact than incorrectly reporting innocuous traffic.
Decision Tree Model Balance: With moderate precision and recall scores, the Decision Tree model performed in a more balanced way. Its balance makes it a helpful tool for fog computing environments' real-time threat detection. Understanding the decision-making process is made easier by its interpretability, which is advantageous when it comes to modifying and enhancing security rules.
Need for Improved Detection Strategies: The results indicate the need for ongoing development of methods of detection. Robust security in fog computing systems requires periodic updates to training data with new patterns of attack,

researching advanced machine learning algorithms, and utilizing more sources of data from IoT devices.

Chapter 5

Discussion

5.1 CIA Maintenance

Confidentiality:
By ensuring that all potential risks are identified and mitigated, improved feature selection, constant advancement in detection methodologies, and improved detection of rare attacks all contribute to the identification and prevention of unauthorized access to confidential information. Sensitive data is shielded against unwanted access by unauthorized parties due to high recall in models like MLP, which ensures that the majority of possible threats are identified.
Integrity:
The integrity of data within the structure of smart cities is maintained by precisely recognizing and reacting to threats, ensuring that data isn't interfered with or altered. Maintaining the integrity of the data is handled and preserved is ensured by detecting and mitigating defects that may point to data manipulation.
Availability:
The availability of smart city facilities is maintained by the effective detection of attacks, which stops them from growing more serious and disrupting normal operations. By preventing overfitting, models are ensured to be capable of detecting and mitigating new forms of attack, hence ensuring service availability. Service availability is maintained by minimizing false positives while retaining good recall, which ensures that benign traffic is not overly stopped.

5.2 Comparison with existing studies

Table 5.1: A qualitative comparison with existing studies

Research study	Use of Algorithm	Focus on Attack	Platform Used/ Simulated	Detecting Attack	Mitigating Attack	Multiple Attack Detection
[16]	Various Security Mechanisms	Not specific	Fog Computing, IoT	Yes	No	No
[55]	ANFIS, SDN	TCP SYN Flood	Fog Computing, SDN	Yes	Yes	No
[30]	Anomaly-based Detection, k-NN	DDoS	Fog Computing	Yes	Yes	No
[11]	VPN, Challenge-Response Authentication	Man in the Middle (MitM), DDoS	Fog Computing	Yes	Yes	No
[37]	Deep Reinforcement Learning	Reconnaissance	Cloud Platforms	Yes	Yes	No
[31]	Visualization, Machine Learning	DNS Tunneling, Data Exfiltration	Custom DNS Proxy Server	Yes	Yes	No
[43]	Federated Learning	Botnet	IoT Networks	Yes	Yes	No
[12]	Intrusion Detection and Prevention	Man in the Middle (MitM)	OMNET++	Yes	Yes	No
Our Proposed Model	Advanced Machine Learning Techniques	Various Security Threats	Fog Computing in IoT-Enabled Smart Cities	Yes	Yes	Yes

The table 5.1 summarizes a number of research publications along with our proposed model that address security measures for various cyberattack scenarios in fog computing and IoT environments. Each section specifies the attack type addressed, the technique that was employed, the platform that was used for simulation or implementation, and if the article deals with attack detection, attack mitigation, or multiple attack detection. The applications include IoT networks, fog, and cloud computing systems, and the methodologies range from machine learning approaches to deep reinforcement learning.

5.3 Limitation

1. The observed disparity between high accuracy and lower precision/recall scores shows there are problems with generalization between various data subsets, emphasizing a constraint in the model's capacity to function reliably in a range of network environments.

2. The models can fail to identify complicated patterns in the data which suggest highly developed attack vectors due to their limitations with the existing set of features.

3. Network traffic is dynamic by nature and is subject to large fluctuations. Static models might not be able to adjust to newer attack types or modifications in safe traffic patterns quickly enough.

4. Real-time analysis and mitigation might prove difficult, which could delay the detection and response to attacks, based on the computational complexity of the models and the volume of traffic.

Chapter 6

Conclusion and Future work

The importance of Smart Cities in tackling the problems caused by urbanization and technological progress is explored in this thesis. By allowing real-time data processing, resource optimization, and improved citizen services, it demonstrates how the combination of IoT and fog computing has dramatically affected the creation of smart cities. It also highlights the interconnection and expanding complexity of these systems, making them vulnerable to various security risks. To lay the groundwork for understanding the relationships between Smart Cities, IoT, and Fog Computing, the research starts by giving a thorough review of each. At that moment, the discussion turns to the security characteristics of Fog Computing in IoT-enabled Smart Cities. The key security issues are data privacy breaches, unauthorized access, network congestion, and related issues including Distributed Denial of Service (DDoS) assaults, virus spread, and physical manipulation. The thesis investigates alternative remedies for these problems, such as robust authentication techniques, encryption protocols, intrusion detection systems, and the use of blockchain technology. The ultimate objective is to guarantee the resilience, integrity, and privacy of these creative urban ecosystems. This work facilitates the development of multiple future works i.e. the implementation of different attack mitigation strategies in the Fog Computing environment. In conclusion, the thesis investigates the security issues generated by the use of fog computing in the environment of IoT-driven Smart Cities and offers proactive ways to safeguard both their functionality and security. Our future research should be focused on combining more discriminative features which will be able to better analyze between benign and malicious traffic. Advanced machine learning techniques, such as deep learning models can be utilized for the improvement in detection accuracy and robustness along with overcoming class imbalance and can be developed as a seamless integration model with existing security systems.

References

[1] K. Lee, D. Kim, D. Ha, U. Rajput, and H. Oh, "On security and privacy issues of fog computing supported internet of things environment," in *2015 6th International Conference on the Network of the Future (NOF)*, IEEE, 2015.

[2] Y. Wang, Y. Li, S. Liu, and B. Li, "Fabrication of chitin microspheres and their multipurpose application as catalyst support and adsorbent," en, *Carbohydr. Polym.*, vol. 120, pp. 53–59, 2015.

[3] M. Ahmad, M. B. Amin, S. Hussain, B. H. Kang, T. Cheong, and S. Lee, "Health fog: A novel framework for health and wellness applications," en, *J. Supercomput.*, vol. 72, no. 10, pp. 3677–3695, 2016.

[4] M. Barcelo, A. Correa, J. Llorca, A. M. Tulino, J. L. Vicario, and A. Morell, "IoT-cloud service optimization in next generation smart environments," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 4077–4090, 2016.

[5] S. Chakraborty, S. Bhowmick, P. Talaga, and D. P. Agrawal, "Fog networks in healthcare application," in *2016 IEEE 13th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, IEEE, 2016.

[6] A. Monteiro, H. Dubey, L. Mahler, Q. Yang, and K. Mankodiya, "Fit: A fog computing device for speech tele-treatments," in *2016 IEEE International Conference on Smart Computing (SMARTCOMP)*, IEEE, 2016.

[7] M. Antonakakis, T. April, M. Bailey, *et al.*, "Understanding the mirai botnet," 2017. [Online]. Available: https://www.usenix.org/conference/usenixsecurity17/technical - sessions / presentation / antonakakis % 7D , booktitle % 20 = %20 % 7BProceedings%20of%20the%2026th%20USENIX%20Security%20Symposium%7D.

[8] S. Khan, S. Parkinson, and Y. Qin, "Fog computing security: A review of current applications and security solutions," en, *J. Cloud Comput. Adv. Syst. Appl.*, vol. 6, no. 1, 2017.

[9] M. Mukherjee, R. Matam, L. Shu, *et al.*, "Security and privacy in fog computing: Challenges," *IEEE Access*, vol. 5, pp. 19 293–19 304, 2017.

[10] T. Nguyen Gia, M. Jiang, V. K. Sarker, *et al.*, "Low-cost fog-assisted healthcare IoT system with energy-efficient sensor nodes," in *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, IEEE, 2017, pp. 1765–1770.

[11] S. Alharbi, P. Rodriguez, R. Maharaja, P. Iyer, N. Bose, and Z. Ye, "FOCUS: A fog computing-based security system for the internet of things," in *2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, IEEE, 2018.

[12] F. Aliyu, T. Sheltami, and E. M. Shakshuki, "A detection and prevention technique for man in the middle attack in fog computing," en, *Procedia Comput. Sci.*, vol. 141, pp. 24–31, 2018.

[13] L. Cui, G. Xie, Y. Qu, L. Gao, and Y. Yang, "Security and privacy in smart cities: Challenges and opportunities," *IEEE Access*, vol. 6, pp. 46 134–46 145, 2018.

[14] R. Mahmud, R. Kotagiri, and R. Buyya, "Fog computing: A taxonomy, survey and future directions," in *Internet of Things*, Singapore: Springer Singapore, 2018, pp. 103–130.

[15] B. Negash, T. N. Gia, A. Anzanpour, *et al.*, "Leveraging fog computing for healthcare IoT," in *Fog Computing in the Internet of Things*, Cham: Springer International Publishing, 2018, pp. 145–169.

[16] J. Ni, K. Zhang, X. Lin, and X. Shen, "Securing fog computing for internet of things applications: Challenges and solutions," *IEEE Commun. Surv. Tutor.*, vol. 20, no. 1, pp. 601–628, 2018.

[17] A. M. Rahmani, T. N. Gia, B. Negash, *et al.*, "Exploiting smart e-health gateways at the edge of healthcare Internet-of-Things: A fog computing approach," en, *Future Gener. Comput. Syst.*, vol. 78, pp. 641–658, 2018.

[18] P. Zhang, J. K. Liu, F. R. Yu, M. Sookhak, M. H. Au, and L. Wang, "A survey on Access Control in fog Computing," *IEEE Communications Magazine*, vol. 56, no. 2, pp. 144–149, Feb. 2018. DOI: 10.1109/mcom.2018.1700333.

[19] K. H. Abdulkareem, M. A. Mohammed, S. S. Gunasekaran, *et al.*, "A review of fog computing and machine learning: Concepts, applications, challenges, and open issues," *IEEE Access*, vol. 7, pp. 153 123–153 140, 2019.

[20] M. A. Khan, "Fog computing in 5G enabled smart cities: Conceptual framework, overview and challenges," in *2019 IEEE International Smart Cities Conference (ISC2)*, IEEE, 2019.

[21] V. Vijayakumar, D. Malathi, V. Subramaniyaswamy, P. Saravanan, and R. Logesh, "Fog computing-based intelligent healthcare system for the detection and prevention of mosquito-borne diseases," en, *Comput. Human Behav.*, vol. 100, pp. 275–285, 2019.

[22] E. Zaheri Abdevand, Islamic Azad University, Ashtian, Iran, S. Ghanbari, Z. Umarova, Z. Iztayev, and South Kazakhstan State University, Shymkent, Kazakhstan, "Introducing a new intrusion detection method in the sdn network to increase security using decision tree and neural network," *Azerbaijan J. High Perform. Comput.*, vol. 2, no. 2, pp. 97–112, 2019.

[23] L. Zhou, H. Guo, and G. Deng, "A fog computing based approach to DDoS mitigation in IIoT systems," en, *Comput. Secur.*, vol. 85, pp. 51–62, 2019.

[24] D. e. D. Abou-Tair, S. Büchsenstein, and A. Khalifeh, "A fog computing-based framework for privacy preserving IoT environments," en, *Int. Arab J. Inf. Technol.*, vol. 17, no. 3, pp. 306–315, 2020.

[25] W. Almobaideen and M. Altarawneh, "Fog computing: Survey on decoy information technology," en, *Int. J. Secur. Netw.*, vol. 15, no. 2, p. 111, 2020.

[26] S. K. Gupta, S. Vanjale, S. Rasal, and M. Vanjale, "Securing IoT devices in smart city environments," in *2020 International Conference on Emerging Smart Computing and Informatics (ESCI)*, IEEE, 2020.

[27] C. Kelly, N. Pitropakis, S. McKeown, and C. Lambrinoudakis, "Testing And Hardening IoT Devices Against the Mirai Botnet," *International Conference on Cyber Security And Protection Of Digital Services (Cyber Security)*, Jun. 2020. DOI: 10.1109/cybersecurity49315.2020.9138887. [Online]. Available: https://doi.org/10.1109/cybersecurity49315.2020.9138887.

[28] M. P. Maharani, P. Tobianto Daely, J. M. Lee, and D.-S. Kim, "Attack detection in fog layer for IIoT based on machine learning approach," in *2020 International Conference on Information and Communication Technology Convergence (ICTC)*, IEEE, 2020.

[29] B. Susilo and R. F. Sari, "Intrusion detection in IoT networks using deep learning algorithm," en, *Information (Basel)*, vol. 11, no. 5, p. 279, 2020.

[30] M. A. Lawal, R. A. Shaikh, and S. R. Hassan, "A DDoS attack mitigation framework for IoT networks using fog computing," en, *Procedia Comput. Sci.*, vol. 182, pp. 13–20, 2021.

[31] Y. F. Mohammed, "Network-based detection and prevention system against DNS-based attacks," Ph.D. dissertation, University of Arkansas, Fayetteville, 2021.

[32] M. Tawfik, N. M. Al-Zidi, B. Alsellami, A. M. Al-Hejri, and S. Nimbhore, "Internet of things-based middleware against cyber-attacks on smart homes using software-defined networking and deep learning," in *2021 2nd International Conference on Computational Methods in Science & Technology (ICCMST)*, IEEE, 2021.

[33] A. Zaheer, M. Z. Asghar, and A. Qayyum, "Intrusion detection and mitigation framework for SDN controlled IoTs network," in *2021 IEEE 18th International Conference on Smart Communities: Improving Quality of Life Using ICT, IoT and AI (HONET)*, IEEE, 2021.

[34] M. D. T. Bennet, M. P. S. Bennet, and D. Anitha, "Securing smart city networks - intelligent detection of DDoS cyber attacks," in *2022 5th International Conference on Contemporary Computing and Informatics (IC3I)*, IEEE, 2022.

[35] "Fog computing with IoT device's data security management using density control weighted election and extensible authentication protocol," *Int. J. Intell. Eng. Syst.*, vol. 15, no. 1, 2022.

[36] M. Lata and V. Kumar, "Fog computing infrastructure for smart city applications," in *Recent Advancements in ICT Infrastructure and Applications*, Singapore: Springer Nature Singapore, 2022, pp. 119–133.

[37] H. Li, Y. Guo, S. Huo, H. Hu, and P. Sun, "Defensive deception framework against reconnaissance attacks in the cloud with deep reinforcement learning," en, *Sci. China Inf. Sci.*, vol. 65, no. 7, 2022.

[38] S. C. Mana, B. Keerthi Samhitha, D. Deepa, and R. Vignesh, "Analysis on application of fog computing in industry 4.0 and smart cities," in *Lecture Notes on Data Engineering and Communications Technologies*, Singapore: Springer Singapore, 2022, pp. 87–105.

[39] S. Barrett, B. Boswell, and G. Dorai, "Exploring the vulnerabilities of IoT devices: A comprehensive analysis of mirai and bashlite attack vectors," in *2023 10th International Conference on Internet of Things: Systems, Management and Security (IOTSMS)*, IEEE, 2023.

[40] E. Bertino, H. Lee, M. Huang, *et al.*, "A pro-active defense framework for IoT systems," in *2023 IEEE 9th International Conference on Collaboration and Internet Computing (CIC)*, IEEE, 2023.

[41] V. Božić, *Applications of fog computing for smart sensors*, 2023.

[42] M. Burhan, H. Alam, A. Arsalan, *et al.*, "A comprehensive survey on the cooperation of fog computing paradigm-based IoT applications: Layered architecture, real-time security issues, and solutions," *IEEE Access*, vol. 11, pp. 73 303–73 329, 2023.

[43] F. L. de Caldas Filho, S. C. M. Soares, E. Oroski, *et al.*, "Botnet detection and mitigation model for IoT networks using federated learning," en, *Sensors (Basel)*, vol. 23, no. 14, p. 6305, 2023.

[44] A. Heidari and M. A. Jabraeil Jamali, "Internet of things intrusion detection systems: A comprehensive review and future directions," en, *Cluster Comput.*, vol. 26, no. 6, pp. 3753–3780, 2023.

[45] B. Hubert, *Spoofing DNS with fragments*, May 2023. [Online]. Available: https://blog.powerdns.com/2018/09/10/spoofing-dns-with-fragments.

[46] N. Kaliya and D. Pawar, "Unboxing fog security: A review of fog security and authentication mechanisms," en, *Computing*, 2023.

[47] M. Kumar and S. K. Dubey, "Network intrusion detection for IoT devices using deep learning," in *2023 10th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)*, IEEE, 2023.

[48] C. Mongare, "Securing the internet of things (iot) devices," vol. 1, May 2023.

[49] P. Mubashir, R. khan, and M. Farooq, "Stout implementation of firewall and network segmentation for securing iot devices," *Indian Journal of Science and Technology*, vol. 16, pp. 2609–2621, Sep. 2023.

[50] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. A. Ghorbani, "Ciciot2023: A real-time dataset and benchmark for large-scale attacks in iot environment," *Sensors*, vol. 23, no. 13, 2023, ISSN: 1424-8220. DOI: 10.3390/s23135941. [Online]. Available: https://www.mdpi.com/1424-8220/23/13/5941.

[51] I. H. Sarker, A. I. Khan, Y. B. Abushark, and F. Alsolami, "Internet of things (IoT) security intelligence: A comprehensive overview, machine learning solutions and research directions," en, *Mob. Netw. Appl.*, vol. 28, no. 1, pp. 296–312, 2023.

[52] K. Singh and N. Singh, "Analysis of IoT attack detection and mitigation," in *2023 International Conference on Artificial Intelligence and Smart Communication (AISC)*, IEEE, 2023.

[53] A. Srivastava and U. Jain, "Securing the future of IoT: A comprehensive framework for real-time attack detection and mitigation in IoT networks," in *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, IEEE, 2023.

[54] A. N. Alvi, B. Ali, M. S. Saleh, M. Alkhathami, D. Alsadie, and B. Alghamdi, "Secure computing for fog-enabled industrial IoT," en, *Sensors (Basel)*, vol. 24, no. 7, p. 2098, 2024.

[55] R. Bensaid, N. Labraoui, A. A. Abba Ari, *et al.*, "Toward a real-time TCP SYN flood DDoS mitigation using adaptive neuro-fuzzy classifier and SDN assistance in fog computing," en, *Secur. Commun. Netw.*, vol. 2024, pp. 1–20, 2024.

[56] A. Khan, *Reconnaissance in cyber security*, Apr. 2024. [Online]. Available: https://intellipaat.com/blog/reconnaissance-in-cyber-security/.

[57] K. S. Rani, G. Parasa, D. Hemanand, *et al.*, "Implementation of a multi-stage intrusion detection systems framework for strengthening security on the internet of things," *MATEC Web Conf.*, vol. 392, p. 01 106, 2024.

[58] M. Project, *MiniNet: An instant virtual network on your laptop (or other PC) - MiniNet*. [Online]. Available: http://mininet.org/.