DESIGNING A SECURE ACCESS NETWORK TECHNIQUE TO MITIGATE THE IMPACT OF DISTRIBUTED DENIAL OF SERVICE ATTACKS ON MOBILE EDGE COMPUTING

by

MMALOTO JOSEPH RAMALEBA

Master's Dissertation

Submitted in fulfilment of the requirements for the degree of

Master of Science

in

COMPUTER SCIENCE

in the

FACULTY OF SCIENCE AND AGRICULTURE,

(SCHOOL OF MATHEMATICAL AND COMPUTER SCIENCES)

at the

UNIVERSITY OF LIMPOPO

SUPERVISOR: Professor M. VELEMPINI

2024

DECLARATION

I, Mmaloto Joseph Ramaleba, declare that the **Designing a secure access network technique to mitigate the impact of distributed denial of service attacks on mobile edge computing** dissertation hereby submitted to the University of Limpopo, for the degree of Master of Science (Computer Science) has not previously been submitted by me for a degree at this or any other university; that it is my work in design and in execution, and that all material contained herein has been duly acknowledged.

Ramaleba MJ (Mr)
Surname, Initials (title)

18 November 2024
Date

Signature: MRamaleba.

ACKNOWLEDGMENTS

I would like to begin by giving honour and gratitude to the Almighty God, creator of heaven and earth, for granting me life, wisdom, and strength throughout this research journey and beyond. To Him be all glory and honour, now and forever. Without His guidance, none of this would have been possible.

I am also deeply grateful to my supervisor, Professor Mthulisi Velempini, whose guidance, dedication, and encouragement have been instrumental in bringing out the best in me throughout this process.

Lastly, I extend my heartfelt thanks to my University of Limpopo ICT eLearning colleagues, family, and friends for their unwavering support throughout this journey. A special acknowledgment goes to my mother, Mmaphetli Patience Ramaleba, whose steadfast prayers and spiritual guidance have been a source of strength. Her unwavering support sustained me during the research period, and I deeply appreciate and recognize the profound impact it has had on my journey.

DEDICATION

This work is dedicated to the loving memory of my late grandmother, Makwela Manti Salome, and my late grandfather, Makwela Eddie. Their unwavering love, wisdom, and guidance have been a source of strength and inspiration throughout my life. Though they are no longer with us, their legacy continues to shape the person I am today. I honour their memory with deep gratitude, and this accomplishment is a testament to the values and principles they instilled in me.

May their souls rest in eternal peace.

ABSTRACT

Mobile Edge Computing (MEC) enhances networking technology by bringing computing and storage close to end users. However, this advancement introduces security challenges, particularly its susceptibility to Distributed Denial of Service (DDoS) attacks that threaten network availability and performance in MEC.

This study tackles these security vulnerabilities by introducing AdaptiveMECShield, an anomaly-based detection system designed to mitigate DDoS attacks in MEC environments. AdaptiveMECShield integrates several key features to efficiently detect and respond to malicious activity, load balancing, adaptive thresholding, windowing strategies, and real-time traffic monitoring using Software Defined Networking (SDN) controllers. The system dynamically adjusts trust values for network nodes based on packet header information, enabling timely and accurate detection of abnormal traffic. The adaptive thresholding mechanism allows the system to effectively respond to varying traffic patterns, minimizing false positives and negatives while maintaining high detection accuracy. Rigorous simulations demonstrated that AdaptiveMECShield outperformed the Self-Organising Map (SOM) scheme in terms of detection accuracy, false positive and negative rates, detection times, and resource consumption efficiency.

In conclusion, AdaptiveMECShield significantly improves the security of MEC systems against DDoS attacks. Its proactive and adaptive nature ensures enhanced security and resilience of edge computing and IoT networks, contributing to uninterrupted availability and improved performance of critical services.

Keywords: Mobile Edge Computing (MEC), Distributed Denial of Service (DDoS) Attacks, AdaptiveMECShield, Anomaly-Based Detection, Load Balancing, Adaptive Thresholding, Windowing, Network Security, Software Defined Network (SDN), Resource Utilisation

Table of Contents

List of Figures

List of Tables

List of Abbreviations

IoT	Internet of Things
MEC	Mobile Edge Computing
DDoS	Distributed Denial of Service
SDN	Software Defined Network
SOM	Self-Organising Map
CPU	Central Processing Unit
Tbps	Terabits per second
AR	Augmented Reality
VR	Virtual Reality
ADS	Anomaly Detection System
PDR	Packet Delivery Ratio

List of Notations

Notation	Meaning
γ	The weight scale factor for header fields' effect
L	Payload length header field
X	Payload length header field list
S	Source IP header field
$\vec{sl}$	Source IP header field list
d	Destination IP header field
$\vec{dl}$	Destination IP header field list
sp	Source port header field
$\vec{spl}$	Source port header field list
lth	Payload length header field threshold
sth	Source IP header field threshold
dth	Destination IP header field threshold
spth	Source port header field threshold
m_{th}	The maximum threshold for user
N	Number of packets reaching network in this time slot
nm	Maximum number of packets reaching network in a normal state
$\vec{tv}$, tv_{max}, tv_{min}	Trust value list, max, and min value
Si	SDN network user with index i
α	Forgetting factor, $\alpha \in (0,1)$
θi	Abnormal threshold for user i
T_{min}	Threshold trust for malicious users
P_i	Total request form user S_i
N_q	Number of priority buffer queues
L_{max}	Total buffer length of N_q buffer queues
Lj	Length of the j^{th} buffer queue
β	Weight scale factor for buffer queues

Notation	Meaning
λ_l	Normal Traffic Arrival Rate
λ_a	Attack Arrival Rate
t_W	Window size
H	Number of Hosts
θ_m	Max value system threshold
θ_u	Max value user threshold

CHAPTER 1: INTRODUCTION

1. Introduction

The ever-expanding technological world exerts significant pressure on organisations and individuals to use technology efficiently and effectively. Individuals and organisations all over the world are increasingly relying on technology. This has resulted in an increase in traffic on communication networks, posing significant challenges in terms of connectivity, computing, and security. In September 2016, for example, many Internet of Things (IoT) devices infected with the Mirai malware flooded a French web host with about one terabit per second (Tbps) of distributed denial of service (DDoS) traffic [1].

The traditional cloud computing paradigm, with its centralised data processing, is not well-suited to meet the requirements of emerging applications, particularly those associated with the IoT [2]. IoT devices generate large amounts of data at the network edge, requiring real-time services and low-latency responses. To address these limitations, mobile edge computing (MEC) has emerged as an exciting new technology with the potential to transform how we distribute and consume digital services. MEC is a technique that addresses these disruptions by bringing computing and storage closer to mobile users, enabling high bandwidth, low latency applications and high-quality services [3], [4]. By processing data closer to the source, MEC reduces network congestion, improves response times, and enhances user experience. MEC, on the other hand, is subject to DDoS attacks, which can disable its capacity to supply critical services [5].

DDoS attacks pose a significant risk to MEC since they can quickly overwhelm its limited resources. DDoS attacks involve attackers flooding a target system with traffic from various sources, rendering it inaccessible to genuine users. This has the potential to compromise MEC-enabled services such as augmented reality/virtual reality (AR/VR), real-time video streaming, and IoT applications [6].

Because of the increased usage of new technologies in MEC, there is a pressing need to develop effective DDoS mitigation strategies. This research focuses on developing new techniques for detecting, preventing, and mitigating DDoS attacks in the context of MEC's specific characteristics, such as its distributed architecture and limited resources.

1.1 Problem Statement

The analogy of traffic congestion on a busy highway during peak hours, exemplifies the issue that MEC networks face. Just as an influx of cars can exceed the road's capacity, DDoS assaults take advantage of MEC networks' limited resources and distributed architecture across multiple locations, posing a substantial risk to their performance and security. The traditional DDoS mitigation techniques are ineffective against DDoS attacks in MEC networks because they are reactive and readily overwhelmed by large-scale attacks [7][8]. The use of distributed attacks such as botnets simplifies the commencement of massive attacks by attackers because they use the power of many devices to launch a distributed attack [9]. Figure 1.1 depicts a Distributed Denial of Service (DDoS) attack, where multiple compromised computer systems, collectively known as botnets, launch a coordinated assault on a target. The excessive number of requests generated by these botnets causes the intended system to slow down or become completely inaccessible, disrupting normal activities and preventing legitimate users from accessing the service. This coordinated attack seeks to drain the target's resources, causing downtime or significant service degradation.

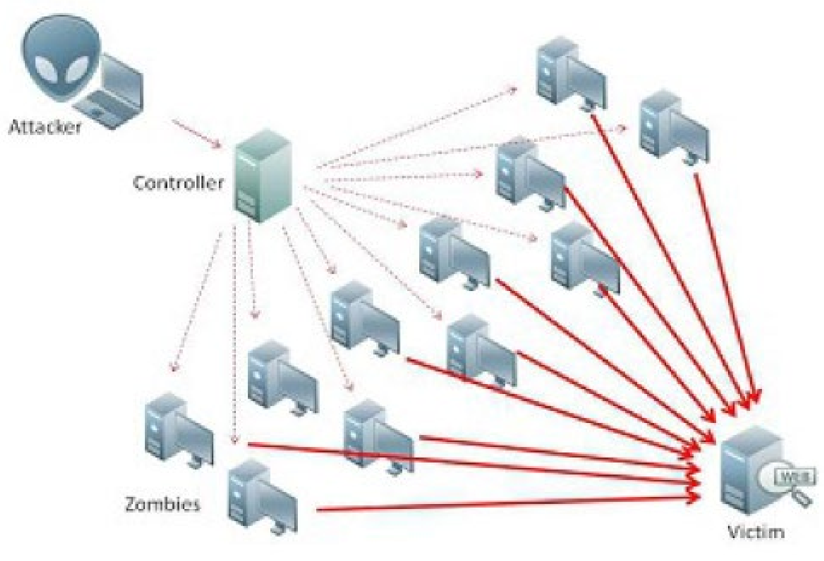

Figure 1.1 A DDoS Attack

Since its start, many DDoS mitigating techniques have been proposed and employed; however, the increasing complexity needs to call for developing technologies that could counteract these attacks and improve the technology's performance. Traditional DDoS mitigation techniques typically involve distributing defence resources based on demand. However, this approach is ineffective because it is reactive and can be easily overwhelmed by large-scale attacks[10]. Nevertheless, a more promising approach is to create dynamic defence resources that involve proactively creating and deploying defence resources based on predicted attack patterns and trends.

1.2 Objectives

This research aims to design a dynamic and collaborative DDoS mitigation technique for mobile edge computing, enabling reliable and secure delivery of MEC services to mobile users. This aim is divided into the following objectives, designed to:

- **1.2.1** Explore the factors and impact of distributed denial of service in MEC.
- **1.2.2** Develop a new technique for dynamically creating defence resources collaboration in MEC networks.
- **1.2.3** Evaluate the effectiveness of the proposed techniques in mitigating DDoS attacks in MEC networks.
- **1.2.4** Implement the proposed techniques in real-world MEC networks.

1.3 Research Questions

The following research questions are answered throughout the study to help develop and implement the best mitigation strategies.

- **1.3.1** How can the impact of DDoS be mitigated in MEC?
- **1.3.2** How can we create and deploy dynamic defence resources to protect critical MEC services from DDoS attacks?
- **1.3.3** How can we evaluate the effectiveness of new DDoS mitigation techniques in MEC?
- **1.3.4** How can we implement new DDoS mitigation techniques in real-world MEC networks?
- **1.3.5** How can we identify and prioritise critical MEC services?
- **1.3.6** What are the most effective ways to detect DDoS attacks in MEC networks?

1.4 Scientific Contribution

“Designing a Secure Access Network Technique to Mitigate the Impact of Distributed Denial of Service Attacks on Mobile Edge Computing” explicates the overall goal of this study. The study begins by exploring the causes and consequences of DDoS assaults on MEC, which is critical for understanding the issue and driving the design of the mitigation architecture. It then focuses on fostering dynamic defense resource coordination within MEC networks, ensuring the establishment of secure procedures for protecting essential services. The mitigation technique's efficacy is dependent on identifying and prioritizing key MEC services, as well as properly detecting DDoS attacks. The study also assesses the effectiveness of various strategies to ensure they meet the required security criteria. The availability of services and resources guarantees that mobile edge cloud providers fulfil the required uptime, which might be hampered by DDoS attacks.

The study, therefore, emphasises service availability in edge computing by tackling the most serious security challenge, a DDoS attack, which exerts an impact on the availability of services and resources provided by the mobile edge provider to users [11]. The study presents technique to mitigate against DDoS attacks in mobile edge computing.

1.4.1 Contribution 1:

By studying the packets flooding the network, this study proposes a change-point monitoring strategy for detecting DDoS flooding attacks on edge cloud services. The approach takes advantage of the fact that most DDoS attacks are automated and follow similar patterns. As a result, they may be separated from normal traffic when closely inspected.

1.4.2 Contribution 2:

Because of the volume of traffic that must be handled during a DDoS attack in edge computing, this study proposes a DDoS detection and mitigation strategy in MEC. The scheme is designed to detect DDoS attacks early and mitigate their impact on network performance. The effectiveness and performance of the early detection MEC-DDoS scheme in mitigating the impact of DDoS is evaluated. This study focuses on enhancing the security performance of emerging technologies and networks, respectively.

1.5 Overview

There is a rapidly growing reliance on technology in both individual and organisational settings, and which has led to the development of new paradigms in networking infrastructure such as edge computing, which provides computing and storage closer to users and the edge of the network. The increasing use of online applications suggests that businesses and institutions rely on IoT technologies now more than ever before. However, the widespread use of these technologies has led to various security compromises and system vulnerabilities. Edge computing has emerged as a distinct paradigm capable of improving IoT performance, serving as an intermediary between edge devices and the cloud [12].

This study also highlights the significant security risks posed by DDoS attacks in the context of MEC and proposes designing and evaluating a secure access network mitigation technique to address the impact of DDoS attacks in MEC. The objectives of the research project include exploring the factors and impact of DDoS in MEC, investigating design techniques for secure access networks based in MEC, analysing key challenges of DDoS in MEC, and designing and evaluating the effectiveness of the access network scheme in MEC. The study proposes two contributions: a change-point monitoring strategy for detecting DDoS flooding attacks on edge cloud services and a DDoS detection and mitigation strategy in MEC. Overall, the study aims to enhance the security performance of emerging technologies and networks and contribute to MEC security.

1.6 Conclusion

In a digital world, reliance on technology is increasing which is a challenge to network infrastructures notably in terms of security. MEC has emerged as an important solution designed to meet the demands of real-time and low-latency applications by bringing computing resources closer to users. However, MEC's distributed architecture and limited resources make it vulnerable to DDoS attacks which compromise the availability of services and performance.

The purpose of this study was to counter MEC security challenges by presenting a dynamic and collaborative DDoS mitigation approach. Through investigating the causes and consequences of DDoS attacks in MEC, the study revealed the limits of standard reactive mitigation solutions and argued for proactive and dynamic defense strategies. The proposed strategies seek to improve DDoS detection and mitigation by using change-point monitoring and early detection systems, as well as to provide dependable service delivery and security for MEC networks.

The contributions of this research offer significant advancements in the field of MEC security, particularly in the detection and mitigation of DDoS attacks. The proposed strategies not only focus on protecting MEC services but also prioritize critical services and optimize network resource allocation. The research enhances the overall security performance of emerging technologies, providing a robust foundation for safeguarding future MEC networks against DDoS threats.

1.7 Dissertation review

The dissertation is organized into six chapters. Chapter 1 presents an overview of the study challenges, objectives, and rationale for designing the technique. Chapter 2 covers existing research on DDoS attacks and their impact on MEC systems. It investigates several DDoS mitigation solutions and identifies gaps in current research. Chapter 3 describes the proposed technique's design and architecture, including essential components such as load balancing, anomaly detection, and packet header analysis, as well as the simulation parameters and tools used to evaluate performance. Chapter 4 outlines the approach for evaluating the system's effectiveness, which includes the simulation environment, network topology, traffic generation process, and metrics for determining detection accuracy and system resilience. Chapter 5 analyses the simulation findings, comparing the proposed technique's performance to various DDoS attack patterns while emphasizing improvements in detection and mitigation over current techniques. Finally, Chapter 6 highlights the research's significant results and contributions, discusses limits, and makes recommendations for future research.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

DDoS attacks have become increasingly common, making it hard to take advantage of the capabilities of MEC. The attack compromises cloud availability, depletes bandwidth, computer resources, memory buffers and it degrades the performance of network protocols of the victims. This chapter examines and evaluates related work and ongoing research initiatives concerning DDoS attack countermeasures. Furthermore, this chapter emphasises inadequacies of the countermeasures in dealing with DDoS attacks.

2.2 Overview of MEC

MEC is a network architecture that enables computational resources and data storage which is located closer to the end-user which significantly enhance the performance and efficiency of applications that require low latency and high bandwidth. MEC enables real-time data processing from mobile devices, Internet of Things (IoT) devices, and other endpoints by extending cloud services to the edge of the network. This architectural change reduces the transmission distance which reduces latency and improves response times. This is important for applications like virtual reality (VR), augmented reality (AR), and driverless cars [13].

MEC is important in modern network technologies because it can handle the problems caused by the exponential increase in mobile data traffic and the increasing demand for real-time, high-quality services. They depend on centralized data centers, traditional cloud computing models which is a challenge to meet these expectations because of bandwidth and latency problems. By decentralizing processing and storage, MEC lessens these challenges and maximizes network resources while improving user experience [14].

The demand for better resource management and improved user experience in mobile networks is responsible for the development of MEC. MEC was first introduced in the context of 4G networks, but with the introduction of 5G technology which prioritizes high-capacity connections and low latency it has gathered significant traction. MEC is incorporated in many other fields, including industrial automation, smart cities, and healthcare. For instance, MEC supports applications in smart cities such as environmental monitoring and real-time traffic management, and by processing data locally, it facilitates telemedicine and remote patient monitoring in the healthcare industry [15].

2.3 Analysis of Techniques

Edge computing is a new technique that brings cloud computing capabilities closer to the network's edge. This decreases latency in the cloud computing paradigm and enables smooth interaction with a variety of application service providers and suppliers [9]. However, adopting the paradigm has come with several security vulnerabilities.

A DDoS attack is a severe threat that hinders legitimate network requests. Even though the first DDoS attack was documented in 1996, the complexity and sophistication of these attacks have grown over time. In the middle of the COVID-19 pandemic, a 2 Tbps attack on essential infrastructure, such as banking, was reported in mid-August 2020 [16]. The authors discussed different approaches for DDoS mitigation using Blockchain in varied domains [11]. Nonetheless, this strategy enables for the public data sharing among all nodes to ensure authenticity; also, the integration of blockchain into the MEC system is rather difficult due to the dynamic nature of channel circumstances and network loads. The dynamic nature of network conditions is a major obstacle as it can cause latency problems during times of high traffic, impeding the timely processing of transactions that are necessary for efficient mitigation. Another issue with scalability is that blockchain architectures is not able to manage the enormous number of transactions created during an attack, which could lead to bottlenecks. Moreover, depending on resource-intensive consensus methods like Proof of Work (PoW) which is not compatible with MEC settings' requirement for quick response. Potential solutions to these problems include creating network-adaptive consensus algorithms, using off-chain transactions to lighten the load on the main chain, and placing blockchain nodes closer to the edge for faster processing. Machine learning can be used for dynamic resource allocation and encouraging cooperative mitigation framework for real-time threat intelligence sharing can also enhance security.

According to the authors in [17], the implementation of software-defined networking (SDN) effectively detects and mitigates DDoS attacks. The suggested strategy, which takes advantage of SDN's programmability and centralised control, seeks to improve network security by protecting both the control and data planes from DDoS attacks. They utilised the SDN infrastructure to address the limitations of a sampling-based anomaly detection system (ADS) and gathered traffic flow statistics, which included several packet forwarding devices such as switches. In the data plane, detection is dependent on the IoT application, implying that the behaviour is predictable for a specific application. Its detection performance may be substantially impacted by the program user’s behaviour. In a flexible application, it may result in low detection accuracy. The technique includes the creation of robust algorithms that apply to both the control and data planes of SDN-based IoT networks. These techniques are intended to identify and mitigate DDoS attacks in real-time, protecting the integrity and availability of network resources and services. By combining security techniques on both planes, the approach provides comprehensive protection against all types of DDoS attacks on IoT devices and infrastructure.

Nonetheless, implementing secure techniques for both the data and control planes may result in greater computing demands, endangering the scalability of SDN-based IoT networks, especially in large installations with many IoT devices. The effort to secure both planes includes developing and deploying complex algorithms and protocols, which may exacerbate the complexity of network administration and configuration. To address scalability difficulties, optimisation procedures can be used to improve security algorithms while reducing computational costs. This could include improving resource allocation, precisely tweaking settings, or incorporating collaborative processing algorithms to spread the effort across multiple network components. Furthermore, in addressing these limitations, machine learning algorithms can be implemented, a hybrid detection system can be used, periodic traffic analysis can be performed, and behavioural analysis can be conducted [18].

The MECshield framework, proposed in [7] is an example of a MEC-based DDoS prevention system that leverages smart filters and MEC's computational power to enhance detection accuracy and reduce latency. A central controller monitors the smart filters and then localises each smart filter based on the attacking behaviour by incorporating pertinent settings for its self-organising map (SOM) component. The framework [7] has a limitation in determining the optimal placement of smart filters, which may lead to suboptimal filtering performance. One possible resolution could be to use an optimisation algorithm to determine the optimal placement of smart filters based on network traffic patterns.

The authors in [19] introduced an architecture that can identify and mitigate DDoS attacks of targeted intelligent infrastructures of a given city as close to their sources as possible. The scheme utilises an algorithm, which collects flow statistics from base stations and then defines an acceptable traffic limit. The scheme can only detect abnormal flow and establish an Anomaly Tree by analysing network traffic fluctuations. The Anomaly Tree-based approach [19] may result in false positives and false negatives. One possible resolution could be to use a more advanced machine learning-based algorithm to detect anomalies and set a dynamic threshold based on the current network conditions. The system can successfully distinguish between normal network behaviour and abnormal activity by dynamically modifying the threshold, avoiding both false positives and false negatives [20].

The authors in [21] investigated the effectiveness of cooperative defence schemes, which enabled defence resources to be distributed among MEC nodes. The authors proposed a global orchestration (GO) mechanism that uses an extensible estimation algorithm based on the mini-bucket elimination algorithm to fully exploit the cooperative defence freedom degrees, allowing defence resources to be deployed when required. The GO mechanism [21] assumes that MEC nodes are trustworthy, which may not always be the case. One possible resolution could be to incorporate a reputation system to evaluate the trustworthiness of MEC nodes before deploying defence resources.

The authors in [22] proposed a strategy called the co-operate defence (CODE) in the MEC environment designed for DDoS attacks where SDN and Network Functions Virtualization (NFV) technologies are implemented. CODE framework uses attack-detection algorithms hosted by mobile network operators (MNO) on special orchestration servers. The framework assumes that MEC nodes in the edge network can be trusted. They utilise IPS services provided by MEC backhaul cloud owned by external stakeholders. The framework uses a local orchestration mechanism in which inadequate defence MEC nodes collaborate with strongly related nodes, preventing cooperative defence from reaching its full potential. The CODE Framework [22] relies on MEC nodes to collaborate, which may not always be feasible. One possible resolution could be to use a centralised coordination mechanism to distribute defence resources among MEC nodes and avoid the dependence on node collaboration.

The authors in [23] proposed a machine learning-based technique for SDN using the NSL-KDD dataset to mitigate DDoS. The scheme improved the performance and accuracy of the SDN as depicted by the CPU utilisation and response time results. The number and quality of datasets used to train the model are directly related to accuracy or precision. The machine learning-based approach proposed in [23] relies on the quality and quantity of datasets used to train the model, which may affect the accuracy and precision of the model. One possible resolution could be to use a more diverse and larger dataset to train the model.

The authors in [24] proposed FAST, a feature adaptation reinforcement learning strategy based on space-time flow regularities in the Internet of Vehicles (IoV), for DDoS mitigation that solves a key challenge in the IoV under MEC. To deal with the dynamic variations in IoV and DDoS attacks, they used Q-learning and DDQN. Using feature adaptation in Q-learning can only ensure that the FAST approach produces a suitable DDoS mitigation result with unlabelled data. The FAST approach proposed by the authors in [24] may not be effective in dealing with new and unknown types of DDoS attacks. One possible resolution could be to incorporate a more advanced deep learning-based algorithm that can adapt to new and unknown types of attacks.

Flowranger [25], which uses trust management, queuing management, and scheduling requests to address the attack on the controller. Flowranger attempts to counteract these attacks by prioritising incoming requests to the controller, depending on their significance and influence on network performance. Its key feature is the ability to dynamically modify request priorities based on network conditions and attack severity. The system prioritises vital requests, such as those linked to network control and management, while lower-priority requests are dropped during attack periods.

Nevertheless, this approach is not quite effective as it fails to address the root causes of the attack. Prioritising and controlling incoming requests in this way may add overhead costs and drain computing resources from the controller, thereby reducing performance, especially during intense attacks. Its classification of requests into distinct priority tiers may lead to mistakes, with critical requests being wrongly deleted or non-critical ones receiving undue precedence, thus reducing network efficiency. To relieve concerns over resource consumption and overheads, improvements can be applied to speed up the request prioritisation process and reduce computational effort. This could include caching frequently accessed data, using efficient data structures, or assigning specific duties to multiple controllers. Decentralizing the control plane to distribute the workload across multiple controllers improves fault tolerance. Malicious traffic can be detected and controlled with the aid of distributed request prioritization and improved trust management employing dynamic traffic analysis and machine learning. Large-scale attacks can be more effectively addressed by rate-limiting, traffic shaping, and cooperative filtering with external services. Improved queuing techniques reduce overhead, and failover and redundancy technologies enable networks to continue running even in the presence of threats. These upgrades strengthen Flowranger's capacity to counter successfully DDoS attacks.

Another proposed scheme [26] uses monitor periods and threshold counters to detect the attack, but this is limited to specific times only. S-Guard [27] offers a lightweight mechanism to avoid address spoofing but affects network performance due to bandwidth consumption. FloodShield [28] is another proposal that helps to treat the DoS attack, but it faces obstacles in verifying the source address. To increase its applicability to DDoS attacks, the source address verification can be reinforced using methods like SYN cookies and ingress filtering, which counters IP spoofing observed in DDoS attacks. Furthermore, AI-based traffic analysis and rate limitation can be used to detect unusual trends in dispersed traffic. A distributed defensive system is made possible by collaborative filtering amongst network nodes; load balancing and Content Delivery Networks (CDNs) can be integrated with this system to spread traffic and avoid overload.

SDN Gradient Descent [29] uses the gradient descent algorithm to predict the DoS attack. Its effectiveness depends on the learning rate, which determines the algorithm's behaviour. It can be made more resistant to DDoS attacks by altering its learning rate and adding features that take into consideration distributed traffic abnormalities occurring across several nodes. The gradient descent algorithm's usefulness in this situation depends on its capacity to adapt dynamically to the higher volume and more varied network traffic that are prevalent in DDoS situations.

Additionally, a hybrid SDN-based approach [30] which combines neural networks and the K-nearest neighbour (KNN) algorithm to reduce the false-positive rate, can be extended to DDoS by training the neural network on a larger dataset that includes scattered attack patterns. By improving the detection of coordinated attack attempts, the hybrid strategy can offer a stronger defense against the volume and dispersion of traffic observed in DDoS attacks.

DoS defender [31] detects attacks by monitoring the number of flows in the OpenFlow switch. The proposed scheme in [32] helps reduce DoS attacks but its effectiveness may may be degraded under DDoS conditions due to the challenge of managing complex storage systems and higher bandwidth consumption that results from multiple, geographically distributed sources of attack. Similarly, the effective platform proposed in [33] is significantly limited. Despite being efficient for DoS, it could not work as well in a DDoS setting because of its limited capacity to scale and adjust to emerging attack techniques. These findings imply that in order to successfully counteract the DDoS attacks, such solutions could significantly be improved. Examples of these improvements could be the inclusion of more sophisticated traffic analysis, machine learning algorithms for anomaly identification, or cloud-based defense mechanisms.

The lightweight scheme proposed in [34] describes a simple method for detecting DDoS flooding attacks using the NOX/OpenFlow architecture. DDoS attacks, specifically flooding attacks, represent major dangers to network infrastructure, hence rapid and efficient detection is critical for network security. The suggested technique uses the NOX/OpenFlow framework's programmability and centralised control features to monitor network traffic and detect unusual patterns that indicate DDoS flooding attacks. The method's key benefits are its simplicity and efficiency, which allow it to be deployed in networks with rapid speeds without inducing significant costs. The approach detects DDoS flooding attacks in real time through analysis of flow data and packet headers, allowing for timely mitigation measures to safeguard network resources and services.

However, while the approach is intended to be lightweight, it can nevertheless add significant overheads, particularly in large-scale networks with high amounts of traffic. Furthermore, the scalability of the method may be an issue when applied in networks with quickly expanding traffic or different traffic patterns. To address these issues, additional optimisation of the technique for detection and use of resources may be required. This could include adjusting variables and thresholds to eliminate false positives and improve the technique's adaptability.

The machine learning-based approach proposed in [35] uses flow-based detection and mitigation to reduce low-rate DDoS attacks. Similarly, the authors in [35] introduced a model trained with a radial kernel function that takes advantage of new and advanced features to reduce DDoS attacks.

In [36], a spatial attention and convolutional neural network (CNNs) based on deep learning is proposed for image-based classification of malware. This is a novel way to detect malicious patterns in the context of DDoS detection and mitigation in MEC. This method, which makes use of deep learning techniques, has a high degree of accuracy when it comes to detecting malware which relates to DDoS attacks. This is the case since harmful software is frequently disseminated across compromised nodes in DDoS attacks. By improving the model's emphasis on important aspects of malware images, the spatial attention mechanism may increase the speed and accuracy of malware detection. However, there may be some challenges when using image-based categorization in MEC settings. Deep learning models, such as CNNs, can be resource-intensive, and MEC nodes normally have limited processing capabilities, which could affect real-time DDoS detection and mitigation.

On the other hand, SDN, as stated in [37] can provide a more direct method of DDoS mitigation in IoT and MEC environments by dynamically managing network flows and isolating malicious traffic. However, SDN systems may become targets of DDoS attacks due to their centralized control plane, which could lead to a single point of failure. While both approaches contribute to enhancing security in MEC environments, they may need to be complemented by additional strategies, such as distributed detection mechanisms and lightweight models, to effectively counter DDoS attacks

The authors [38] described a unique methodology for mitigating DDoS threats in MEC environments. The suggested framework is both scalable and adaptive, with the intended goal of effectively countering DDoS attacks while keeping MEC systems running efficiently. The system includes dynamic attack detection and mitigation techniques that can adapt to the changing nature of DDoS attacks. It uses MEC's distributed design to efficiently identify and mitigate threats closer to the network edge, minimising the impact on core network resources. Furthermore, the architecture was created to be adaptable, capable of withstanding different levels of attack strength and internet traffic volumes. The study includes data from experiments that show how effective the proposed architecture is at mitigating a variety of DDoS attacks in various environments. It offers potential enhancements in performance when compared to standard centralised mitigation systems, demonstrating the possibility of employing MEC to enhance network security.

However, one of the recognised limitations is a lack of scalability and adaptability to changing attack patterns. Many established structures are unable to adapt quickly to shifting attack techniques and intensities, thereby reducing their effectiveness. To overcome this limitation, the paper recommends using dynamic scaling techniques and adaptive algorithms that may reallocate resources in real-time, based on changing network conditions. MEC settings can better tolerate shifting DDoS attack patterns by dynamically allocating resources, hence improving overall network safety and durability.

The authors in [39] explored the use of machine learning approaches to detect DDoS attacks in the context of MEC. The authors offer an approach that uses machine learning techniques to detect aberrant network behaviour that is suggestive of DDoS attacks against MEC environments. The paper addresses emerging security challenges in MEC systems caused by the dynamic nature of the network environment and the resource limits that come with mobile devices and edge servers. The authors hope to build a proactive way to detect and mitigate DDoS attacks using machine learning, ensuring the availability and reliability of MEC services. The proposed anomaly detection framework is likely to include extracting significant characteristics from network activity data, training machine learning models on labelled datasets to recognise normal and attack traffic patterns and then deploying these models for real-time anomaly detection in MEC environments.

However, because of user and device mobility, MEC settings experience continuous variations in network circumstances, making anomaly identification difficult because normal behaviour might vary dramatically. Furthermore, the computational and energy availability of mobile devices and edge servers in MEC contexts are frequently constrained, hindering the implementation of effective anomaly detection techniques. Implementing adaptive learning techniques is critical for overcoming these challenges since it allows models to dynamically change to novel network conditions and dynamic attack strategies, assuring accurate detection. Furthermore, it is critical to properly evaluate the performance of anomaly detection systems using a variety of datasets and real-world scenarios to validate their effectiveness while reducing false positives.

The authors in [40] presented a cooperative defence mechanism to deal with the growing threat of DDoS assaults in MEC environments. It underscores the special issues provided by DDoS assaults on MEC systems and the need for proactive defence measures. It also requires collaboration across multiple MEC ecosystem elements, including edge servers, mobile devices, network infrastructure components, and cloud resources. This technique intends to strengthen MEC systems' resilience to DDoS attacks by combining their intelligence and resources. The technique defines the design and architecture of the cooperative defence mechanism, as well as how various entities coordinate and communicate information in real-time to identify and neutralise DDoS attacks. It may also address practical deployment issues like communication protocols, information-sharing techniques, and coordination strategies. Furthermore, the study is most likely to use simulations or experiments to examine the effectiveness of the suggested mechanism in detecting and mitigating various forms of DDoS attacks while avoiding false positives and false negatives.

Nonetheless, the study lacks implementation details for the cooperative defensive mechanism, such as communication protocols, information exchange mechanisms, and coordination strategies. While the proposed mechanism intends to improve MEC system resilience to DDoS attacks, scalability issues may occur when deploying it in large-scale MEC environments with many participating entities. To solve this conundrum, it is critical to give complete recommendations for deploying the cooperative defensive mechanism. This includes defining communication protocols, information sharing systems, and coordinating procedures to ensure uniform and effective deployment across several MEC environments. Furthermore, investigating optimisation strategies such as distributed processing, load balancing, and resource allocation algorithms might improve the scalability of the cooperative defensive mechanism, ensuring its efficacy in large-scale MEC deployment.

The authors in [41] discussed the necessity for adequate safety precautions in MEC systems, considering an increasing number of DDoS attacks. The study describes the design and architecture of an SDN-based DDoS attack detection and mitigation system, as well as how SDN principles are used to improve security in MEC situations. It may address the use of SDN's programmability and centralised control to enable real-time detection and mitigation of DDoS attacks on MEC infrastructure. Furthermore, the article discusses the system's mechanisms for identifying and effectively responding to DDoS attacks. This involves techniques like traffic analysis, anomaly detection, and dynamic reconfiguration of network resources to reduce attack flow. In addition, it also includes simulations to assess the performance of the SDN-based DDoS attack detection and mitigation system. This evaluation looks at the system's capacity to detect and mitigate various types of DDoS assaults while limiting disruption to legitimate network traffic.

However, the system must be scalable to meet the growing volume of network traffic, as well as the number of devices and users in MEC environments. Achieving effective detection and mitigation across large-scale deployments can be difficult. DDoS attack patterns are always developing as attackers devise new techniques to avoid detection. The system must be adaptive to successfully recognise and neutralise fresh attack patterns. To overcome this, dynamic traffic analysis algorithms could adjust to changing network conditions in MEC contexts. This may require machine learning algorithms that can learn and adapt to changing traffic patterns. Optimisation for resource-constrained devices would call for the creation of lightweight DDoS detection and mitigation algorithms tailored to resource-constrained mobile edge devices and servers. This could include strategies like algorithmic optimisations and effective resource utilisation.

Table 2.1: Literature Summary: DDoS Mitigation Techniques in Edge Computing Environments.

Table 2.1: Summary of our Literature review

Key Concepts and References	General Description	Advantages	Disadvantages
Edge computing [8]	Brings cloud computing capabilities closer to the network's edge, reducing latency and enabling smooth interaction with various application service providers and suppliers.	Decreases latency, enhances interaction with application service providers and suppliers.	Introduces security vulnerabilities.
[10]	Blockchain-based DDoS mitigation.	Public data sharing for authenticity.	Integration complexity with MEC.
DDoS attacks [11]	Severe threats hindering legitimate network requests.	Discusses mitigation approaches using Blockchain.	Integrating Blockchain into MEC system is difficult due to dynamic nature of channel circumstances and network loads.
SDN-based DDoS mitigation [12]	Uses SDN to detect and mitigate DDoS attacks effectively.	Provides comprehensive protection against DDoS attacks on IoT devices and infrastructure.	May endanger scalability of SDN-based IoT networks, complex administration and configuration.
MECshield DDoS prevention [6]	Utilizes MEC power to establish smart filters along the edge of attack-source/destination networks.	Offers localization of smart filters.	Suboptimal filtering performance due to difficulty in determining optimal placement.
Anomaly Tree-based approach [14]	Identifies and mitigates DDoS attacks in intelligent infrastructures of cities.	Dynamic threshold setting to avoid false positives and negatives.	May result in false positives and negatives.
Cooperative defense schemes [16]	Distributes defense resources among MEC nodes using global orchestration mechanism.	Efficiently utilizes defense resources.	Assumes trustworthiness of MEC nodes.
CODE framework [17]	Implements cooperative defense in MEC environment using SDN and NFV.	Relies on MEC nodes collaboration.	Collaboration among MEC nodes may not always be feasible.
Machine learning-based SDN [18]	Uses machine learning to mitigate DDoS attacks.	Improves performance and accuracy.	Relies on quality and quantity of datasets, may affect accuracy.

Key Concepts and References	General Description	Advantages	Disadvantages
FAST approach [19]	Uses feature adaptation reinforcement learning for DDoS mitigation in IoV under MEC.	Addresses dynamic variations in IoV and DDoS attacks.	May not be effective against new and unknown DDoS attack types.
Flowranger [20]	Mitigates DoS attacks using trust management, queuing, and scheduling.	Dynamically modifies request priorities based on network conditions.	Doesn't address root causes of the attack, adds overhead to the controller.
[22]	S-Guard lightweight mechanism for DoS in SDN.	Avoids address spoofing.	Bandwidth consumption, affects performance.
[24]	SDN Gradient Descent for DoS prediction.	Uses gradient descent algorithm.	Learning rate dependency.
Lightweight DDoS detection [29]	Uses NOX/OpenFlow architecture for efficient detection of DDoS flooding attacks.	Simple and efficient detection.	Adds overhead, scalability issues in large-scale networks.
[30]	Machine learning-based approach for low-rate DDoS.	Flow-based detection, radial kernel function.	Dependency on feature quality/quantity.
[31]	Spatial attention CNN for malware classification.	Deep learning-based classification.	Specific to image-based classification.
Cooperative defense mechanism [35]	Addresses DDoS threats in MEC environments through collaboration.	Strengthens MEC systems' resilience to DDoS attacks.	Lacks implementation details, scalability issues in large-scale deployments.
SDN-based DDoS mitigation system [36]	Uses SDN principles to detect and mitigate DDoS attacks in MEC systems.	Enables real-time detection and mitigation.	Needs to be scalable, adaptive to changing attack patterns.
[33]	Scalable and adaptive DDoS mitigation in MEC.	Dynamic attack detection, adaptive architecture.	Lack of scalability and adaptability.
[34]	ML-based anomaly detection for DDoS in MEC.	Proactive detection, anomaly identification.	User/device mobility challenges, resource constraints.
[35]	Cooperative defense mechanism for DDoS in MEC.	Resilience improvement, resource collaboration.	Lack of implementation details, scalability issues.

2.4 Conclusion

Edge computing is a new technique that brings cloud computing capabilities towards the network's edge, reducing latency and enabling interaction with various application service providers and suppliers. However, this also creates security vulnerabilities, and DDoS attacks are significant threats.

The aforementioned papers present various techniques and frameworks to mitigate DDoS attacks in MEC using SDN, NFV, and machine learning-based approaches. These techniques aim to detect and localise the attacks as close to their sources as possible, distribute defence resources among MEC nodes, and adapt to the dynamic variations in IoV and DDoS attacks. However, several approaches have been proposed to mitigate DDoS attacks in edge computing.

One approach involves using blockchain, which allows for public sharing of data on all nodes, but integration into the MEC system is challenging due to dynamic circumstances. SDN has also been proposed as a solution, utilising flow statistics from base stations and IoT applications to detect and mitigate attacks. Localisation is another solution, utilising MEC power to establish smart filters along the edge, while a global orchestration mechanism uses mini-bucket elimination algorithms for cooperative defence. CODE Framework uses IPS services provided by MEC backhaul cloud and local orchestration to prevent insufficient defence. Machine learning-based techniques using datasets and reinforcement learning have also been proposed.

DoS attacks are a significant problem in SDN, and several solutions have been proposed, including Flowranger, S-Guard, FloodShield, SDN Gradient Descent, a hybrid SDN-based approach, DoS defender, and Self Organising Maps. However, most of these solutions have limitations, such as being constrained to specific times, affecting network performance, or not allowing communication between different systems.

CHAPTER 3: PROPOSED APPROACH AND SIMULATION PARAMETERS

3 Introduction

DDoS attacks have become a major threat to the availability and performance of MEC systems. To mitigate the impact of these DDoS attacks, there is a need for a secure access network technique that effectively detects and filters out malicious traffic. In response to this challenge, this research proposes an anomaly-based detection technique that mitigates the impact of DDoS attacks on MEC systems. Anomaly-based detection techniques are increasingly recognised as effective mechanisms for identifying and responding to DDoS attacks, particularly in dynamic and distributed computing environments like MEC [42].

This chapter introduces the AdaptiveMECShield, a novel anomaly-based detection technique designed to mitigate DDoS attacks on MEC systems. The approach builds on the foundation of anomaly-based detection, leveraging techniques tailored to the unique characteristics of MEC systems. By analysing network traffic patterns and deviations from normal behaviour, this approach aims to detect and mitigate threats in real time, thereby enhancing the resilience of the MEC systems against DDoS attacks.

The chapter describes the technological underpinning the proposed technique, including its main characteristics, operational framework, and real-time threat adaptation. The chapter also includes the simulation parameters and settings that were be used to assess how well the AdaptiveMECShield approach performs. The technique improves MEC system security and dependability in the event of a DDoS attack.

3.1 The proposed approach

Our proposed technique, AdaptiveMECShield, presents a strategy that aims to reduce the adverse effects of DDoS attacks on the network while also improving the ordinary user experience. At its core, AdaptiveMECShield integrates anomaly-based detection techniques, which enable the identification of malicious traffic patterns and the prompt initiation of mitigation measures.

To address the challenges posed by DDoS attacks, we employ a combination of load balancing mechanisms, anomaly detection, and packet header analysis. We devised a load balancer at the network edge to distribute incoming traffic across multiple servers and configured it to perform SYN flood detection and mitigation, thereby reducing the likelihood of network congestion and packet loss during attack scenarios.

AdaptiveMECShield intends to mitigate DDoS attacks by allowing controllers to make appropriate packet traffic decisions. Attackers frequently send many packets, necessitating an increase in the number of flow rules within switching tables, leading to the emergence of numerous new flows. As a result, the controller is overburdened, resulting in an excess of dropped packets due to buffer overflow. The technique is based on the fundamental MECShield [43] framework, which includes a central controller and multiple nodes at the network's edge, wherein the communication between these entities is secured through robust protocols.

3.2 Windowing Technique

To improve the effectiveness of our anomaly detection systems, we applied a windowing methodology derived from the sliding window approach. This technique is crucial in maintaining an adaptable threshold for identifying network anomalies.

3.2.1 Sliding Window Concept

In network security, the sliding window concept refers to a dynamic monitoring strategy that constantly monitors traffic patterns over a given period while adjusting its analysis window in real time. This method enables the prompt detection of anomalies indicative of possible DDoS attacks. According to [44], the sliding window technique improves network security by dynamically modifying detection thresholds based on current traffic behaviour, resulting in fewer false positives and negatives. It also helps generate trust scores from traffic analysis, influences packet priority, and improves overall network resilience [44].

In contrast to its use in TCP for flow control, which provides reliable data transmission by regulating the quantity of unacknowledged data, the sliding window method is used in security applications such as AdaptiveMECShield to detect and mitigate anomalies [44]. Deviations from regular traffic patterns can be easily detected by monitoring indicators such as packet rates and IP addresses within the sliding window. When an abnormality is detected, the system can take proactive measures like rate limiting or blocking questionable sources to prevent DDoS attacks [44]. This adaptive technique dramatically improves the security of MEC settings by responding to developing threats and providing continuous real-time monitoring.

3.2.2 Adaptive Threshold

To improve on the sliding window approach, AdaptiveMECShield incorporates an adaptive threshold mechanism. Instead of using a static threshold, which may not be useful under changing network conditions, the adaptive threshold adapts dynamically based on traffic patterns detected within the window. This strikes a balance between detecting malicious traffic and reducing false positives.

The adaptive threshold is calculated by looking at the statistical features of the traffic within the window, such as the mean and standard deviation of packet speeds. By adjusting the threshold based on these attributes, AdaptiveMECShield can more accurately differentiate between normal traffic fluctuations and actual anomalies indicative of a DDoS attack.

Moreover, our solution incorporates windowing techniques and trust management algorithms to dynamically adjust trust values based on packet header information. By monitoring the header fields such as IP addresses, destination IP addresses, and source ports, we detect anomalies indicative of DDoS attacks and prioritise packet delivery accordingly. When a node transmits more packets than a predefined threshold, the controller evaluates its Packet Delivery Ratio (PDR) concerning surrounding nodes. Nodes that flood the network have extremely high PDR values. When anomalous PDR patterns are detected, the controller flags the implicated node as malicious, indicating the source of attack. This classification is then disseminated through policy dispatches, effectively causing the network to server communication with the flagged node that exceeds the packet threshold [45].

Our primary goal is to develop a solution that successfully minimises the disruptive impact of DDoS attacks on MEC networks. Through testing and fine-tuning the detection thresholds, we aim to optimise the performance of our technique while minimising false alarms. This empowers authorised users to carry out their tasks unhindered by poor network performance or bandwidth limits. A dynamic equation has been devised to adjust the packet threshold in real time, as outlined below:

$$\mathrm{threshold} = \mathrm{f}(\mathrm{time}, \mathrm{traffic}, \mathrm{bandwidth})$$

The function f in the equation represents a model that calculates the threshold based on the variables time, traffic, and bandwidth.

$$\mathrm{th}(s) = \mathrm{th}(s) + \mathrm{tv}(s) \quad (1)$$

Where the th(s) represents the threshold at state s and tv(s) represents the trust value at state.

$$\mathrm{th}(s) = \mathrm{th}(s) - \mathrm{tv}(s) \quad (2)$$

Where the threshold th(s) is adjusted by subtracting the trust value tv(s).

Equation (1) operates when the network is in its normal functioning state, with no indications of an ongoing attack. It incorporates the threshold value (th(s)) and the user's trust value (tv(s)). This prioritises packet forwarding from users with higher trust levels over those with lower trust values.

When indicators of an attack surface are present, Equation (2) comes into play. It entails subtracting the trust value (tv(s)) of the user from the threshold value (th(s)). As a result, packets from users with greater trust values are given lower priority for forwarding than packets from users with lower trust values.

This cognitive equation recalculates a new threshold value by taking three essential variables into account. The resulting threshold value is subsequently used to evaluate whether a packet should be dropped or not. Dropped packets are intercepted before they reach their destination server. The dynamic modification of the threshold value via this equation is critical in protecting authorised users from the disruptive impacts of DDoS attacks. Figures 3.1 and 3.2 illustrate the steps applied to compute the threshold and the proposed DDoS attack detection approach.

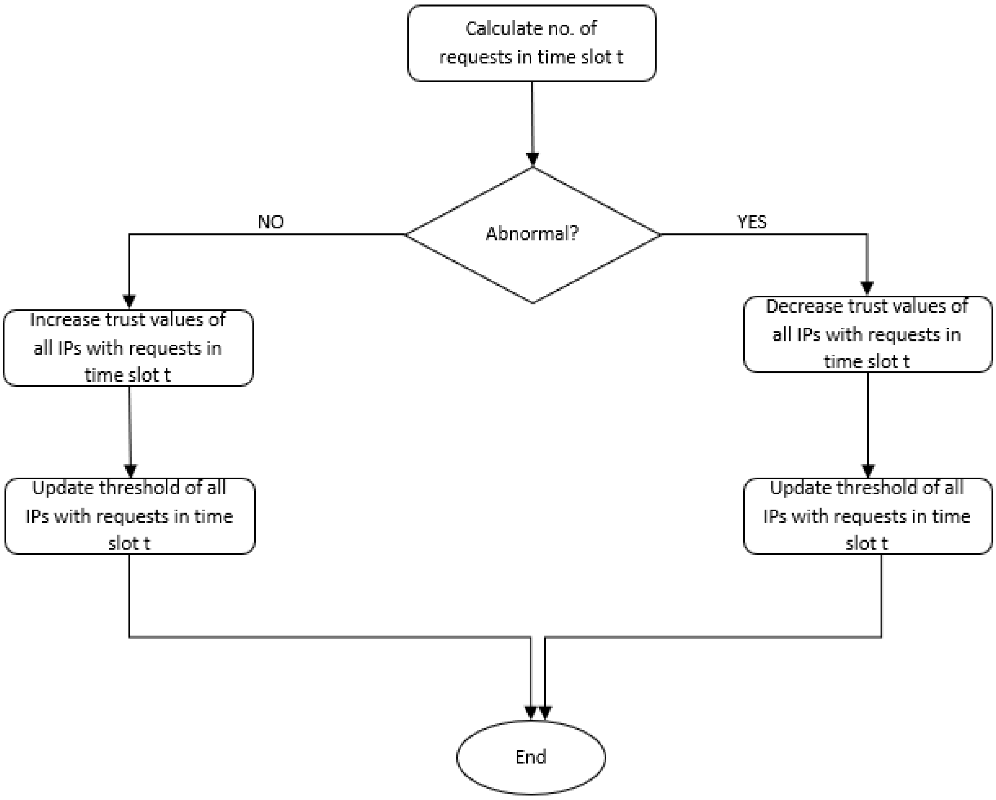

Figure 3.1: Steps of Computing the Threshold

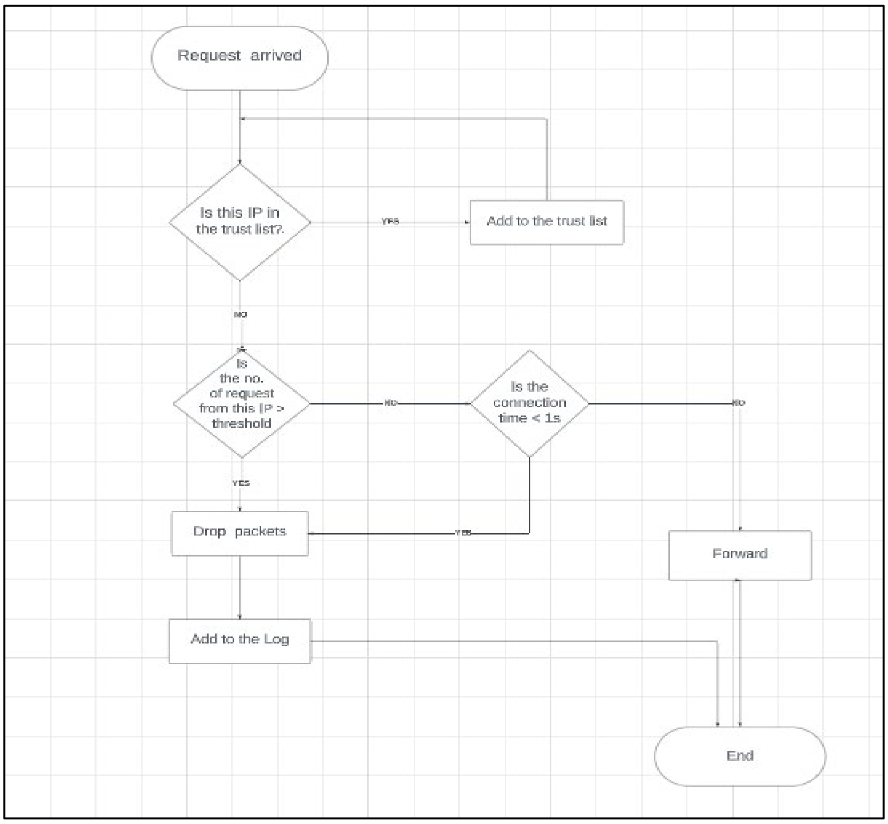

Figure 3.2: The DDoS attack detection Technique.

3.2.3 Header Field Analysis

We begin by identifying the header data connected with the flow under attack in the controller. These header elements are checked to aid in the detection of DDoS attacks. The results are checked against predetermined thresholds within time intervals. If the measured counts for the header fields exceeds the threshold, this indicates a DDoS attack. In contrast, during normal network traffic with no DDoS attack, the values for the header fields remain below the threshold levels.

3.2.4 Calculating the trust value based on header field information using windowing.

The major goal of this study is to protect legitimate users from the negative effects of DDoS attacks on edge controllers. To achieve this goal, we introduce an algorithm that uses windowing to monitor the header field attributes of packets originating from certain users. Following that, the acquired data is used to compute a trust score for each user, which informs the prioritisation of packet delivery. Using this technique, header field values are monitored and compared to predetermined criteria. Utilizing data from packet header fields, we can compute user trust levels. Algorithms 1 (**Appendix A**) and 2 (**Appendix B**) provide details on how to calculate trust management.

The windowing strategy allows us to dynamically react to changes in network traffic patterns. For example, if a user initiates a large flood of packets with unusual header field properties, their trust score is bound to suffer. Here is how it works:

3.2.4.1 Monitoring Packet Headers:

- Within a defined time window, the system collects packet header data such as IP addresses, source ports, destination ports, and other relevant fields.
- This data is stored and analysed in real-time to identify patterns and deviations from normal traffic behaviour.

3.2.4.2 Identifying Anomalous Behaviour:

- If a user starts sending a large flood of packets with unusual or suspicious header field properties (e.g., high volume of SYN packets or packets with spoofed IP addresses), these packets will stand out against the normal traffic patterns.

- The system compares the current traffic against historical data stored in previous windows to detect significant deviations.

3.2.4.3 Updating Trust Scores:

- Each user has an associated trust score that reflects their behaviour over time. When anomalous activity is detected, the trust score of the user responsible for the suspicious traffic is adjusted accordingly.
- For example, if a user is observed sending an unusually high number of packets with headers indicative of a SYN flood attack, their trust score is accordingly decreased.

3.2.4.4 Algorithm for Trust Score Adjustment:

Based on the behaviour, the algorithm updates the trust value and threshold for known sources. When a known source sends a packet, it indicates that it is a legitimate source. The parameter α denotes the rate of adjustment. When the source sends a packet under normal conditions, it indicates legitimacy. As a result, the trust value and threshold for that source increase, taking into account the header field values (as illustrated in equations (3), (4), and (1)).

$$\mathrm{tv(s)} = \propto * \mathrm{tv(s)} + 1 \qquad (3)$$

$$\mathrm{tv(s)} = \propto * \mathrm{tv(s)} + 0.1 * \mathrm{header\_factor} \qquad (4)$$

The next action to take if this occurs is to determine whether the source's trust value is lower than the required minimum for a valid source before blocking it. When a DDoS attack is launched against the controller, the algorithm uses the header field values provided in Equations (5), (4), and (2) to lower the trust and threshold values.

$$\mathrm{tv(s)} = \propto * \mathrm{tv(s)} - 1 \qquad (5)$$

According to p equation (6), each source that does not deliver a packet within the time period increases its trust value.

$$\mathrm{tv(s)} = \propto * \mathrm{tv(s)} \qquad (6)$$

3.2.4.5 Prioritising Packet Delivery:

- The system prioritises packets based on the trust score of the source. Users with higher trust scores have their packets forwarded with higher priority, while users with lower trust scores (indicating suspicious behaviour) face reduced priority or potential blocking.

- This dynamic adjustment ensures that legitimate users are less likely to be affected by the mitigation measures while malicious users are effectively curtailed.

3.2.4.6 Blocking Malicious Sources:

- If a user's trust score falls below a certain threshold, the system identifies them as a potential threat and takes action to block their traffic.
- This blocking is communicated through policy dispatches, effectively severing communication with the malicious node.

Example Scenario: Consider a user who starts a SYN flood attack. The windowing strategy allows the system to quickly detect the high volume of SYN packets from the same source within a short time window. The algorithm then decreases the user's trust score based on the abnormal header field properties observed. As the trust score drops, the system dynamically adjusts the threshold for forwarding packets from this user, prioritising other traffic and eventually blocking the user's packets if the behaviour persists.

Users that routinely submit packets with conventional header field values, on the other hand, have their trust value increase.

To summarise, the algorithm keeps track of regular users and their trust levels. When the network is in a normal state, ordinary users' trust values rise. When the network is under attack, all sources' trust values are reduced. A source is prohibited if its trust value falls below a certain level.

3.3 Simulation environment

Tables 3.1 and 3.2 present the system specification to be used and the simulation parameters respectively.

Table 3.1: System specification

System model	DELL Latitude 5520
System type	x64-based PC
Processor	Intel® Core ™ i7-1165GT @ CPU 2.80GHz
Installed Physical Memory (RAM)	8 GB
Operating System (OS) Name	Ubuntu 14.0 LTS

Table 2.2: Simulation environment and tools

Simulation environment/parameter	**Tools**
Network emulator	Mininet
Packet Analyzer	Wireshark
Types of attack	DDoS
Network tool	Hping3
Coding Language	Python3

3.3.1 Network emulator

In our implementation, we employed the Mininet emulation tool to establish the network topology on the host user. Mininet [46] is a network emulation platform that uses the OpenFlow protocol. It makes use of lightweight virtualisation to simulate many end hosts, switches, routers, and links within a single Linux kernel. The emulator contains tools for determining potential bandwidth, node connectivity, deep node connectivity, and flow speed. It enables the creation of lifelike virtual networks using simple commands. Through its command line interface (CLI) mode, users can connect to the network. Mininet offers a wide range of network topologies suitable for experimentation and testing purposes [46]. To run Mininet, we utilised a Linux virtual machine, which acts as a software program emulating a physical Linux computer on the host system. This approach ensures a controlled environment for running Mininet and facilitates easy exploration of diverse network topologies. For our project, we opted for a customised topology consisting of 100 hosts and 60 switches for various reasons:

3.3.1.1 Scalability and Realism:

This topology can be used for modelling due to the use of large networks. The 100 hosts at this level of abstraction are quite different devices on complex networks whereas 60 switches are interconnection/routing devices and support that scale.

3.3.1.2 Performance Testing:

The performance testing of devices as well as the communication protocols is usually conducted by modelling and analyzing a network that has such a topology. It is useful to comprehend the behaviour of the network in regard to large amounts of data, many data transfers and routing of the network to understand how to improve the performance of the network.

3.3.1.3 Topological Flexibility:

Given the custom topology, we are able to recreate network layouts in particular: routing algorithms, positions of switches and flow control. This level of control makes it possible that the experiments are constructed as per project requirements.

3.3.1.4 Fine-Grained Network Monitoring:

Due to the superiority of Distributed environment, with many clients as well as switches, many parameters in relation to the network can be investigated such as throughput, flow rates, and the connectivity of nodes to the network. This helps in the behavioural patterns of the networks and their performance, including the possibilities of bottlenecks or possibilities of improvements.

3.3.1.5 Custom Experimentation:

In the case when the topology is modified, we are able to change the speed of the links or the activities of the hosts or add some undesirable features of the internal network. This allows one to collect relevant data and information that can, be manipulated.

3.3.2 Traffic Generation

Scapy [47], developed by Phillipe Biondi in Python, is a powerful computer network manipulation tool. It is able to construct and interpret packets, transmit them through a network, intercept them, and establish connections between requests and responses. In addition, it offers a range of functionalities including scanning, tracing, searching, and system monitoring. Its Python interface resembles Wireshark's graphical user interface for capturing and displaying network data. Moreover, Scapy surpasses other similar tools by supporting packet injection, custom packet formats, and scripting capabilities [47].

3.3.3 Conclusion

This study presents a secure access network technique to protect MEC devices from DDoS attacks. Implementing a load balancer, applying SYN flood detection and mitigation, and leveraging SDN controllers for real-time traffic monitoring and decision-making are all part of the technique. The objective is to reduce DDoS attacks while improving the user experience.

The simulation environment and parameters were specified, outlining the system model and tools used for analysis and evaluation. Traffic generation techniques, including the utilisation of Scapy for packet manipulation and analysis, were described.

The proposed technique incorporates load balancing setup, SYN flood detection, traffic filtering, rate limiting, real-time statistics collection, header field analysis, trust management algorithms and simulation techniques. These measures collectively contribute to the mitigation of DDoS attacks and the protection of MEC systems.

The algorithm for detecting SYN flood attacks and blocking the source IP when the threshold is exceeded was provided. The algorithm utilises packet sniffing, packet analysis, and IP blocking mechanisms to address SYN flood attacks effectively.

In summary, our proposed anomaly-based detection technique represents a proactive and adaptive approach to protecting MEC systems against DDoS attacks. By combining various strategies such as load balancing, traffic filtering, and header field analysis, AdaptiveMECShield enhances the security and resilience of MEC systems. This ensures uninterrupted availability and improved performance of critical edge computing services. Through a comprehensive simulation and evaluation process, we demonstrate that our approach has the potential to effectively mitigate DDoS attacks and protect MEC which improves the user experience and maintain the integrity of the network.

CHAPTER 4: EVALUATION METHODOLOGY

4. Introduction

In this section, we offer results of our techniques and performance evaluation. We begin by outlining the evaluation process used to determine the efficacy of our detection system. In our technique, we create a threshold in the device's table, and surpassing it indicates that the gadget is "under attack." When choosing the threshold, it is critical to strike a balance between capturing all attacks and minimising false alarms in the absence of actual attacks. We used empirical testing to evaluate our strategy, experimenting with different thresholds to establish the best balance of detection accuracy, false negatives, and false positives. It is worth mentioning that the assessment metrics and equations described here can be used to measure the performance of numerous domains, including load balancing for assessing the performance and effectiveness of detection systems.

4.1 Evaluation methodology

We demonstrate the evaluation methodology and show the performance of our proposed method.

AdaptiveMECShield flags a device as "under attack" when the number of entries in the table exceeds a certain threshold. It is critical to set the threshold low enough to capture all attacks, but not to trigger warnings when no attacks are attempted. Balancing this threshold is crucial to minimise both false positives and false negatives. False negatives refer to undetected attacks, whereas false positives refer to false alarms that occur in the absence of such attacks [48]. Empirical testing can identify the appropriate threshold by experimenting with different thresholds and comparing detection accuracy while limiting the occurrence of both false negatives and false positives [48]. Table 4.1 outlines the threshold rate on detection accuracy, false negative, and false positive.

Table 3.1: Threshold Values and Corresponding Accuracy in Attack Detection and Classification Errors

Threshold	Detection Accuracy	False Negatives	False Positives	Attack/Normal
10	95%	2%	3%	Normal
15	92%	5%	3%	Normal
20	88%	8%	4%	Borderline (could be either)
25	85%	12%	5%	Attack
30	82%	15%	8%	Attack

We simulated our technique with a network of 100 legitimate hosts and 5 attacking devices. We used a sliding window of 10 seconds to check for a DDoS attack. We set the forgetting factor α to 0.9, the number of queues in the priority buffers Nq to 10, the priority factor in request scheduling increase β to 1.5, and the trust increasing factor due to the value of a header field increase γ to 0.1. We set the minimum threshold for the user Tmin to 12, the maximum number of packets for any user to send Lmax to 20, and the maximum number of packets that may reach the controller in the normal state in a one-time slot Mmax to 50.

We generated normal traffic with an inter-arrival time of 0.1 seconds and an average arrival rate of 10 packets per second. We generated attack traffic with an inter-arrival time of 0.01 seconds and an average arrival rate of 100 packets per second.

The controller maintains a sliding window of trust values for each source. For each packet arriving at the controller, the algorithm updates the trust value for the source based on the packet headers and the sliding window.

If the trust value for a source fall below the minimum threshold, the source is blocked. The controller periodically checks the sliding window to identify any sources that are sending a high volume of traffic or traffic with suspicious header values. If the controller detects such a source, it is blocked. The sliding window allows the algorithm to adapt to changes in the network traffic. For example, if a source suddenly starts sending a high volume of traffic, the algorithm gradually decreases the trust value for that source. If the source continues to send a high volume of traffic, it is eventually blocked.

This algorithm is effective at mitigating DDoS attacks because it can identify and block malicious sources early on. By blocking malicious sources, the algorithm can reduce the load on the controller and protect legitimate users from being overwhelmed.

True Positive (TP) is the number of attack events that are successfully identified, whereas False Negative (FN) is the number of attack events that are incorrectly identified as normal. True Negative (TN) denotes the number of normal events successfully detected, while False Positive (FP) denotes the number of normal events incorrectly identified as attacks [49].

The detection accuracy statistic indicates the proportion of packets that are accurately classified and tagged concerning the total number of packets in the sample [49]. It quantifies the detection system's ability to sense the right packet types as depicted in Eq. (7).

$$\text{Detection accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{7}$$

The false-positive rate is a metric that calculates the percentage of legitimate packets that are mistakenly labelled as attacks in comparison to the total number of legitimate packets. It draws attention to the incidence of false alerts or the misclassification of normal traffic as malicious traffic [49]. The formula for calculating the false-positive rate is given in Eq. (8).

$$\text{False} - \text{positive rate} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{8}$$

The detection rate is a measure that calculates the percentage of accurately identified and labelled threat packets concerning the total number of threat packets [49]. It demonstrates the system's capacity to identify and respond to malicious conduct as defined in Eq. (3).

$$\text{Detection rate} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{9}$$

4.1.1 Performance parameters

The proposed detection system's efficacy is comprehensively evaluated by examining its capacity to properly and effectively identify and intercept malicious network traffic. The following performance parameters are used to achieve this comprehensive evaluation:

I. Detection accuracy: Indicates the proportion of packets that are accurately classified as attacks, demonstrating the effectiveness of the detection system in identifying malicious activities.

II. False negatives: Denotes an instance where an attack is not detected, leading to undetected malicious activity.

III. False positives: Denotes the number of normal events incorrectly identified as attacks, leading to unnecessary alerts.

IV. Detection time: The amount of time it takes to detect an attack and enable timely response to mitigate and minimise the impact of the attacks on the network performance.

V. Memory Utilisation: Amount of memory used over time

VI. CPU Utilisation: CPU resources consumed over time.

4.2 Conclusion

The evaluation of our detection system demonstrates its robustness and efficiency in detecting and mitigating DDoS attacks in a network environment. By utilising the AdaptiveMECShield technique, we efficiently adjust the detection threshold to enhance the capturing of attack attempts while minimising false alarms. The empirical testing and simulation of our technology, which includes a network of 100 legitimate hosts and 5 attacking devices, show that the system is feasible and performs well.

The thorough investigation of performance measures such as detection accuracy, false negatives, false positives, and detection time confirms our approach's strengths as well as possible spaces for improvement. The system's capacity to adaptively adjust trust values depending on network traffic patterns allows a dynamic reaction to changing attack scenarios, protecting genuine users while maintaining the network's speed.

CHAPTER 5: PERFORMANCE EVALUATION

5.1 Introduction

In this chapter, we present the findings and performance evaluation results of our proposed technique designed to detect malicious network traffic, the DDoS attacks. We show that our strategy of detecting and mitigating such risks is effective through simulation. Our technique effectively adjusts to changing network conditions and detects potential attacks using a windowing and trust-based procedure.

Using a network simulation with 100 legitimate hosts and 5 attacking nodes, we evaluated the scheme using many performance metrics to determine the effectiveness of our detection technique. The metrics include detection accuracy, false negatives, false positives, and detection time.

5.2 Performance evaluation

Figures 5.1, 5.2, 5.3 and 5.4 depict the detection accuracy, false negatives, false positives, and detection time of the four DDoS attack detection techniques. The figures provide a comparison of the performance of different DDoS attack detection techniques using multiple metrics, including accuracy, false positives/negatives, and detection time.

Figure 5.1: The comparative detection results

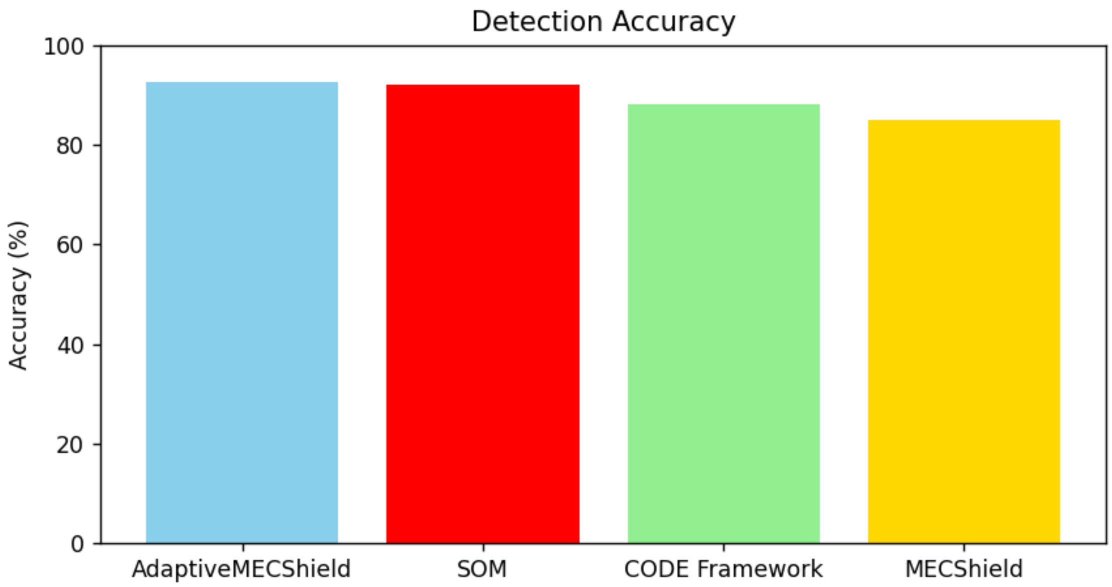

Figure 5.1: Detection Accuracy results

The results depicted in Figure 5.1 show the detection accuracy of four different DDoS attack detection techniques.

Table 4.1: Presents the detection accuracy amongst four attack detection techniques.

Table 5.1: Detection Accuracy Rate

Scheme	Detection Accuracy (%)
AdaptiveMECShield (proposed technique)	92.5
SOM	92
CODE Framework	88
MECShield framework	85

The AdaptiveMECShield has the highest detection accuracy because it uses the adaptive threshold to sense potential attacks, achieving a balance of sensitivity and specificity. By doing so, it ensures that the system is sensitive enough to detect most attacks (resulting in high detection accuracy) while remaining particular enough to reduce false positives, hence avoiding the unnecessary designation of benign traffic as threats. This balance is critical for establishing effective security measures while maintaining legitimate operations [50], [51], [52]. By adjusting the threshold through empirical testing, AdaptiveMECShield achieves a high level of accuracy in differentiating between normal and malicious network traffic. Its adaptive algorithms enable the approach to dynamically change its detecting processes in response to real-time network conditions such as variations in traffic volume and patterns, anomalies in packet characteristics, deviations in protocol usage, fluctuations in resource utilisation, changes in behavioural patterns. This is unlike the SOM, CODE, and MECShield frameworks, which have slightly poorer detection accuracy. While still displaying decent performance, these approaches have disadvantages such as inferior threshold settings and less adaptive algorithms than the AdaptiveMECShield. We present below an outline of the key components and factors that contribute to the detection accuracy of AdaptiveMECShield:

I. Timely Detection of Anomalies

AdaptiveMECShield excels in identifying anomalies quickly because of its real-time monitoring and good anomaly detection algorithms. This capacity is crucial for decreasing the risk window by quickly identifying anomalous network patterns. The system's use of statistical analysis and machine learning models enables it to detect both known attack signatures and emerging threats, facilitating early mitigation efforts. By recognising anomalies as they occur, AdaptiveMECShield switches from a reactive to a proactive defensive technique, dramatically improving network security.

II. Adaptive Thresholding

A key feature of AdaptiveMECShield is its dynamic thresholding method, which adapts in real time, depending on traffic circumstances. This adaptive technique keeps the system aware of actual abnormalities while reducing false positives and negatives. Using continuous traffic pattern analysis and a feedback loop, the system may learn and improve its threshold settings over time. As a result, AdaptiveMECShield can successfully adapt to abrupt traffic spikes typical of DDoS attacks while maintaining excellent detection accuracy across a wide range of network environments.

III. Advanced Trust Management Algorithm

A significant aspect of AdaptiveMECShield is its real-time trust management system, which continuously examines each network node's trustworthiness depending on its activity. This system prioritises packet delivery for valid users, resulting in minimal disruption to normal operations even during an assault. The trust management system constantly adjusts trust scores, penalising nodes that engage in suspicious behaviour while rewarding consistent, benign activity. This method enables the system to swiftly isolate and mitigate risks, creating a proactive security mechanism that adapts to changing attack techniques.

IV. Comprehensive Multi-layered Defence

AdaptiveMECShield integrates several defensive approaches, such as load balancing, packet header analysis, and traffic filtering, to provide a strong multi-layered defence against DDoS attacks. Load balancing distributes traffic evenly across servers to avoid overloads and ensure service uptime. Packet header analysis examines communication at the granular level, allowing for precise anomaly detection. Traffic filtering improves security by enabling only legitimate traffic to access crucial network resources. This multi-layered method provides comprehensive security by tackling numerous components of DDoS attacks, resulting in a proactive and reactive defence capable of reacting to new threats while reducing existing attacks.

V. Efficient Resource Management

Resource efficiency is a significant benefit of AdaptiveMECShield. The system uses less CPU and memory than classic techniques such as SOM. Efficient resource management ensures that network performance remains optimal even under peak traffic or attack scenarios. This efficiency also means that AdaptiveMECShield can competently scale to manage huge networks with many nodes without significantly degrading performance. By limiting resource consumption, the system ensures that network resources are available for genuine traffic, hence increasing overall network resilience.

VI. Limitations of Traditional Schemes

Traditional DDoS mitigation techniques, such as those based on static thresholds, may fail to adapt fast to abrupt traffic changes, resulting in delayed anomaly identification and a higher incidence of false alarms. Fixed thresholds cannot adapt to changing traffic circumstances, resulting in an excessive number of false alarms or missed detections. These techniques are also resource-intensive, requiring more CPU and memory, which might affect network performance. Furthermore, many traditional solutions lack a complete approach, focusing instead on specific tactics such as rate limiting or basic traffic monitoring. This narrow concentration makes them vulnerable to sophisticated attacks that can quickly get past single-layer defence. Furthermore, traditional techniques sometimes do not include constant trust evaluation, which leads to slower response times and less effective threat isolation.

Figure 5.2 presents the false negative results.

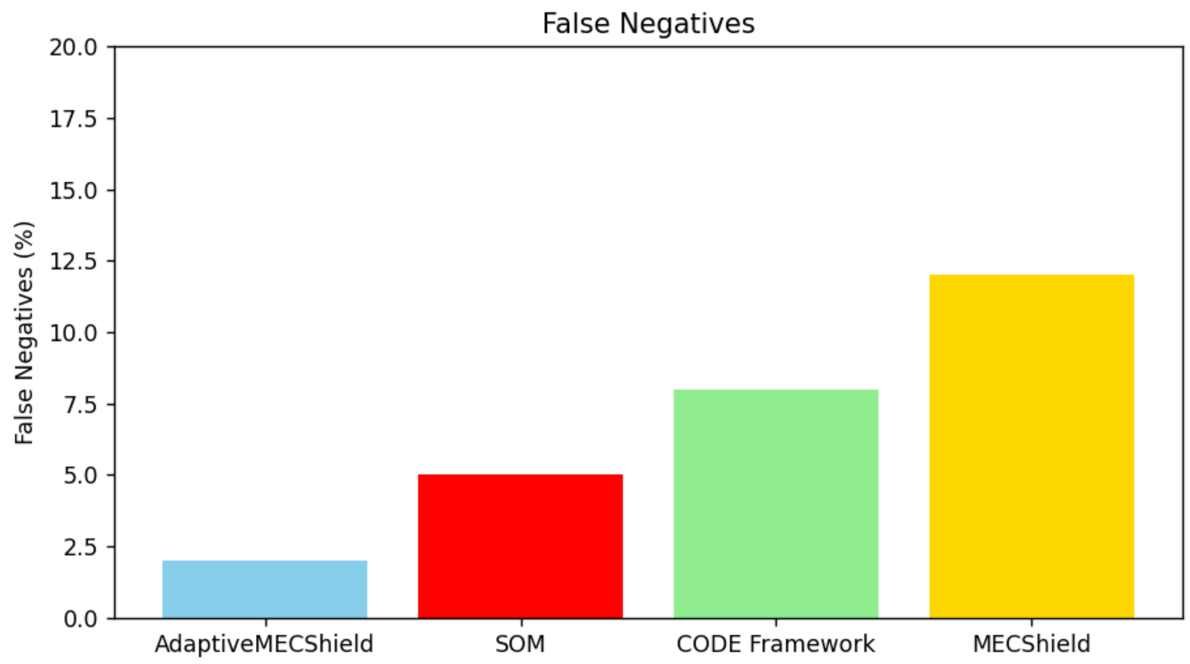

Figure 5.2: False Negative Results

Table 5.2 presents the false negatives of the four DDoS attack detection techniques.

Table 5.2: False Negative Rate

Scheme	False Negative (%)
AdaptiveMECShield (proposed technique)	2
SOM	5
CODE Framework	8
MECShield framework	12

The results presented in Figure 5.2 highlight the performance of four different DDoS attack detection techniques in terms of their false negative rates. False negatives represent instances where a system fails to detect an actual DDoS attack.

The proposed technique, AdaptiveMECShield, has the lowest false negative rate of all the evaluated techniques at 2%. This indicates that AdaptiveMECShield has a high capability to accurately detect DDoS attacks and minimize instances where attacks are undetected. Following AdaptiveMECShield, the SOM technique demonstrated a slightly higher false negative rate of 5%. While not as low as the proposed technique, it still represents a relatively effective performance in terms of detecting DDoS attacks.

The CODE framework showed a higher false negative rate of 8%, indicating a somewhat higher likelihood of failing to detect DDoS attacks compared to the previous two techniques. However, it still performs reasonably well in terms of false negatives. The MECShield framework has the highest false negative rate among the evaluated techniques, at 12%. This suggests a relatively lower effectiveness in detecting DDoS attacks compared to the other techniques evaluated.

The results indicate that the proposed AdaptiveMECShield technique outperforms the other techniques in terms of minimizing false negatives, followed by SOM, CODE framework, and finally the MECShield framework. These findings underscore the importance of selecting an appropriate detection technique to effectively mitigate the risks associated with DDoS attacks. AdaptiveMECShield outperforms other techniques due to several factors as detailed below:

- Dynamic Adaptability: It can update its detection parameters in real time, allowing it to respond to changing attack patterns and network traffic. This versatility allows consistent performance under different settings.
- Advanced Algorithms: The employment of advanced algorithms improves the detection of DDoS attacks with high accuracy. These techniques are intended to detect small irregularities in network behaviour while minimizing the possibility of false negatives.
- Comprehensive Analysis: AdaptiveMECShield examines a wide range of data points and metrics to provide a complete picture of network traffic. This robust methodology enables it to uncover trends and signs that less thorough systems may overlook.
- Low False Negative Rate: With a false negative rate of around 2%, AdaptiveMECShield is extremely effective at reducing undetected DDoS attacks. This low rate is critical for maintaining strong network security and detecting threats immediately.

Figure 5.3 depicts the false *positive* results between different detection systems.

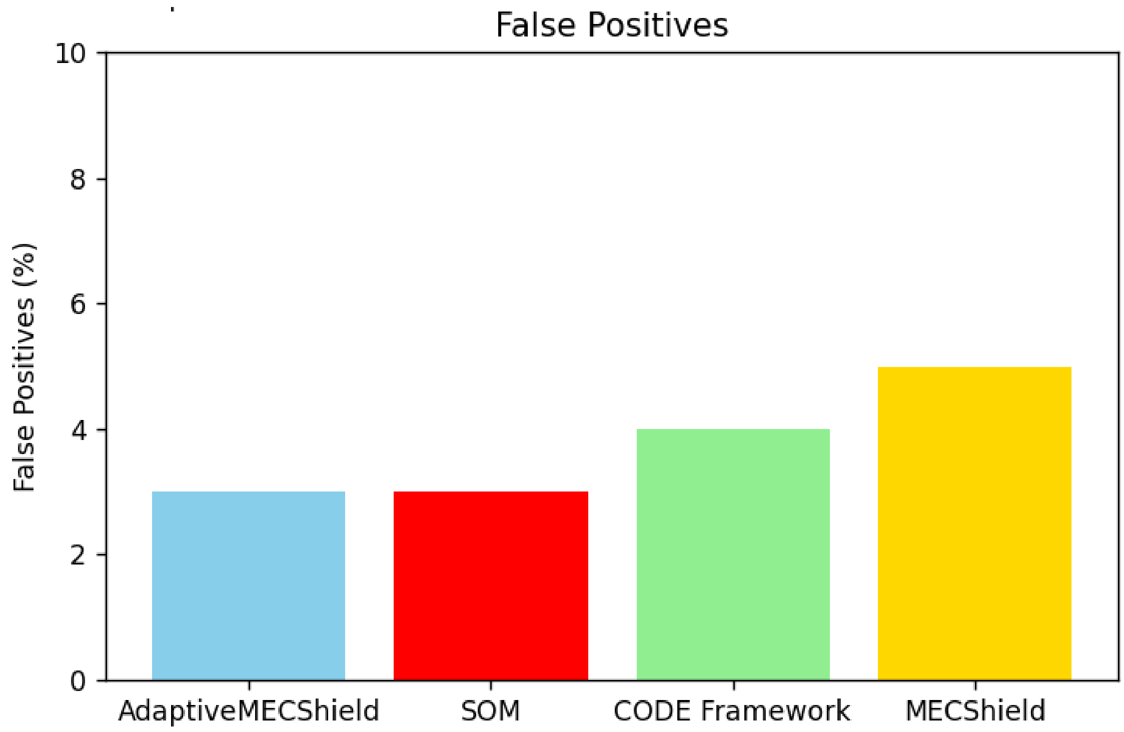

Figure 5.3: False Positives Results

Table 5.3 depicts the false positives of the four DDoS attack detection techniques.

False positives occur when non-malicious nodes are detected as an attack. Minimizing the false positives prevents unnecessary disruptions to network operations.

Table 5.3: False Positives Rate

Scheme	False Positives (%)
AdaptiveMECShield (proposed technique)	3
SOM	3
CODE Framework	4
MECShield framework	5

The results presented in Figure 5.3 offer insights into the performance of four DDoS attack detection techniques in terms of their false positive rates. False positives occur when non-malicious nodes are incorrectly flagged as malicious, leading to unnecessary disruptions in network operations.

Both the proposed technique, AdaptiveMECShield, and SOM demonstrated the lowest rate of false positives, each at 3%. This suggests that these techniques are highly effective in distinguishing between legitimate network traffic and potential DDoS attacks, thereby minimizing the occurrence of false alarms and unnecessary disruptions.

Following closely, the CODE framework exhibited a slightly higher false positive rate of 4%, indicating a marginally increased likelihood of false alarms compared to the previous two techniques. However, it still maintains a relatively low false positive rate, indicating a good balance between sensitivity and specificity in detecting DDoS attacks.

The MECShield framework showed the highest false positive rate among the evaluated techniques, at 5%. While still relatively low, this suggests a slightly higher propensity for incorrectly identifying non-malicious nodes as malicious, potentially leading to more frequent false alarms and network disruptions.

The results indicate that both the proposed AdaptiveMECShield technique and SOM perform exceptionally well in minimizing false positives, followed closely by the CODE framework. The MECShield framework, while still effective, demonstrates a slightly higher false positive rate compared to the other techniques evaluated. These findings underscore the importance of selecting a detection technique with a balanced approach that minimizes false alarms and maintains smooth network operations. AdaptiveMECShield outperforms other techniques in minimizing false positives due to several key factors as elaborated hereunder.

- Advanced Precision: It uses highly sophisticated algorithms to distinguish between legitimate traffic and potential DDoS attacks. This precision lowers the likelihood of false positives, ensuring that normal network operations are not disturbed.
- AdaptiveMECShield balances sensitivity with specificity, allowing it to detect true threats while reducing false alarms. This balance is critical for ensuring network integrity while avoiding unwanted interventions.
- Sophisticated Filtering Mechanisms: The system uses advanced filtering mechanisms to precisely evaluate traffic patterns. This facilitates identifying and excluding non-malicious nodes, lowering the likelihood of false positives.

Figure 5.4 presents the detection time results.

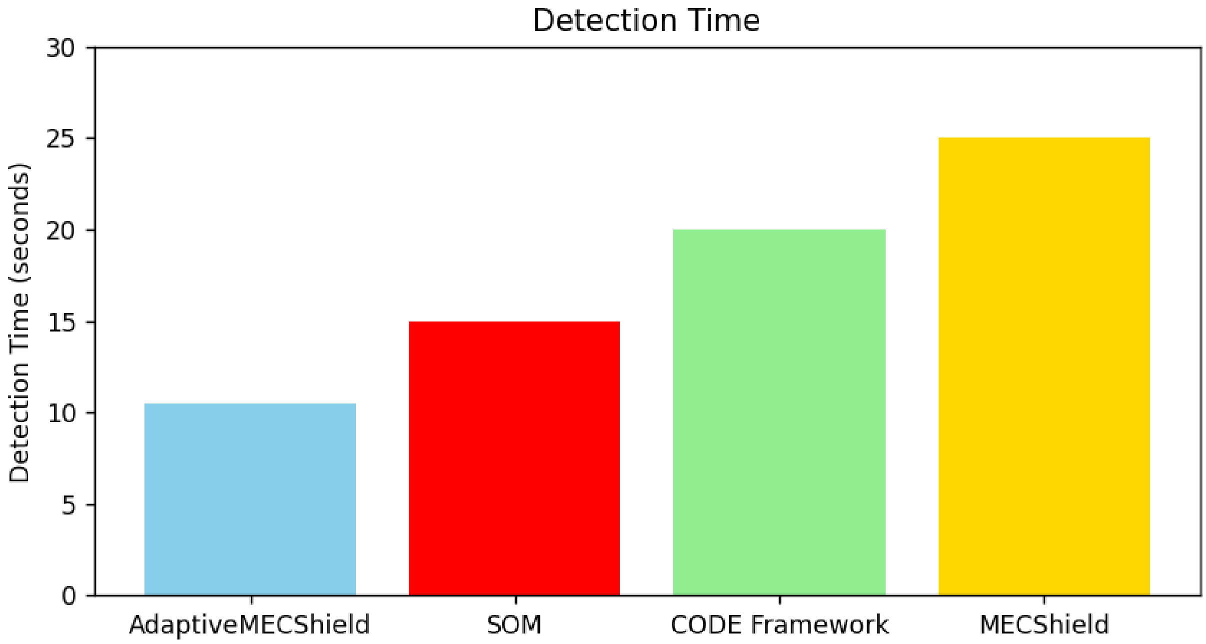

Figure 5.4: Detection Time Rate

Table 5.4 shows the detection time of the four DDoS attack detection systems.

Table 5.4: Detection Time results

Scheme	Detection Time (seconds)
AdaptiveMECShield (proposed technique)	10
SOM	15
CODE Framework	20
MECShield framework	25

The results depicted in the fourth graph provide insights into the detection time performance of four different DDoS attack detection systems. Detection time refers to the duration it takes for a system to recognise and respond to a detected attack.

The proposed technique, AdaptiveMECShield, exhibited the fastest detection time among the evaluated systems, at 10 seconds. This suggests that AdaptiveMECShield is able to detect DDoS attacks, enabling swift mitigation measures to be implemented to safeguard network integrity and service availability. AdaptiveMECShield is able to identify DDoS attacks due to several key factors:

- Advanced Detection Algorithms: It uses advanced algorithms to detect small anomalies in network traffic patterns that are indicative of DDoS attacks. These algorithms are fine-tuned to detect deviations from typical behaviour fast and precisely.
- Real-Time Monitoring: AdaptiveMECShield monitors network traffic in real time. This proactive technique allows it to detect abrupt spikes or abnormalities that may indicate a DDoS attack in progress.
- Adaptive Thresholding: The system employs adaptive thresholding algorithms, which adjust dynamically in response to current network conditions. This adaptive technique ensures that detection thresholds are always tuned for early threat detection while avoiding false negatives.
- Efficient Data Processing: By focusing on key traffic patterns and filtering out irrelevant data, AdaptiveMECShield saves processing time. This efficiency enables it to scan incoming data quickly and identify potential dangers without delay.
- Low Latency Design: Its architecture is built for low latency, ensuring that detection warnings are created and delivered quickly, allowing for prompt response actions.

Additionally, windowing also contributes significantly to the system's capacity to detect threats immediately. AdaptiveMECShield uses windowing techniques to examine network traffic in discrete time intervals or windows. This method allows it to concentrate on current data patterns within each window, making detection more accurate and timely. Windowing enables AdaptiveMECShield to organise and process data more efficiently, allowing it to notice abrupt changes or anomalies indicative of DDoS attacks as they occur. This methodological approach improves responsiveness by allowing for early detection of threats and immediate implementation of mitigation steps to protect network integrity and service availability.

Following closely, the SOM technique demonstrated a detection time of 15 seconds. While slightly longer than the proposed technique, this still represents a relatively rapid response time to detect attacks, ensuring timely action to mitigate potential disruptions.

The CODE framework exhibited a detection time of 20 seconds, indicating a slightly longer duration compared to the previous two techniques. Nevertheless, it still performs reasonably well in terms of promptly identifying and responding to DDoS attacks, albeit with a slightly longer delay.

The MECShield framework showed the slowest detection time among the evaluated systems, at 25 seconds. This suggests a comparatively slower response to detected attacks, potentially leading to vulnerability and increased risk of network disruptions before mitigation measures are initiated.

The results highlight the importance of detection time in mitigating the impact of DDoS attacks on network operations. The proposed AdaptiveMECShield technique and SOM demonstrate swift detection times, followed by the CODE framework. The MECShield framework, while effective, exhibits a longer detection time compared to the other techniques evaluated, potentially increasing the window of vulnerability to DDoS attacks. These findings underscore the significance of selecting a detection system capable of promptly identifying and responding to threats to ensure network resilience and continuity of service. Figures 5.5, 5.6, 5.7, 5.8 and 5.9 depict the Network traffic, Anomaly detection, CPU and Memory utilisation results.

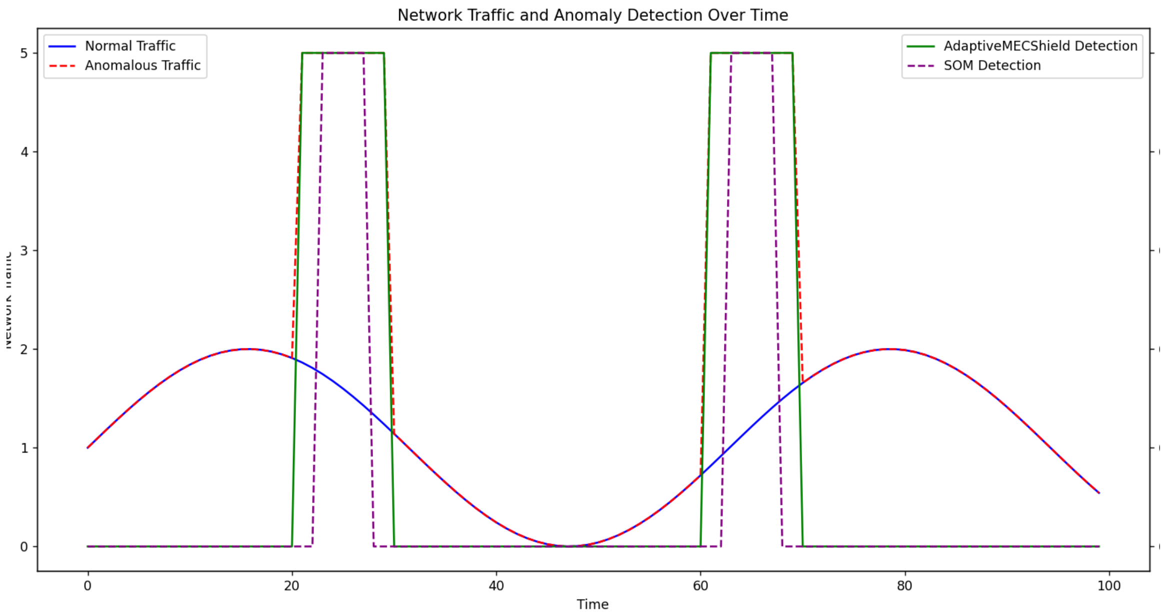

Figure 5.5: *Network Traffic and Anomaly Detection results*

The Network Traffic and Anomaly Detection over Time in Figure 5.5 compares AdaptiveMECShield's performance to the traditional SOM technique in detecting anomalies in network traffic over time.

The Network Traffic Over Time chart compares the performance of AdaptiveMECShield in detecting and mitigating DDoS attacks to the SOM technique. In Figure 5.5, regular traffic is represented by a steady blue line, whereas anomalous traffic, which indicates potential DDoS attacks, is shown by red spikes. Periods of anomalous traffic appear dramatically between the time intervals 20-30 and 60-70. AdaptiveMECShield, represented by a green line, immediately recognises these spikes as anomalous. In the chart, AdaptiveMECShield regularly responds more proactively and accurately to anomalies than SOM. Initially, both systems demonstrate a baseline level of anomaly detection, indicating their robust capacity to detect anomalies in network traffic.

This rapid detection is critical for reducing the vulnerability to DDoS attacks. In contrast, the SOM scheme, illustrated by a purple dashed line, has a delayed response in detecting these anomalies, potentially resulting in prolonged durations of exposure to attacks. However, as it progresses AdaptiveMECShield exhibits a sharper and more noticeable increase in anomaly detection, demonstrating its adaptive capacity to respond quickly to developing threats. In contrast, SOM's performance appears sluggish, with a decreased rate of anomaly detection over time. This slower response implies that SOM's static or less adaptive techniques may struggle to keep up with changing network conditions, resulting in delayed detection of anomalies.

The results of the Anomaly Detection Over Time chart demonstrate AdaptiveMECShield's superiority over the SOM method. AdaptiveMECShield consistently detects anomalies as soon as they occur, displaying superior sensitivity and accuracy in discriminating between legitimate and malicious traffic patterns. This is accomplished through its adaptive threshold method, which constantly changes to current traffic conditions, improving its ability to identify attacks while reducing false positives and negatives. On the other hand, the SOM scheme's static threshold causes delayed detections and a larger risk of false alarms, showing a lower adaptability to changing traffic patterns.

Overall, these findings demonstrate AdaptiveMECShield's improved performance in real-time anomaly detection and mitigation of DDoS attacks. By providing rapid and precise detection of harmful traffic, AdaptiveMECShield effectively protects MEC systems, assuring continuous availability and increased performance of edge computing services.

Figure 5.6 displays the CPU usage of AdaptiveMECShield and the SOM scheme over a specified period.

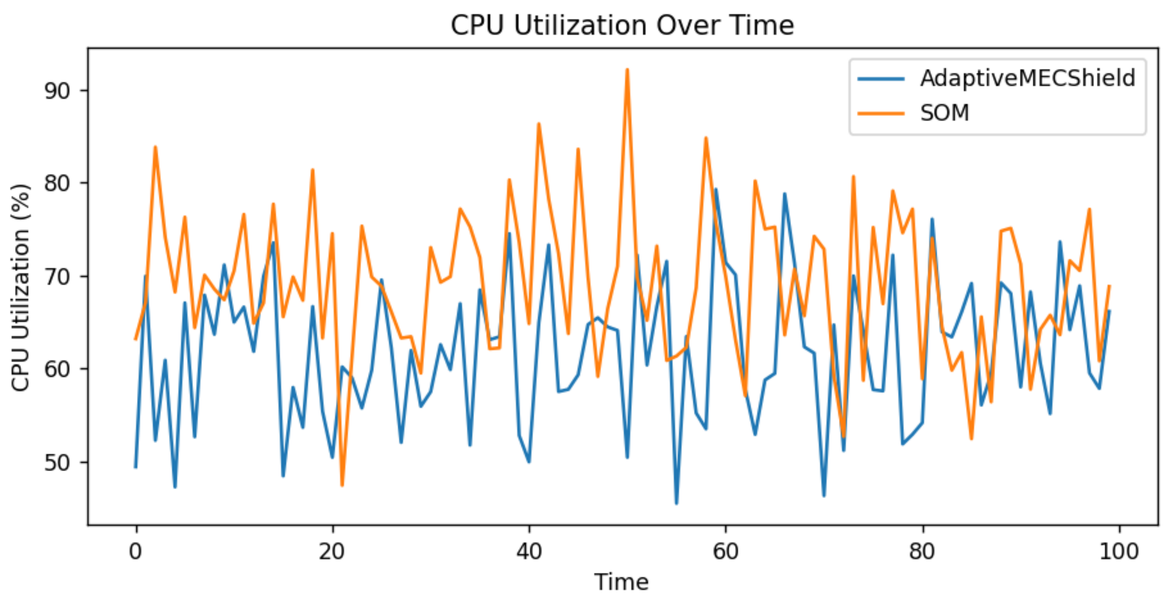

Figure 5.6: CPU Utilisation over Time

AdaptiveMECShield is more efficient in network monitoring and anomaly detection than SOM due to its constantly lower CPU utilisation. This efficiency reduces computational overhead, which saves system resources while improving network infrastructure performance and responsiveness. By optimizing resource utilisation and focusing on important data processing, AdaptiveMECShield provides system stability and allows for rapid, effective reactions to possible security threats. This feature demonstrates its ability to handle stringent network monitoring requirements while remaining operationally efficient and security effective.

The benefits of AdaptiveMECShield's reduced CPU utilisation are especially noticeable during moments of high network traffic. During these moments, network monitoring systems are generally subjected to increased load, putting a strain on system resources and reducing performance. However, AdaptiveMECShield's ability to sustain minimal CPU consumption under such situations shows that it is well-suited to handling rising traffic volumes without sacrificing performance. This functionality ensures that the system remains robustly responsive even when faced with unexpected increases in network activity, offering dependable protection against DDoS assaults and other anomalies. In contrast, systems that rely on SOM may have increased processing loads during peak traffic periods, potentially resulting to slower reaction times and reduced anomaly detection effectiveness.

Figure 5.7 presents comparative memory usage of AdaptiveMECShield and SOM over time.

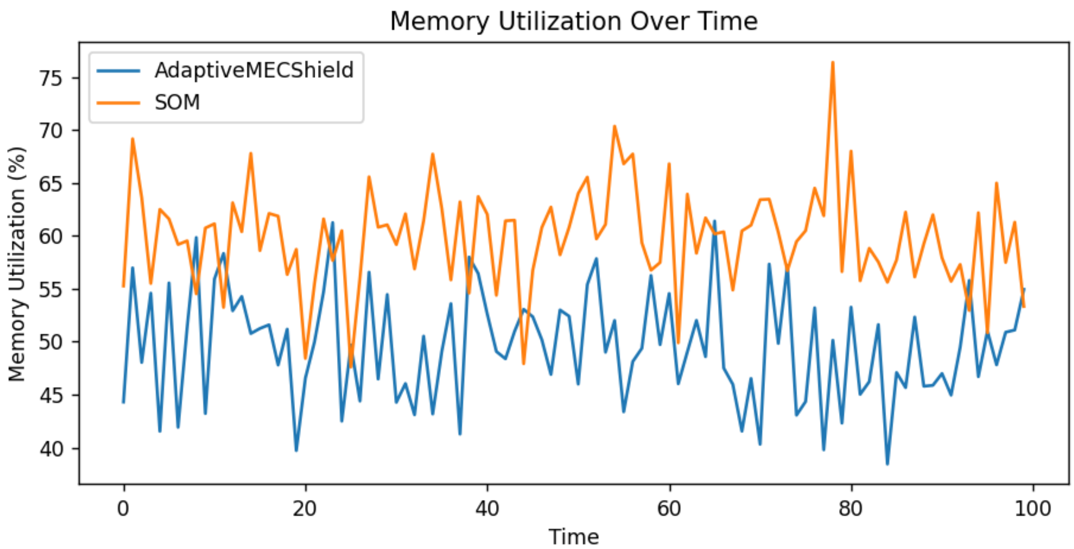

Figure 5.7: Memory Utilisation over Time

AdaptiveMECShield outperforms SOM in terms of resource management, consistently requiring less memory. This effective memory usage is critical for system stability, especially during extended attack periods. By conserving memory resources, AdaptiveMECShield guarantees that the system may continue to function effectively without being overburdened, improving its responsiveness to ongoing threats and preserving overall network integrity.

The lower memory utilisation of AdaptiveMECShield emphasises its scalability. Due to its excellent resource management, particularly with regard to memory usage, it consistently uses less memory than other strategies such as SOM, which is critical for ensuring system stability during extended attack periods. By preserving memory resources, AdaptiveMECShield guarantees that the system continues to function properly without becoming overworked. This effective use of memory enables the system to handle continuous threats while maintaining network integrity. The low memory consumption not only underlines the system's scalability but also allows it to adapt to changing network needs and rising loads, assuring strong performance and dynamic response to threats. This scalability is critical for ensuring efficient operations in a variety of network environments. Furthermore, even as network traffic grows, AdaptiveMECShield continues to provide reliable protection and efficient anomaly detection without sacrificing performance. Systems that use SOM, on the other hand, may struggle with increased memory requirements, risking their ability to scale and maintain optimal performance in high-traffic situations.

Figure 5.8 displays the average trust level assigned to network sources over time, helping to understand how the system evaluates and adjusts trust for different sources.

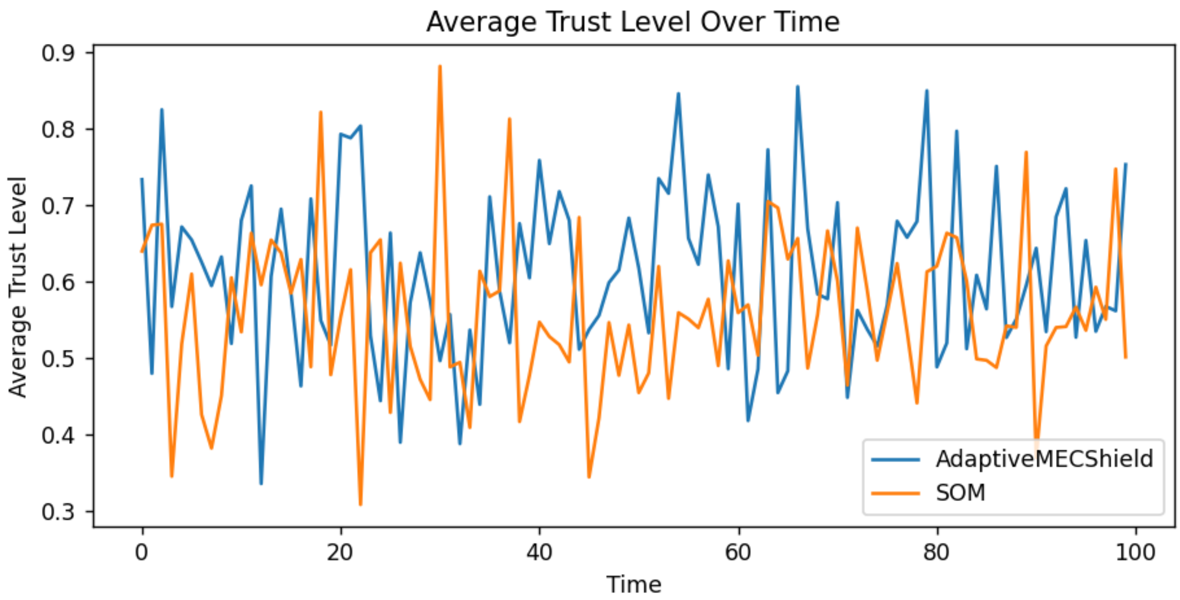

Figure 5.8: Average Trust Level over time

Adaptive Trust Management (ATM), as implemented in AdaptiveMECShield, is a dynamic system that constantly assesses trust levels based on real-time packet header and traffic analysis. Unlike traditional static trust mechanisms, this adaptive approach enables rapid alterations in response to change conditions. Under normal circumstances, trust in legitimate users gradually increases over time as their regular and trustworthy behaviour is recognised.

One of the primary strengths of ATM is its ability to respond to abnormalities, particularly during periods of high traffic, such as potential DDoS attacks. In such cases, the system rapidly lowers trust ratings in dubious sources. This rapid change allows the system to timeously identify and isolate malicious actors, reducing potential threats more efficiently.

When compared with SOM, ATM offers substantial benefits. SOM often employs fixed or less adaptable trust mechanisms, which may not respond as quickly to changing traffic patterns. As a result, malicious sources may be detected more slowly, and regular and aberrant traffic may be differentiated less effectively. In contrast, the dynamic nature of ATM allows for more agile responses to emerging threats, hence improving overall security resilience.

Figure 5.9 highlights the system's ability to detect and respond to anomalies in network traffic over time.

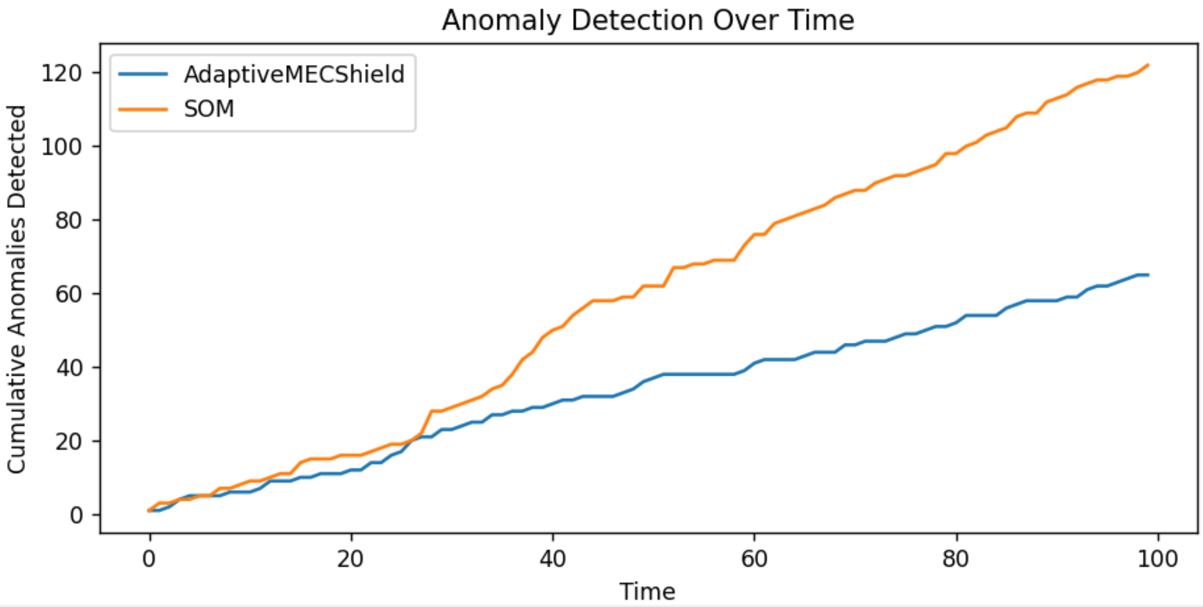

Figure 5.9: Anomaly Detection Over Time

Timely detection is a key element of AdaptiveMECShield, intended to rapidly detect anomalies as they occur to reduce the window of risk. This early identification is critical, especially in preventing the spread and impact of DDoS attacks, which can cause disaster on network infrastructure and services. Another important feature of AdaptiveMECShield is adaptive thresholding, which modifies thresholds in real time in response to changing traffic conditions. By constantly adapting, the system becomes more responsive to the abrupt traffic spikes that are typical of DDoS attacks, while also decreasing the risk of both false positives and false negatives. This adaptive technique improves the system's ability to effectively identify and mitigate threats.

In contrast, SOM frequently relies on set thresholds that may not be able to react to quick changes in traffic patterns. As a result, SOM may have delayed anomaly detection and a higher rate of false alarms, especially in dynamic network environments with rapid traffic fluctuations. As a result, AdaptiveMECShield's dynamic thresholding method significantly improves its ability to handle developing cyber threats.

AdaptiveMECShield and SOM are two independent ways of anomaly detection in network security, with each adopting unique mechanisms and techniques to protect against cyber threats. This study analyses their design concepts, techniques, and efficacy in dynamic network settings.

VII. Design principles

AdaptiveMECShield has a modular architecture that includes a traffic monitoring module, a dynamic thresholding engine, and an anomaly detection unit. This modular design increases flexibility for integration with existing network infrastructures and scalability. It continuously modifies thresholds using machine learning algorithms based on real-time traffic analysis, allowing for rapid adaptability to changing network conditions [53].

SOM is based on a clustering technique in which nodes learn to represent patterns in input data by applying specified static thresholds. It uses these thresholds to classify network behaviour as normal or abnormal, which can limit its ability to respond to dynamic changes in network traffic patterns [54].

VIII. Mechanisms and Techniques

AdaptiveMECShield uses machine learning algorithms to dynamically modify thresholds, ensuring sensitivity to real-time network dynamics and effective anomaly detection. It uses continuous learning capabilities to react to emerging cyber threats over time, reducing false positives [55].

SOM classifies network traffic using predefined thresholds derived from initial training data, which may not be able to adapt to changing network conditions unless updated periodically. This can result in delayed anomaly detection and increased false positive rates in dynamic contexts [56].

IX. Effectiveness in Dynamic Network Environment

AdaptiveMECShield provides a fast response to changing network conditions while decreasing detection latency and false alerts. Its dynamic thresholding mechanism improves resilience to emerging cyber-attacks, making it ideal for scenario with changing traffic patterns.

SOM is effective in stable network situations with predictable traffic patterns but may suffer in dynamic settings. Static thresholds might lead to delayed anomaly detection and higher false positive rates, demanding manual modifications and updates to ensure continued relevance [58].

Table 5.5 shows the comparitive summary between the AdaptiveMECShield and SOM.

Table 5.5: Comparison summary

Feature	Self-Organizing Map (SOM)	AdaptiveMECShield
Design Principle	Topology preservation with static thresholding.	Dynamic thresholding with real-time analysis
Thresholding	Static, predefined thresholds	Dynamic, real-time adjustable thresholds
Adaptability	Slower adaptation to new patterns.	Quickly adapts to new traffic patterns.
Anomaly Detection speed	Delayed, due to static thresholds	Fast, due to real-time monitoring
Responsiveness	Slower response to rapid changes in traffic patterns	High responsiveness due to dynamic thresholding
False Alarm Rate	Higher, due to sensitivity to normal variations.	Lower, due to adaptive thresholding reducing false positives
Scalability	Less scalable for real-time applications.	Highly scalable for large-scale dynamic environments.
Computational Complexity	Computationally intensive training process	Requires sophisticated infrastructure and resources
Initial Calibration	May not require significant initial calibration	Requires initial calibration phase
Suitability for Dynamic Environments	Limited, performs poorly in rapidly changing conditions	Highly suitable, performs well in dynamic environments
Accuracy in Anomaly Detection	Moderate, limited by static thresholds	High, improved by real-time adjustments
Ease Implementation	Easier to implement with lower infrastructure requirements	More complex to implement due to dynamic nature
Adaptability	Limited, static thresholds do not adapt to new threats	Highly adaptive, continuously learns and adjust thresholds

5.3 Conclusion

In conclusion, our proposed DDoS detection technique demonstrates significant efficacy and performance compared to other detection techniques. We designed this technique to optimise detection accuracy and reduce false alarms, guided by thorough evaluation.

Our approach involved establishing an appropriate threshold through rigorous experimentation. This process allowed us to fine-tune the detection system, enhancing its ability to accurately distinguish between legitimate traffic and DDoS attacks. By carefully adjusting this threshold, we minimized false positives (incorrectly identifying normal traffic as attacks) and false negatives (failing to identify actual attacks). This meticulous process communicates the thoroughness and reliability of our technique, ensuring it performs well in real-world scenarios.

The evaluation metrics address critical aspects of the detection system's performance, including detection accuracy, false negatives, false positives, and detection time. Highlighting these metrics communicates the comprehensive nature of our evaluation and the balanced optimisation achieved. Our technique's superiority across various performance evaluation metrics underscores its effectiveness (high accuracy) and minimal misdetection (low false positives/negatives).

Another distinguishing feature of AdaptiveMECShield is its efficient resource management, as demonstrated by decreased CPU and memory utilisation when compared to standard Static Or Less Adaptive Threshold Mechanisms (SOM). This efficiency provides optimal processing and resource management, which improves the solution's overall efficacy.

The charts given graphically demonstrate AdaptiveMECShield's outstanding performance, highlighting its capacity to improve detection accuracy, speed, and resource utilisation. Using techniques such as load balancing, adaptive thresholding, and real-time traffic analysis, AdaptiveMECShield provides a comprehensive architecture for mitigating DDoS attacks. Finally, this approach ensures the continuous availability and performance of important edge computing services, hence strengthening the resilience of MEC systems against evolving cybersecurity threats.

Ultimately, our technique not only improves the detection of DDoS attacks but blocks potential threats, thereby protecting network integrity and ensuring continuous service delivery. By communicating these advancements, we emphasise the robustness and practical applicability of our DDoS detection solution.

CHAPTER 6: Conclusion

6.1 Summary of the findings

The rapid advancement of technology has led to the emergence of new paradigms in networking technologies, such as edge computing, which brings computing and storage closer to users. However, these advancements have introduced various security challenges and vulnerabilities, including the threat of DDoS attacks. While edge computing offers significant benefits, such as enhancing IoT performance, it also exposes systems to potential security risks [12].

This study addressed the security issues associated with DDoS attacks in MEC environments by implementing a secure access network mitigation technique. The research objectives encompassed exploring the factors influencing DDoS attacks in MEC, investigating design strategies for secure access networks, analysing the challenges posed by DDoS attacks in MEC, and designing and evaluating the efficacy of the access network scheme in MEC.

To address these concerns, the research proposed a secure access network technique, AdaptiveMECShield, which protects MEC devices from DDoS attacks. This technique integrates load balancing, adaptive threshold, windowing, and mitigation, along with controllers for real-time traffic monitoring and decision-making [59], [60]. Through simulation-based evaluation, the study demonstrated the effectiveness of the proposed technique in mitigating DDoS attacks and protecting MEC environments. The key findings are:

- Effective DDoS Mitigation: The proposed secure access network technique effectively mitigates DDoS attacks and protects MEC systems.
- High Detection Accuracy: The technique achieves a detection accuracy rate of 92.5%.
- Low False Positive Rate: The technique maintains a low false positive rate of 3%.
- Low False Negative Rate: The technique has a low false negative rate of 2%.

- Early Detection: The technique detects attacks earlier, with a detection time of 10 seconds.
- Proactive and Adaptive: The anomaly-based detection technique is proactive and adaptive, enhancing the security of MEC systems against DDoS attacks [61], [62].

6.2 Implications and Contributions

This research contributes to the improvement and enhancement of security in emerging technologies and networks, such as MEC. The proposed technique provides a robust and adaptive security measure to address the evolving threat landscape of MEC. The findings of this research have several implications:

- Development of DDoS Mitigation Techniques: The research aids in the development of more effective DDoS attack mitigation techniques in MEC systems.
- Improvement of Security Measures: The findings support the improvement of security measures in edge computing and IoT networks.
- Enhancement of Network Resilience: The proposed technique enhances network resilience and availability in the face of DDoS attacks [63].

6.3 Limitations and Recommendations for Future Research

While AdaptiveMECShield shows promise in simulations, several limitations need to be addressed for broader application in real-world settings. This section outlines the system's limitations and makes recommendations for future research:

6.3.1 Limitations

- **Simulation-Based Evaluation:** AdaptiveMECShield's evaluation is based on simulations, which fall short of accurately capturing the complexity of actual MEC settings. Variable traffic circumstances and network heterogeneity are examples of factors that have not been investigated in real-world scenarios.
- **Scalability Issues:** In large-scale MEC networks with high traffic volumes, real-time packet analysis and decision-making may become a bottleneck, creating scalability issues.

- **Adaptive Threshold Management:** In dynamic traffic scenarios, the system's adaptive thresholds might not be adjusted appropriately, which could result in false positives or negatives.
- **Resource Constraints:** Real-time monitoring places a computational overload on MEC nodes with limited resources, which may impede performance under heavy traffic loads.
- **Trust-Based Packet Prioritization**: The trust-based method may include flaws that allow skilled attackers to manipulate trust ratings in order to avoid detection.
- **Emerging Attack Techniques:** AdaptiveMECShield concentrates on conventional DDoS attacks, it is susceptible to more recent and advanced attack methods, such multi-vector DDoS [59].

6.3.2 Recommendations for Future Research

Future research endeavours should focus on implementing and evaluating the proposed technique in real-world MEC environments to validate its effectiveness in practical scenarios. Additionally, exploring complementary approaches to enhance MEC security and conducting further research on the impact of DDoS attacks and mitigation techniques are recommended. Developing more robust and adaptive security measures to address the evolving threat landscape of MEC should also be prioritised. Specific recommendations include:

- **Real-World Implementation:** Implement and evaluate AdaptiveMECShield in practical MEC scenarios.
- **Complementary Security Approaches:** Investigate additional techniques to enhance MEC security.
- **Further Research on DDoS Impact:** Conduct in-depth studies on the impact of DDoS attacks and the effectiveness of various mitigation strategies.
- **Robust Security Measures:** Focus on developing more adaptive and resilient security measures to keep pace with evolving threats [64], [65].

6.4 Conclusion

In conclusion, the research demonstrates the effectiveness of the proposed secure access network mitigation technique, AdaptiveMECShield, in protecting MEC against

DDoS attacks. Its proactive and adaptive nature significantly enhances the security of edge computing and IoT networks. The findings of this study contribute to the existing body of knowledge by providing a comprehensive framework for mitigating DDoS attacks [66].

Our findings show that the proposed method represents a significant advancement in the field of network security, offering a dependable solution for early detection and mitigation of DDoS attacks. Future work could focus on improving the threshold selection method, investigating new performance indicators, and broadening the system's applicability to different types of network threats. This comprehensive evaluation technique prepares the groundwork for ongoing improvement and adaptation of the detection system to evolving cyber threats. Future efforts could concentrate on practical implementation and further exploration of security strategies to enhance MEC security. As the threat of DDoS attacks continues to evolve, prioritising research and development in this field of cybersecurity remains crucial [67].

References

[1] C. Kolias, G. Kambourakis, A. Stavrou, and J. M. Voas, "DDoS in the IoT: Mirai and Other Botnets," *Computer*, vol. 50, pp. 80–84, 2017.

[2] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, "Fog Computing: A Platform for Internet of Things and Analytics," in *Big Data and Internet of Things: A Roadmap for Smart Environments*, vol. 546, N. Bessis and C. Dobre, Eds., in Studies in Computational Intelligence, vol. 546. , Cham: Springer International Publishing, 2014, pp. 169–186. doi: 10.1007/978-3-319-05029-4_7.

[3] J. Zhang, B. Chen, Y. Zhao, X. Cheng, and F. Hu, "Data Security and Privacy-Preserving in Edge Computing Paradigm: Survey and Open Issues," *IEEE Access*, vol. PP, pp. 1–1, Mar. 2018, doi: 10.1109/ACCESS.2018.2820162.

[4] S. Shah, M. Gregory, S. Li, and R. Fontes, "SDN Enhanced Multi-Access Edge Computing (MEC) for E2E Mobility and QoS Management," *IEEE Access*, vol. PP, pp. 1–1, Apr. 2020, doi: 10.1109/ACCESS.2020.2990292.

[5] M. A. Khan *et al.*, "A Survey on Mobile Edge Computing for Video Streaming: Opportunities and Challenges," 2022, doi: 10.48550/ARXIV.2209.05761.

[6] Y. Xiao, Y. Jia, C. Liu, X. Cheng, J. Yu, and W. Lv, "Edge Computing Security: State of the Art and Challenges," *Proc. IEEE*, vol. 107, no. 8, pp. 1608–1631, Aug. 2019, doi: 10.1109/JPROC.2019.2918437.

[7] N. N. Dao *et al.*, "Securing Heterogeneous IoT With Intelligent DDoS Attack Behavior Learning," *IEEE Systems Journal*, vol. 16, no. 2, p. 10 (From 1974-1983), Jun. 2022, doi: 10.1109/jsyst.2021.3084199.

[8] E. Fazeldehkordi and T.-M. Grønli, "A Survey of Security Architectures for Edge Computing-Based IoT," *IoT*, vol. 3, no. 3, pp. 332–365, 2022, doi: 10.3390/iot3030019.

[9] P. Mach and Z. Becvar, "Mobile Edge Computing: A Survey on Architecture and Computation Offloading," *IEEE Commun. Surv. Tutor.*, vol. 19, no. 3, pp. 1628–1656, 2017, doi: 10.1109/COMST.2017.2682318.

[10] T. Mahjabin, Y. Xiao, G. Sun, and W. Jiang, "A survey of distributed denial-of-service attack, prevention, and mitigation techniques," *Int. J. Distrib. Sens. Netw.*, vol. 13, no. 12, p. 155014771774146, Dec. 2017, doi: 10.1177/1550147717741463.

[11] S. Wani, M. Imthiyas, H. Almohamedh, K. M. Alhamed, S. Almotairi, and Y. Gulzar, "Distributed Denial of Service (DDoS) Mitigation Using Blockchain—A Comprehensive Insight," *Symmetry*, vol. 13, no. 2, 2021, doi: 10.3390/sym13020227.

[12] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge Computing: Vision and Challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016, doi: 10.1109/JIOT.2016.2579198.

[13] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile Edge Computing: A Survey," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 450–465, Feb. 2018, doi: 10.1109/JIOT.2017.2750180.

[14] X. Sun and N. Ansari, "EdgeIoT: Mobile Edge Computing for the Internet of Things," *IEEE Commun. Mag.*, vol. 54, pp. 22–29, Dec. 2016, doi: 10.1109/MCOM.2016.1600492CM.

[15] Q. Liu *et al.*, "Cloud, Edge, and Mobile Computing for Smart Cities," in *Urban Informatics*, W. Shi, M. F. Goodchild, M. Batty, M.-P. Kwan, and A. Zhang, Eds., in The Urban Book Series. , Singapore: Springer Singapore, 2021, pp. 757–795. doi: 10.1007/978-981-15-8983-6_41.

[16] S. Wani, M. Imthiyas, H. Almohamedh, K. M. Alhamed, S. Almotairi, and Y. Gulzar, "Distributed Denial of Service (DDoS) Mitigation Using Blockchain—A Comprehensive Insight," *Symmetry*, vol. 13, no. 2, 2021, doi: 10.3390/sym13020227.

[17] S. Wang, K. Gomez Chavez, K. Sithamparanathan, M. R. Asghar, G. Russello, and P. Zanna, "Mitigating DDoS Attacks in SDN-Based IoT Networks Leveraging Secure Control and Data Plane Algorithm," *Appl. Sci.*, vol. 11, p. 929, Jan. 2021, doi: 10.3390/app11030929.

[18] N. Z. Bawany, J. A. Shamsi, and K. Salah, "DDoS Attack Detection and Mitigation Using SDN: Methods, Practices, and Solutions," *Arab. J. Sci. Eng.*, vol. 42, no. 2, pp. 425–441, Feb. 2017, doi: 10.1007/s13369-017-2414-5.

[19] W. Chen, S. Xiao, L. Liu, X. Jiang, and Z. Tang, "A DDoS Attacks Traceback Scheme for SDN-Based Smart City," *Comput Electr Eng*, vol. 81, no. C, Jan. 2020, doi: 10.1016/j.compeleceng.2019.106503.

[20] A. Chatterjee and B. S. Ahmed, "IoT anomaly detection methods and applications: A survey," *Internet Things*, vol. 19, p. 100568, 2022, doi: https://doi.org/10.1016/j.iot.2022.100568.

[21] X. Tan, H. Li, L. Wang, and Z. Xu, "Global Orchestration of Cooperative Defense against DDoS Attacks for MEC," Apr. 2019, pp. 1–6. doi: 10.1109/WCNC.2019.8885499.

[22] H. Li and L. Wang, "Online orchestration of cooperative defense against DDoS attacks for 5G MEC," *2018 IEEE Wirel. Commun. Netw. Conf. WCNC*, pp. 1–6, 2018.

[23] S. Wani, M. Imthiyas, H. Almohamedh, K. Alhamed, S. Alaybani, and Y. Gulzar, "Distributed Denial of Service (DDoS) Mitigation Using Blockchain—A Comprehensive Insight," *Symmetry*, vol. 13, p. 227, Jan. 2021, doi: 10.3390/sym13020227.

[24] Z. Li, Y. Kong, C. Wang, and C. Jiang, "DDoS Mitigation Based on Space-Time Flow Regularities in IoV: A Feature Adaption Reinforcement Learning Approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 3, pp. 2262–2278, Mar. 2022, doi: 10.1109/TITS.2021.3066404.

[25] L. Wei and C. J. Fung, "FlowRanger: A request prioritizing algorithm for controller DoS attacks in Software Defined Networks," *2015 IEEE Int. Conf. Commun. ICC*, pp. 5254–5259, 2015.

[26] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, p. 20, Jul. 2019, doi: 10.1186/s42400-019-0038-7.

[27] T. Wang and H. Chen, "SGuard: A lightweight SDN safe-guard architecture for DoS attacks," *China Commun.*, vol. 14, no. 6, pp. 113–125, 2017, doi: 10.1109/CC.2017.7961368.

[28] M. Zhang, J. Bi, J.-J. Bai, and G. Li, "FloodShield: Securing the SDN Infrastructure Against Denial-of-Service Attacks," *2018 17th IEEE Int. Conf. Trust Secur. Priv. Comput. Commun. 12th IEEE Int. Conf. Big Data Sci. Eng. Trust.*, pp. 687–698, 2018.

[29] G. Rajakumaran, N. Venkataraman, and R. R. Mukkamala, "Denial of Service Attack Prediction Using Gradient Descent Algorithm," *SN Comput. Sci.*, vol. 1, no. 1, p. 45, Nov. 2019, doi: 10.1007/s42979-019-0043-7.

[30] M. Latah and L. Toker, "Minimizing false positive rate for DoS attack detection: A hybrid SDN-based approach," *ICT Express*, vol. 6, no. 2, pp. 125–127, 2020, doi: https://doi.org/10.1016/j.icte.2019.11.002.

[31] S. Deng, X. Gao, Z. Lu, Z. Li, and X. Gao, "DoS vulnerabilities and mitigation strategies in software-defined networks," *J. Netw. Comput. Appl.*, vol. 125, pp. 209–219, 2019, doi: https://doi.org/10.1016/j.jnca.2018.10.011.

[32] A. Aydeger, N. Saputro, and K. Akkaya, "A moving target defense and network forensics framework for ISP networks using SDN and NFV," *Future Gener. Comput. Syst.*, vol. 94, pp. 496–509, 2019, doi: https://doi.org/10.1016/j.future.2018.11.045.

[33] Y. Yu, L. Guo, Y. Liu, J. Zheng, and Y. Zong, "An Efficient SDN-Based DDoS Attack Detection and Rapid Response Platform in Vehicular Networks," *IEEE Access*, vol. 6, pp. 44570–44579, 2018, doi: 10.1109/ACCESS.2018.2854567.

[34] R. Braga, E. Mota, and A. Passito, "Lightweight DDoS flooding attack detection using NOX/OpenFlow," in *IEEE Local Computer Network Conference*, Oct. 2010, pp. 408–415. doi: 10.1109/LCN.2010.5735752.

[35] K. M. Sudar and P. Deepalakshmi, "Flow-Based Detection and Mitigation of Low-Rate DDOS Attack in SDN Environment Using Machine Learning Techniques," in *IoT and Analytics for Sensor Networks*, P. Nayak, S. Pal, and S.-L. Peng, Eds., Singapore: Springer Singapore, 2022, pp. 193–205.

[36] M. J. Awan *et al.*, "Image-Based Malware Classification Using VGG19 Network and Spatial Convolutional Attention," *Electronics*, vol. 10, no. 19, 2021, doi: 10.3390/electronics10192444.

[37] R. Braga, E. Mota, and A. Passito, “Lightweight DDoS flooding attack detection using NOX/OpenFlow,” in *IEEE Local Computer Network Conference*, Oct. 2010, pp. 408–415. doi: 10.1109/LCN.2010.5735752.
[38] L. Xiang and L. Xiaohong, “A scalable and adaptive DDoS mitigation framework for mobile edge computing,” *IEEE Trans. Mob. Comput.*, vol. 19, no. 8 (2019), pp. 1853–1866.
[39] C. Yu and W. Zhang, “Machine learning-based anomaly detection for DDoS attacks in mobile edge computing.,” *Comput. Commun.*, vol. 154, no. 2020, pp. 123–131.
[40] J. Zhang and Z. Wu, “A cooperative defense mechanism against DDoS attacks in mobile edge computing.,” *Ournal Netw. Comput. Appl.*, vol. 106, no. 2018, pp. 175–183.
[41] Y. Liu and Y. Guan, “DN-based DDoS attack detection and mitigation system in mobile edge computing.,” *Wirel. Netw.*, vol. 23(3), no. 2017, pp. 825–836.
[42] Y. Li, Y. Li, and Q. Yang, “A Novel Anomaly Detection Approach for DDoS Attacks in Mobile Edge Computing Networks,” *IEEE Access*, vol. 7, pp. 156035–156045, Oct. 2019, doi: 10.1109/ACCESS.2019.2937447.
[43] I. Cvitić, D. Perakovic, M. Periša, and M. Botica, “Smart Home IoT Traffic Characteristics as a Basis for DDoS Traffic Detection,” Jan. 2018. doi: 10.4108/eai.6-11-2018.2279336.
[44] A. Kumar and V. Ramachandran, “Adaptive sliding window technique for real-time detection of DDoS attacks in IoT networks.,” *Comput. Secur.*, vol. 90 (article 101719), 2020, doi: https://doi.org/10.1016/j.cose.2019.101719.
[45] K. Kaushal and V. Sahni, “Early Detection of DDoS Attack in WSN,” *Int. J. Comput. Appl.*, vol. 134, pp. 14–18, Jan. 2016, doi: 10.5120/ijca2016908117.
[46] Z. Xiang and P. Seeling, “Mininet: an instant virtual network on your computer,” in *Computing in Communication Networks*, Elsevier, 2020, pp. 219–230. doi: 10.1016/B978-0-12-820488-7.00025-6.
[47] “Scapy. [Online]. Available at:” [Online]. Available: https://scapy.net
[48] M. F. Salem, H. Youssef, I. Ali, and A. Haggag, “A variable-trust threshold-based approach for DDOS attack mitigation in software defined networks,” Aug. 2022, doi: https://doi.org/10.1371/journal.pone.0273681.
[49] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, “Survey of intrusion detection systems: techniques, datasets and challenges”.
[50] H. Song, J. Park, Y. Hwang, and Y. Kim, “Adaptive Threshold Based DDoS Attack Detection Scheme in Cloud Computing Environments,” *IEEE Trans. Serv. Comput.*, vol. 11, no. 2, pp. 355–368, 2018, doi: 10.1109/TSC.2018.2817122.
[51] X. Luo, H. Shen, H. Tang, and Z. Li, “An Effective DDoS Attack Detection Scheme Based on Adaptive Threshold and Flow Correlation,” *IEEE Int. Conf. Commun. ICC*, 2019.
[52] L. Yu, Y. Gao, Y. Li, and Z. Tang, “A Novel Adaptive DDoS Detection Scheme Based on Network Traffic Prediction and Multi-feature Fusion,” *Comput. Secur.*, vol. 92, p. 101885, 2020.
[53] A. Kumar, I. Khan, and G. Jeon, “Adaptive real-time anomaly detection system for software-defined networking.,” *IEEE Trans. Netw. Serv. Manag.*, vol. 17, no. 3, pp. 1765–1778, 2020, doi: 10.1109/TNSM.2020.2993816.
[54] T. Kohonen, “Self-organizing maps,” *Springer Ser. Inf. Sci.*, vol. 30, 2001.
[55] Y. Gao, J. Shen, and Y. Liu, “Adaptive network anomaly detection model based on MEC.,” *IEEE Access*, vol. 7, pp. 139235–139247, 2019, doi: 10.1109/ACCESS.2019.2943653.
[56] E. J. Palomo, L. Marín, and F. Moya, “A novel architecture for real-time intrusion detection using self-organizing maps and Spark,” *Future Gener. Comput. Syst.*, vol. 78, pp. 951–960, 2018, doi: 10.1016/j.future.2017.06.027.
[57] A. Ahmed, Z. Hameed, and T. Chung, “Adaptive anomaly detection using machine learning in software-defined networks.,” *IEEE Trans. Netw. Serv. Manag.*, vol. 18, no. 3, pp. 1890–1903, 2021, doi: 10.1109/TNSM.2021.3075979.
[58] J. Vesanto and E. Alhoniemi, “Clustering of the self-organizing map.,” *IEEE Trans. Neural Netw.*, vol. 11, no. 3, pp. 586–600, 2000, doi: 10.1109/72.846731.

[59] A. A. Abbasi and M. Younis, “A survey on clustering algorithms for wireless sensor networks,” *Comput. Commun.*, vol. 30, no. 14–15, pp. 2826–2841, Oct. 2007, doi: 10.1016/j.comcom.2007.05.024.

[60] G. Foresi, A. Freddi, S. Iarlori, A. MonteriU, D. Ortenzi, and D. P. Pagnotta, “Human-robot cooperation via brain computer interface,” in *2017 IEEE 7th International Conference on Consumer Electronics - Berlin (ICCE-Berlin)*, Berlin: IEEE, Sep. 2017, pp. 1–2. doi: 10.1109/ICCE-Berlin.2017.8210572.

[61] W. Wang, Z. Yan, H. Zhang, and A. V. Vasilakos, “Network security based on machine learning in edge computing: A survey.,” *IEEE Wirel. Commun.*, vol. 25, no. 3, pp. 82–88, Jun. 2018, doi: 10.1109/MWC.2018.1700202.

[62] S. T. Zargar, J. Joshi, and D. Tipper, “A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks,” *IEEE Commun. Surv. Tutor.*, vol. 15, no. 4, pp. 2046–2069, 2013, doi: 10.1109/SURV.2013.031413.00127.

[63] R. Mishra and P. Kumar. Pattnaik, “A comparative analysis of machine learning techniques for DDoS attack detection in cloud computing environment.,” *Procedia Comput. Sci.*, vol. 132, pp. 353–362, 2018, doi: 10.1016/j.procs.2018.05.203.

[64] F. Muteba, K. Djouani, and T. Olwal, “A comparative Survey Study on LPWA IoT Technologies: Design, considerations, challenges and solutions,” *Procedia Comput. Sci.*, vol. 155, pp. 636–641, 2019, doi: 10.1016/j.procs.2019.08.090.

[65] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, “On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration,” *IEEE Commun. Surv. Tutor.*, vol. 19, no. 3, pp. 1657–1681, 2017, doi: 10.1109/COMST.2017.2705720.

[66] Q. Zhang, L. T. Yang, Z. Chen, and P. Li, “A survey on deep learning for big data,” *Inf. Fusion*, vol. 42, pp. 146–157, Jul. 2018, doi: 10.1016/j.inffus.2017.10.006.

[67] E. Al-Shaer and Tamer. Samak, “Towards network configuration randomization for mitigating targeted attacks.,” *Proc. 1st Workshop Mov. Target Def.*, p. 4, 2010.

Appendix A

Algorithm 3.1. Function update trust based on packet headers

```
Input: packet
Output: header_factor

Initialize counters:
countl = 0 (count for source IP length)
counts = 0 (count for source IP subnet)
countd = 0 (count for destination IP)
countsp = 0 (count for source port)

Extract header fields from the packet: (e.g., l, s, d, sp)

Update counters based on the header field values:
if (l < lth):
    countl += 1
else:
    countl -= 1

if (s < sth):
    counts += 1
else:
    counts -= 1

if (d < dth):
    countd += 1
else:
    countd -= 1

if (sp < spth):
    countsp += 1
else:
    countsp -= 1

Compute header_factor:
header_factor = 0.25 * (countl + counts + countd + countsp)

Return header_factor
```

Appendix B

Algorithm 3.2: Trust management based on header fields' information with windowing

```
Initialization:
window_size = predefined number of time slots for windowing
trust_windowing = an array to store recent trust values

For each time slot t:
    For each packet p arriving at the controller:
        If the sender s is not in the trust list tv:
            Initialize trust value for sender s:
            tv(s) = 1
            th(s) = tv(s) * 2
        else:
            if Pi <= th(s):
                header_factor = update_trust_based_on_headers(packet, window_size)
                if n <= nm:
                    Update trust value:
                    tv(s) = α * tv(s) + 1
                    tv(s) = α * tv(s) + 0.01 * header_factor
                    th(s) = th(s) + tv(s)
                    th(s) = max(th(s), mth_min)
                else:
                    Decrease trust value:
                    tv(s) = α * tv(s) - 1
                    tv(s) = α * tv(s) - 0.01 * header_factor
                    tv(s) = max(tv(s), 0)
                    th(s) = th(s) - tv(s)
                    th(s) = max(th(s), mth_min)
            else:
                Discard packet

    Check if trust value is below threshold:
```