Global Scientific JOURNALS

GSJ: Volume 12, Issue 7, July 2024, Online: ISSN 2320-9186
www.globalscientificjournal.com

DEVELOPING AUTONOMOUS REMEDIATION STRATEGIES FOR NETWORK SECURITY.

A Thesis Presented For the Master of Science Degree

Africa Centre of Excellence on Technology Enhanced Learning (*ACETEL*)

DECLARATION

I hereby declare that this thesis is my original work, conducted under the supervision of Dr Saheed Kayode. All sources of information and references used have been duly acknowledged and cited.

I take full responsibility for the content and findings presented in this thesis.

Jabir Abbas Sambo

CERTIFICATE OF APPROVAL OF THESIS OR DISSERTATION

Developing Autonomous Remediation Strategies for Network Security.

By

Jabir Abbas Sambo

Graduate Advisory Committee:

_____
Name

_____
Name

_____
Name

_____
Name

DEDICATION

This thesis is dedicated to my family, whose unwavering love and support have been the driving force behind my pursuit of knowledge. Their encouragement and belief in my abilities have been instrumental in shaping my academic journey. I am grateful for their sacrifices, understanding, and constant motivation throughout this endeavor.

I would also like to dedicate this work to my advisors and mentors, whose guidance, expertise, and valuable insights have shaped my research and challenged me to push the boundaries of my knowledge. Their dedication and commitment to academic excellence have been inspiring and instrumental in my growth.

Finally, I extend my heartfelt appreciation to all those who have contributed to this thesis, whether through their assistance, encouragement, or intellectual discussions. Your contributions have enriched my understanding and made this research possible.

This thesis is a culmination of the collective efforts, support, and belief of all those who have accompanied me on this journey. I dedicate it to each and every one of you, with profound gratitude and appreciation.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Dr. Saheed Kayode, my advisor and mentor, for his unwavering support and guidance throughout the duration of this research. His expertise, valuable insights, and continuous encouragement have been instrumental in shaping the direction of this thesis.

I am also deeply grateful to the faculty members of Africa Centre of Excellence on Technology Enhanced Learning *(ACETEL),* whose dedication to teaching and research has enriched my academic journey. Their intellectual contributions and constructive feedback have greatly influenced the development of this thesis.

I extend my heartfelt appreciation to my family and friends for their unwavering support, encouragement, and understanding. Their belief in me and constant encouragement have been a source of strength and motivation during challenging times.

Furthermore, I would like to express my gratitude to all the participants and organizations who generously contributed their time and resources for the purpose of this study. Their cooperation and willingness to share their experiences have been invaluable in shaping the outcomes of this research.

Finally, I would like to acknowledge the contributions of all those who have directly or indirectly supported me in this endeavor. Your encouragement, assistance, and belief in my abilities have been crucial in making this thesis a reality.

Thank you all for being an integral part of this journey.

TABLE OF CONTENTS

ABSTRACT

The objective of this research was to develop an innovative Autonomous Network Remediation System (ANRS) designed to address the limitations of current network security systems. Leveraging advanced machine learning algorithms, the ANRS aims to achieve automated anomaly detection, threat classification, and remediation within a secure and scalable architecture. The research methodology involved designing the ANRS with key components including Network Monitoring Agents (NMAs), a Centralized Anomaly Detection Engine (CADE), a Policy Enforcement Module (PEM), and a Self-healing Module (SHM). These components work together to collect, analyze, and respond to network traffic data, enforcing security policies and initiating remediation actions as needed. Evaluation of the ANRS was conducted using the CICIDS2017 dataset, specifically the Thursday-WorkingHours-Morning-WebAttacks.pcap_ISCX.csv file. The system was tested for various performance metrics, including anomaly detection accuracy, threat classification precision and recall, remediation effectiveness, policy enforcement consistency, system security, and scalability. The results indicated a high anomaly detection accuracy of 99%, with precise and timely classification and remediation capabilities. The ANRS demonstrated a precision of 1.00 for benign traffic and varied precision for different types of web attacks, with an overall system accuracy of 99%. The findings highlight the ANRS's superiority over existing network remediation systems, particularly in terms of detection accuracy, response speed, and policy enforcement consistency. The automated nature of the ANRS reduces the need for manual intervention, enhancing operational efficiency and allowing network administrators to focus on strategic tasks. Additionally, the system's scalability and minimal resource utilization make it suitable for deployment in diverse network environments. However, certain limitations were identified, such as the need for improved classification of rare attack types. Future research directions include refining the machine learning models, enhancing integration with legacy systems, and exploring broader application scenarios such as IoT and edge computing environments.

LIST OF TABLES AND FIGURES

TABLES

FIGURES

Chapter 1: Introduction

1.1 Background to the Study

Network security plays an important role in safeguarding the integrity, confidentiality, and availability of data and resources within any IT infrastructure (Qadir & Quadri, 2016). In today's digital landscape, where cyber threats are constantly evolving and becoming more sophisticated, the importance of robust network security measures cannot be overstated (Johnson, 2016). The threat landscape is continuously evolving, with cybercriminals employing increasingly advanced tactics to breach networks, steal sensitive information, disrupt operations, and cause financial or reputational damage (Corradini & Corradini., 2020). Threat actors may utilize malware, phishing attacks, ransom ware, or social engineering techniques to exploit vulnerabilities in network systems and gain unauthorized access (Narwal et al., 2019). Techniques such as zero-day exploits, polymorphic malware, and advanced persistent threats (APTs) have also been used in recent years to bypass traditional security defenses. These attacks are often highly targeted, persistent, and difficult to detect, requiring organizations to adopt proactive security measures to mitigate the risks effectively (Sharma et al., 2023).

Traditionally, network security measures have focused on preventive measures such as firewalls, intrusion detection systems (IDS), and encryption protocols to deter and block potential threats from infiltrating network environments (Chaabouni et al., 2019). While these measures are essential for safeguarding against known vulnerabilities and attack vectors, they may not be sufficient to defend against rapidly evolving and sophisticated cyber threats (Butun & Song., 2019). Furthermore, the demand for skilled cybersecurity professionals far outpaces the supply. Organizations often lack the expertise to effectively manage their network security posture, leaving them susceptible to attacks (Mughal, 2019).

One area of network security that is getting so much attention is the concept of autonomous remediation strategies (Neshenko et al., 2019). Autonomous remediation refers to the ability of a network security system to detect, analyze, and respond to security incidents automatically, without human intervention. Unlike traditional manual remediation processes, which rely on human operators to identify and mitigate threats, autonomous remediation leverages advanced technologies such as artificial intelligence (AI), machine learning (ML), and automation to detect and respond to security incidents in real-time. (Sontan & Samuel., 2024). Autonomous remediation strategies offer several key advantages for enhancing network security posture. For one these systems excel in detecting and responding to security threats instantly (Nankya et al., 2023). This capability significantly reduces the time it takes to identify and address potential cyber-attacks, thereby minimizing the impact and damage caused by such incidents. Real-time threat response is crucial in today's rapidly evolving threat landscape where swift action can make a substantial difference in preventing security breaches (Naseer et al., 2021).

Notably, automation of remediation processes provides organizations with the ability to scale their security operations efficiently (Fung, 2014). By automating repetitive tasks and responses to security incidents, companies can handle a larger volume of threats without overburdening their human security teams. This scalability ensures that security measures can keep pace with the growing complexity and frequency of cyber-attacks, ultimately enhancing the overall efficiency of security operations. Also, since autonomous remediation systems leverage advanced technologies such as Artificial Intelligence (AI) and Machine Learning (ML) algorithms they can continuously adapt and improve their response mechanisms (Sontan & Samuel 2024). By learning from past security incidents, these systems become more intelligent over time, enhancing their ability to detect and mitigate both known and emerging threats. The adaptability and intelligence of autonomous remediation systems make them invaluable assets in maintaining a robust and proactive security posture (Sontan & Samuel 2024).

Despite the potential benefits they offer, the development and implementation of autonomous remediation strategies in network security can pose various technical and operational challenges (Cheminod et al.,

2012). For instance, integrating autonomous remediation systems with existing security infrastructure can be complex and time-consuming, requiring compatibility with diverse technologies and protocols. Also ensuring the accuracy of threat detection and response mechanisms is crucial to avoid false positives (incorrectly identifying benign actions as threats) and false negatives (failing to detect actual threats). Not to mention, autonomous systems often require access to sensitive data for effective threat analysis, raising concerns about data privacy, regulatory compliance, and the potential for misuse (Taeihagh & Lim., 2019). Addressing these technical and operational challenges is essential to successfully harness the benefits of autonomous remediation strategies while maintaining a secure and resilient network environment (Olaniyi et al., 2023). Organizations must carefully plan and execute their implementation strategies, considering these challenges to maximize the effectiveness of autonomous security measures.

In light of these challenges and opportunities, there is a growing need for research and development efforts to advance the state-of-the-art in autonomous remediation strategies for network security. This study seeks to contribute to this endeavor by investigating novel approaches, algorithms, and technologies for autonomous threat detection and response, with the aim of enhancing the resilience and effectiveness of network security defenses in the face of evolving cyber threats.

1.2 Problem Statement

Traditional network security relies on manual intervention for threat detection, analysis, and remediation. This reactive approach is becoming increasingly insufficient. The ever-growing volume and sophistication of cyber-attacks constantly bombard security teams, making timely and effective response a significant challenge. Furthermore, the complexity of modern network infrastructures, with highly distributed and interconnected systems, creates blind spots and expands the attack surface. This complexity makes it difficult to identify and isolate threats effectively. Finally, the scarcity of skilled cyber security professionals creates another obstacle. The demand for qualified security personnel far outpaces the supply, hindering organizations' ability to effectively manage their security posture. This reactive approach leaves networks vulnerable to significant breaches and data loss. To address these limitations, we need a new paradigm: autonomous remediation strategies for network security.

This thesis proposes the development of an intelligent system that can autonomously perform several crucial tasks. The system will be able to detect and classify security threats in real-time with high accuracy. Additionally, it will make informed decisions on the most appropriate remediation strategy for each threat. Finally, the system will be able to execute remediation actions automatically, minimizing human intervention and response times. By achieving these goals, we can significantly improve network security posture, reduce the burden on security teams, and enhance an organization's overall resilience against cyber-attacks.

1.2.1 Research Questions

Developing a robust and effective autonomous remediation strategy for network security requires delving into several key areas. Here, we'll explore the research questions that will guide our investigation and formulate hypotheses to be tested throughout the research process.

Research Question 1: What combination of techniques can be most effectively employed to achieve real-time threat detection and classification with high accuracy within an autonomous remediation system?
Hypothesis (H1): A hybrid approach that utilizes signature-based detection for known threats combined with anomaly detection powered by machine learning will achieve superior accuracy in real-time threat identification compared to relying solely on either technique.

Rationale: Signature-based detection excels at identifying known threats but struggles with novel attacks. Machine learning can learn from network behavior patterns to detect anomalies indicative of potential threats. Combining these methods leverages the strengths of each, enhancing overall detection accuracy.

Research Question 2: What range of automated remediation actions can be implemented within the system to effectively address diverse security threats while minimizing collateral damage and disruption to network operations?

Hypothesis (H2): The system's remediation strategy toolbox should encompass a spectrum of actions, including:

1. **Isolation:** Isolating infected devices to prevent lateral movement and further compromise.
2. **Patching:** Automatically deploying security patches to known vulnerabilities exploited by the detected threat.
3. **Countermeasures:** Activating countermeasures specific to the identified threat type, such as blocking malicious IP addresses or filtering network traffic.
4. **Resource Throttling:** Limiting resource access for compromised systems to minimize potential damage.

Rationale: A diverse remediation toolkit allows the system to tailor its response to the severity and nature of the threat. Isolation prevents further infection, patching addresses vulnerabilities, countermeasures neutralize specific attack methods, and throttling minimizes damage potential.

Research Question 3: What decision-making algorithms can be developed to enable the system to select the most appropriate remediation strategy for a given threat scenario, considering factors like threat severity, potential impact, and available resources?

Hypothesis (H3): A multi-layered decision-making algorithm will be employed, incorporating the following elements:

1. **Threat severity classification:** The system will categorize threats based on their potential impact on the network (e.g., critical systems compromise vs. information theft).
2. **Risk assessment:** The algorithm will assess the potential damage caused by the threat if left unchecked.
3. **Resource availability:** The system will consider available resources (e.g., bandwidth, processing power) when selecting a remediation strategy to avoid overloading the network.

Rationale: A multi-layered approach ensures the system prioritizes critical threats, balances risk mitigation with resource limitations, and avoids unintended consequences from resource-intensive remediation actions.

Research Question 4: How can machine learning be effectively integrated into the system to enhance threat detection, automate remediation actions, and continuously improve its response capabilities over time?

Hypothesis (H4): Machine learning will play a crucial role in the system's functionality through several mechanisms:

1. **Supervised learning:** Training machine learning models on historical network data and labeled threat examples to improve real-time threat detection accuracy.
2. **Unsupervised learning:** Identifying anomalies in network behavior that may indicate novel or zero-day attacks.

3. **Reinforcement learning:** The system will learn and adapt its remediation strategies based on the effectiveness of past actions and feedback from security personnel.

Rationale: Machine learning provides the system with the ability to learn and adapt. Supervised learning refines threat detection, unsupervised learning identifies novel threats, and reinforcement learning optimizes remediation strategies over time.

1.3 Aim of the Study

The core aim of this study is to bridge the gap between traditional, reactive network security and the growing need for proactive and autonomous threat mitigation. This will be achieved through the development and evaluation of an intelligent system capable of autonomously remediating real-time security threats within a network environment. This translates to creating a system that can effectively:

1 Identify and classify security threats in real-time with high accuracy, minimizing the window of opportunity for attackers.
2 Automate a diverse set of remediation actions, tailoring the response to the specific threat and minimizing collateral damage.
3 Employ intelligent decision-making algorithms to select the most appropriate remediation strategy for each scenario, considering factors like threat severity, impact, and resource availability.
4 Continuously learn and improve its capabilities by leveraging machine learning techniques for threat detection, anomaly identification, and optimizing remediation strategies over time.

1.3 Research Objectives

To achieve the overarching aim of developing and evaluating an autonomous remediation system for network security, this study will pursue a series of specific objectives:

1. Develop a real-time threat detection engine combining signature-based and machine learning techniques.
2. Identify and develop a comprehensive set of automated remediation actions that the system can execute upon threat detection.
3. Develop a multi-layered decision-making algorithm that considers various factors to select the optimal remediation strategy for each threat scenario. These factors may include
4. Develop and integrate machine learning techniques to enhance the system's capabilities over time.
5. Design and implement a comprehensive testing methodology to rigorously evaluate the developed autonomous remediation system.

1.4 Scope of the Research

The system will be designed to detect and classify security threats in real-time using a combination of signature-based detection and machine learning-powered anomaly detection. This allows for a comprehensive approach to threat identification, encompassing both known threats and potential novel attacks. Upon threat detection, the system will have a predefined set of automated remediation actions at its disposal. These actions may include isolating infected devices, deploying security patches, activating countermeasures specific to the identified threat type, and throttling resources to minimize potential damage. The system will employ multi-layered decision-making algorithms that consider factors like threat severity, potential impact, and available resources to select the most appropriate remediation strategy for each scenario. This ensures a measured and targeted response that avoids unnecessary disruption to network operations.

Supervised and unsupervised learning will be used to refine threat detection, identify novel threats, and optimize remediation strategies over time. Reinforcement learning may also be explored to allow the

system to learn from past experiences and continuously improve its effectiveness. The developed system will undergo rigorous evaluation through simulated attacks. Performance metrics like detection rate and response time will be measured to assess the system's accuracy and efficiency. Additionally, the system's integration with a security operations workflow will be evaluated to ensure usability and effectiveness within a realistic security team environment.

It's important to note that the study's scope excludes certain aspects. While the system is designed for network security applications, the actual implementation within a specific physical network environment is beyond its focus. Integration with existing security tools and infrastructure on a real network would require collaboration with security professionals and engineers. The study also won't delve into highly specialized threat detection techniques like advanced persistent threat (APT) detection or zero-day exploit analysis. These areas require significant expertise and may be the focus of future studies. Social engineering attacks, which rely on human manipulation, will not be a central theme either. However, the system may be able to integrate with existing solutions that address social engineering attempts.
Mitigating large-scale Denial-of-Service (DoS) attacks is a complex challenge. While the system may be able to detect and initiate basic countermeasures against DoS attempts, a comprehensive DoS mitigation strategy would likely require additional research and development efforts beyond the scope of this study.

1.5 Significance of the Study

The traditional approach to network security relies heavily on manual intervention for threat detection, analysis, and remediation. This reactive posture leaves networks vulnerable to a constantly evolving threat landscape. The ever-increasing volume and sophistication of cyber-attacks overwhelm security teams struggling to keep pace. Modern network infrastructures, highly distributed and interconnected, create blind spots and an expanded attack surface, making it difficult to identify and isolate threats quickly. Finally, the cyber security skills shortage means organizations often lack the expertise to effectively manage their network security posture. These factors highlight the need for a new paradigm: autonomous remediation strategies for network security. This study aims to address these challenges by developing and evaluating an intelligent system capable of real-time threat detection and analysis with high accuracy. It will utilize a combination of signature-based detection and machine learning, minimizing the window of opportunity for attackers.

Upon threat detection, the system will possess a diverse set of automated responses, allowing it to tailor its actions to the specific threat encountered. This includes isolating infected devices, deploying security patches, activating countermeasures, and throttling resources as needed. By employing multi-layered algorithms, the system will consider factors like threat severity, potential impact, and available resources to select the most appropriate remediation strategy, ensuring a measured and targeted response. The successful development of this autonomous remediation system will have a significant impact on the field of network security. Security personnel can be freed from tedious tasks like threat detection and basic remediation, allowing them to focus on more strategic initiatives. Autonomous remediation allows for immediate action against threats, minimizing the potential damage and disruption caused by cyber-attacks. The system can effectively handle large volumes of security events, making it suitable for complex and expansive network infrastructures. By automating remediation actions, the system can potentially take preventative measures to stop attacks before they even occur.

This study's significance extends beyond the immediate benefits of the developed system. It will contribute valuable knowledge and insights to the field of autonomous security by demonstrating the feasibility of autonomous remediation. The findings and lessons learned will pave the way for further advancements in autonomous security solutions. By highlighting the potential of autonomous remediation, this study can foster collaboration between researchers, developers, and security professionals to accelerate the development and adoption of these technologies.

1.6 Definition of Terms

For a clear understanding of the concepts presented in this study, here are definitions of key terms:

1. **Autonomous Remediation:** The ability of a system to automatically detect, analyze, and respond to security threats within a network environment without requiring constant human intervention.
2. **Threat Detection:** The process of identifying and classifying malicious activity within a network. This can involve signature-based detection (matching known threats) and anomaly detection (identifying unusual behavior patterns).
3. **Anomaly Detection:** A technique used to identify deviations from normal network behavior patterns that may indicate potential security threats. Machine learning algorithms play a crucial role in anomaly detection.
4. **Machine Learning:** A branch of artificial intelligence that allows computers to learn and improve without explicit programming. In this context, machine learning will be used to refine threat detection, identify novel threats, and optimize remediation strategies.
5. **Remediation:** The actions taken to neutralize a security threat and minimize potential damage. This can involve isolating infected devices, deploying security patches, activating countermeasures, or throttling resources.
6. **Signature-Based Detection:** A method of threat detection that relies on pre-defined signatures (patterns) of known threats. Security software can identify malicious activity by comparing network traffic or file characteristics to these signatures.
7. **Decision-Making Algorithms:** A set of rules and logic used by the system to analyze threat information and select the most appropriate remediation strategy for each scenario. Factors like threat severity, potential impact, and available resources will be considered in the decision-making process.
8. **Supervised Learning:** A machine learning technique where the system is trained on labeled data sets. In this context, supervised learning will be used to train models for threat detection by providing examples of both malicious and benign network activity.
9. **Unsupervised Learning:** A machine learning technique where the system learns patterns from unlabeled data. In this context, unsupervised learning will be used to identify anomalies in network behavior that may indicate novel or zero-day attacks.
10. **Reinforcement Learning:** A machine learning technique where the system learns through trial and error, receiving rewards for successful actions and penalties for failures. This approach could be explored to allow the autonomous remediation system to learn and adapt its strategies based on the effectiveness of past actions and feedback from security personnel.
11. **Zero-Day Attack:** A novel cyberattack for which no security patches or signatures exist at the time of the attack.
12. **Denial-of-Service (DoS) Attack:** An attack aimed at disrupting the normal operation of a network or service by overwhelming it with a flood of traffic, making it unavailable to legitimate users.

By understanding these key terms, the reader will gain a deeper understanding of the functionalities and goals of the proposed autonomous remediation system.

1.8 Organization of the Study

The thesis is divided into five chapters. Chapter 1 introduces the challenges of network security and the need for autonomous remediation systems. It also outlines the research objectives and potential impact. Chapter 2 reviews existing research on autonomous security and network remediation. It analyzes related works and identifies key trends and opportunities. Chapter 3 details the research methodology, explaining the proposed autonomous remediation system, the development process, and how the system will be evaluated. Chapter 4 presents the findings from evaluating the system and discusses their significance for network security. Chapter 5 summarizes the key findings, offers conclusions and recommendations, and highlights the research contributions.

Chapter 2: Introduction

2.1 Preamble

This chapter builds upon the foundation established in Chapter 1, which highlighted the necessity for autonomous remediation strategies in network security due to the limitations of traditional, reactive approaches. Here, we delve into the existing body of research on autonomous remediation, examining various methodologies, architectures, and key findings to inform the design and development of our own intelligent remediation system.

Through a critical analysis of existing solutions, we aim to identify best practices, potential shortcomings, and areas for further development. This comprehensive review will equip us with the necessary insights to create a robust and effective autonomous remediation system that addresses the evolving threat landscape of today's networks.

2.2 Theoretical Framework

The escalating complexity and frequency of cyber-attacks necessitate the exploration of more agile and intelligent security solutions. This thesis investigates the development of autonomous remediation strategies for network security incidents. To achieve this, the research draws upon established theoretical concepts that provide a robust foundation for building a self-governing and intelligent security system.

2.2.1 Continuous Monitoring and Anomaly Detection

The foundation of any network security strategy is continuous monitoring (Wu et al., 2020). Intrusion Detection Systems (IDS) act as the frontline defense mechanism within network security architectures (Rege et al., 2024). They continuously analyze network traffic for activities that deviate from established baselines, potentially signifying malicious attempts. These deviations, often referred to as anomalies, can manifest in various forms, including:

1. **Unauthorized Access Attempts:** IDS can detect activities indicative of unauthorized access attempts, such as login attempts with invalid credentials or brute-force attacks targeting specific accounts (Burch 2018). These attempts often exhibit rapid bursts of authentication requests or exploit known weak password policies.
2. **Unusual Data Transfer Patterns:** Significant deviations from normal data transfer patterns can serve as red flags for potential security incidents (Zhang et al., 2003). For instance, a sudden surge in outbound data traffic originating from a typically low-volume workstation might indicate malware exfiltrating sensitive information. Conversely, a drastic decline in expected traffic from a critical server could suggest a denial-of-service attack targeting its resources.
3. **Attempts to Exploit Known Vulnerabilities:** IDS can be configured to identify attempts to exploit known vulnerabilities in network devices, operating systems, or applications. This is achieved by matching network traffic patterns against signatures of these vulnerabilities stored within the IDS knowledge base (Lippmann et al., 2002). These signatures are constantly updated based on the latest threat intelligence feeds, ensuring the system remains vigilant against evolving attack techniques.

By leveraging the data collected by IDS, the autonomous remediation system can achieve real-time identification of potential security incidents. This expeditious identification allows for a faster and more effective response, potentially mitigating the damage caused by the attack before it escalates. However, while signature-based IDS offer a valuable layer of security, they have inherent limitations. These limitations stem from their reliance on predefined attack signatures. Novel attack vectors, constantly developed by malicious actors, may not be reflected in existing signatures. This renders traditional IDS ineffective against such zero-day attacks, highlighting the need for a more comprehensive approach to network security monitoring. To address the limitations of signature-based IDS, anomaly detection

techniques are increasingly being incorporated. Anomaly detection focuses on identifying deviations from normal network behavior, regardless of whether a specific attack signature is known. This approach offers greater flexibility and adaptability in the face of evolving threats. Here are some key approaches to anomaly detection relevant to this research:

1. **Statistical Anomaly Detection:** This approach analyzes network traffic data using statistical methods to identify significant departures from established baselines. For instance, it might calculate metrics like average packet size, number of connections per host, or overall network traffic volume. Deviations exceeding pre-defined thresholds can be flagged as potential anomalies. Statistical anomaly detection offers a computationally efficient approach, making it suitable for real-time monitoring of large network infrastructures.

2. **Machine Learning-Based Anomaly Detection:** Machine learning algorithms can be trained on historical network data and continuously updated threat intelligence feeds to learn the characteristics of normal network activity. Once trained, these algorithms can continuously monitor network traffic and identify significant deviations as potential security incidents. This approach offers greater flexibility and adaptability compared to purely statistical methods, as it can learn complex patterns and relationships within the data that might not be readily apparent through traditional statistical analysis.

   Supervised learning techniques form the cornerstone for ML-based threat classification within the autonomous remediation system. These techniques are grounded in Vapnik's seminal work on Statistical Learning Theory (Vapnik, 1998), which establishes the theoretical foundation for learning algorithms that can generalize well from a finite training dataset to unseen data points. Supervised learning involves training ML models on curated historical network data and continuously updated threat intelligence feeds. The training data consists of labeled examples, where each network event is categorized as either normal network activity or a specific type of security incident. This labeling process enables the ML models to learn the distinctive features and characteristics that differentiate between normal and malicious network behavior, aligning with Mitchell's concept of version space learning (Mitchell, 1997). Common supervised learning algorithms employed for threat classification in network security settings include:

   1. **Support Vector Machines (SVMs):** These powerful algorithms excel at effectively classifying data points by identifying a hyperplane that maximizes the margin between distinct classes (normal traffic vs. malicious traffic). Their inherent ability to handle high-dimensional data makes them particularly well-suited for network security applications, as network traffic data often encompasses a multitude of features. This effectiveness is theoretically grounded in the principles of large margin classification (Cortes & Vapnik, 1995).
   2. **Decision Trees:** These algorithms classify data points by traversing a tree-like structure where each node represents a specific feature of the network traffic and each branch represents a possible value for that feature. The model meticulously traverses the tree based on the features of the incoming data point, ultimately reaching a leaf node that represents the predicted class (normal or malicious). Decision trees offer the advantage of interpretability, allowing for a clearer understanding of the decision-making process behind the classification. This interpretability aligns with the theoretical concept of explainable AI (XAI) (Samek et al., 2019), which emphasizes the importance of understanding how ML models arrive at their predictions.

   Nevertheless, deep learning models, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) can learn complex patterns and relationships within large datasets potentially enabling them to identify more sophisticated and novel attack vectors compared to traditional machine learning models. This is based on the principles of artificial

neural networks (ANNs) pioneered by McCulloch & Pitts (1943) and further developed by researchers like Rosenblatt (1958). Deep learning models build upon these foundations by employing multiple layers of interconnected artificial neurons, enabling them to learn complex, non-linear relationships within the data. Here's a breakdown of how deep learning algorithms can be utilized for enhanced threat detection:

i. **Convolutional Neural Networks (CNNs):** CNNs excel at identifying patterns within spatial data, making them suitable for analyzing network traffic data represented as sequences of bytes or packets. By processing this data through multiple convolutional layers, CNNs can learn to extract features that are indicative of specific attack types. This capability aligns with the concept of convolutional filters, which are used by CNNs to identify local patterns within the data (LeCun et al., 1998).

ii. **Recurrent Neural Networks (RNNs):** RNNs are adept at handling sequential data, making them well-suited for analyzing network traffic flows that unfold over time. RNNs can capture temporal dependencies within the data, allowing them to identify complex attack patterns that may involve multiple steps or interactions between different network elements. This capability is based on the concept of back propagation through time (Rumelhart et al., 1986), which allows RNNs to learn from past network traffic data when making predictions about current traffic.

The integration of deep learning models within the autonomous remediation system has the potential to significantly enhance threat detection capabilities, particularly in the face of novel attacks that traditional methods might struggle to identify. By analyzing the classified threat and its associated features, the autonomous system can assess the severity of the incident based on the principles of risk management (Kaplan & Garrick, 1981). Risk management posits that risk is a function of the likelihood of an event occurring and the potential consequences of that event. In the context of network security, the autonomous system leverages threat intelligence feeds and historical data to estimate the likelihood of a particular attack succeeding and prioritizes responses to high-risk attacks, which aligns with the Common Vulnerability Scoring System (CVSS) (Mell et al., 2007), a standardized framework for assessing the severity of vulnerabilities.

The autonomous remediation system can also assess the potential consequences of the attack based on the targeted assets and the attacker's objectives. This assessment considers factors like data confidentiality, integrity, and availability (CIA triad) (Parker, 1998). By understanding the potential impact, the system can prioritize responses to threats that pose a significant risk to critical network resources. Analyzing historical data on similar attacks can also provide valuable insights into attacker behavior and potential remediation strategies. By incorporating this knowledge, the autonomous system can select more effective responses. This approach aligns with the principles of incident response (IR), a well-established cyber security practice that emphasizes learning from past incidents to improve future response capabilities. This enables the system to choose the most appropriate remediation strategy for each unique incident, optimizing the response effectiveness while minimizing resource consumption.

2.2.2 Maintaining Network Stability During Remediation

An effective autonomous remediation system must possess the ability to strike a delicate balance between swift and decisive action in mitigating security incidents and minimizing disruption to legitimate network operations. Network stability encompasses the ability of a network infrastructure to deliver consistent and reliable performance while maintaining acceptable levels of latency, packet loss, and resource utilization (Zheng et al., 2019). During security incidents, the implementation of autonomous remediation strategies can inadvertently disrupt network stability. For instance, quarantining an infected system may be an effective way to contain an attack, but it could also disrupt legitimate network traffic if the

quarantined system is a critical resource. This highlights the inherent multi-objective nature of the autonomous remediation challenge: achieving optimal security posture while minimizing disruption to network performance.

Control theory is a valuable option for maintaining network stability during autonomous remediation as it focuses on the design of systems that regulate themselves based on feedback mechanisms (Lamnabhi-Lagarrigue et al., 2017). By applying these principles, the autonomous remediation system can dynamically adjust its strategies based on real-time feedback regarding network performance metrics. The integration of control theory principles within the autonomous remediation system can be implemented in various ways:

1. **Adaptive Quarantine:** Instead of completely quarantining a potentially compromised system, the system could implement a more granular approach based on the principles of model predictive control (MPC) (Sai, 2018). This might involve isolating specific network segments or restricting the compromised system's access to critical resources, thereby mitigating the attack while minimizing disruption to legitimate traffic.
2. **Resource Throttling:** If a remediation strategy is consuming excessive network resources, such as during a deep system scan, the control mechanisms can dynamically throttle the resource allocation based on the principles of optimal control (Zomaya & Lee, 2012). This ensures that the remediation process itself does not become a bottleneck for network performance.
3. **Prioritized Remediation:** The autonomous system could prioritize remediation actions based on the potential impact on network stability and security posture. This aligns with the concept of multi-objective decision-making in control theory (Lamnabhi-Lagarrigue et al., 2017). For instance, addressing vulnerabilities on critical servers might take precedence over remediating vulnerabilities on less critical systems, while still ensuring overall network security is maintained.

Hence, the autonomous remediation system can achieve a delicate balance between security and network stability by leveraging control theory principles. This ensures that security incidents are effectively mitigated while maintaining the smooth operation of critical network services.

2.3 Review of Relevant Literature

There is a body of knowledge surrounding network security fundamentals, with a specific focus on technologies and methodologies relevant to the development of an autonomous network remediation system. This knowledge includes understanding the principles of network security, the various security threats and vulnerabilities, and the technologies and tools used to mitigate these risks. Some key topics include firewalls, intrusion detection systems, secure routing protocols, and software-defined networking.

2.3.1 Network Security Fundamentals

Network security fundamentals involve understanding the critical elements of network security, which should be implemented within all networks including home, business, and internet networks. Effective network security requires protection of wired and wireless networks with firewalls, anti-malware software, intrusion detection systems, access control, and encryption. Network security is a complex topic that involves many different technologies with configurations that are sometimes difficult to manage.

2.3.1.1 Firewalls and Network Segmentation

Firewalls act as the first line of defense within network security architectures. They thoroughly examine incoming and outgoing network traffic based on a predefined set of security rules. Packets that adhere to the established rules are permitted passage, while those deemed suspicious or malicious are blocked (Song et al., 2020). Firewalls can be categorized based on their deployment strategy:

1. **Stateful Firewalls:** These firewalls maintain state information about ongoing network connections, allowing them to make more informed decisions about permitting or blocking traffic. They can differentiate between legitimate established connections and potential intrusion attempts.
2. **Stateless Firewalls:** These firewalls solely analyze individual packets without considering any context or connection state. They offer a simpler and faster approach to traffic filtering but may be less effective in identifying sophisticated attacks.

Network segmentation complements firewalls by dividing the network into smaller, logically isolated segments. This approach restricts the lateral movement of attackers within the network, potentially limiting the impact of a security breach (Treider, 2023). Firewalls can be strategically deployed at the boundaries of these segments, further enhancing overall network security. Firewalls and network segmentation contribute to autonomous network remediation in the following ways:

1. **Threat Containment:** Firewalls can be dynamically configured by the autonomous system to block suspicious traffic associated with identified threats. This can help contain the spread of an attack and minimize its impact.
2. **Improved Threat Detection:** Network segmentation can simplify traffic analysis for intrusion detection systems (IDS) deployed within each segment. This can potentially improve the accuracy and efficiency of threat detection.

2.3.1.2 Intrusion Detection/Prevention Systems (IDS/IPS)

Intrusion Detection Systems (IDS) continuously monitor network traffic for activities that deviate from established baselines. These deviations, often referred to as anomalies, might signal potential security incidents such as unauthorized access attempts, denial-of-service attacks, or malware propagation (Muhati & Rawat 2024). IDS can be categorized based on their detection methods:

1. **Signature-based IDS:** These systems rely on predefined attack signatures to identify malicious activity. They offer high accuracy in detecting known threats but may struggle to identify novel or zero-day attacks.
2. **Anomaly-based IDS:** These systems analyze network traffic patterns and identify deviations from established baselines. This approach can be effective in detecting novel attacks, but it may also generate false positives due to legitimate traffic fluctuations.

Intrusion Prevention Systems (IPS) take a more proactive approach by not only detecting but also actively blocking or mitigating identified threats. They leverage the same detection techniques as IDS but are integrated with the network infrastructure to enforce security policies (Thapa and Mailewa, 2020). IDS/IPS provide insights into potential security incidents by continuously monitoring network traffic and identifying anomalies. This information is crucial for the autonomous system to trigger appropriate remediation actions. In some cases, IPS can be configured to automatically take pre-defined actions upon detecting a threat, such as blocking malicious traffic or quarantining compromised systems. This can expedite the initial response to security incidents. However, it's important to acknowledge the limitations of IDS/IPS. They rely heavily on accurate and up-to-date threat intelligence feeds to function effectively. Additionally, fine-tuning these systems to minimize false positives requires ongoing effort and expertise.

2.3.2 Autonomous Systems in Security

Autonomous systems in security are becoming increasingly important due to the ever-evolving threat landscape. These systems are designed to operate independently of human intervention, capable of identifying, analyzing, and responding to threats in real-time. Machine learning (ML) and artificial intelligence (AI) are emerging as potent forces in this domain, particularly in the application of autonomous network remediation. ML is the most common approach in cyber security and shows great promise, although it does have some drawbacks. It can provide a great deal of support to a cyber-security

or IT team, handling tasks such as classification, clustering, recommendations, generative frameworks, predictions, and more.

2.3.2.1 Machine Learning for Security Automation

Machine Learning (ML) algorithms help in automating various security tasks within autonomous remediation systems including:

1. **Threat Classification:** ML algorithms can be trained on historical network traffic data and threat intelligence feeds to effectively classify incoming traffic as normal, malicious, or anomalous. Supervised learning techniques excel at identifying patterns within labeled data, enabling them to distinguish between legitimate and malicious activity (Bouchama & Kamal, 2021)..
2. **Anomaly Detection:** Unsupervised learning techniques like clustering and outlier detection algorithms are adept at identifying deviations from established baselines within network traffic patterns. This allows the autonomous system to detect novel attacks that might evade signature-based detection methods (Usmani et al., 2022).
3. **Threat Severity Assessment:** By analyzing the characteristics of a classified threat, ML models can estimate its potential impact on the network. This assessment might consider factors like the targeted assets, the attacker's objectives, and the exploitability of the vulnerability. Techniques like probabilistic modeling and decision trees can be employed for this purpose (Manoharan & Sarker, 2023).
4. **Response Selection:** Based on the threat classification, severity assessment, and the capabilities of the system, ML algorithms can recommend or even automate the selection of the most appropriate remediation strategy. Reinforcement learning techniques can be particularly valuable in this context, as they allow the system to learn from its past experiences and refine its response selection over time (Zhang et al., 2020).

The integration of ML into autonomous network remediation systems offers several advantages:

i. **Enhanced Detection Rates:** ML algorithms can analyze vast amounts of data in real-time, enabling them to identify subtle anomalies and novel attack patterns that might evade traditional signature-based detection methods.
ii. **Improved Efficiency:** By automating threat classification, severity assessment, and response selection, ML streamlines the security response process, allowing the system to react swiftly and decisively to security incidents.
iii. **Reduced Reliance on Manual Expertise:** ML reduces the dependence on highly skilled security personnel to continuously monitor and analyze network traffic. This frees up their time to focus on more strategic tasks.

However, it's important to acknowledge the challenges associated with ML-based security solutions:

i. **Data Quality:** The effectiveness of ML models hinges on the quality and quantity of training data. Biased or incomplete data can lead to inaccurate threat classification and suboptimal response selection.
ii. **Adversarial Attacks:** Malicious actors might attempt to manipulate network traffic patterns to evade ML-based detection systems. Continuous monitoring and adaptation of the ML models are essential to counter such attacks.
iii. **Explainability and Transparency:** Understanding the rationale behind an ML model's decision-making process is crucial for building trust and ensuring responsible use within security applications. Explainable AI (XAI) techniques are being actively researched to address this challenge.

2.3.2.2 Artificial Intelligence in Cybersecurity

Artificial intelligence (AI) encompasses a broader range of techniques than Machine Learning. While ML focuses on algorithms that learn from data, AI encompasses various approaches to simulating human intelligence, including knowledge representation, reasoning, and problem-solving. Here are some potential applications of AI within autonomous network remediation:

1. **Automated Incident Investigation:** AI systems can analyze forensic data, network logs, and threat intelligence to automate the investigation process of security incidents. This can significantly reduce the time and resources required to identify the root cause of an attack.
2. **Threat Prediction and Proactive Defense:** AI can be used to analyze historical attack data and identify emerging trends or patterns. This enables the autonomous system to predict potential threats and take proactive measures to mitigate them before they materialize.
3. **Continuous Learning and Adaptation:** AI systems can continuously learn and adapt their security strategies based on new data and experiences. This is particularly valuable in the face of constantly evolving cyber threats.

While AI holds immense potential for transforming cyber-security, significant challenges remain including:

i. **Complexity and Interpretability:** Developing and maintaining robust AI systems can be complex and resource-intensive. Additionally, ensuring the transparency and explainability of AI-driven decisions is crucial for building trust in security applications.
ii. **Data Requirements:** As with ML, effective AI systems require access to vast amounts of high-quality data to train and refine their models.
iii. **Ethical Considerations:** The potential misuse of AI for malicious purposes necessitates careful consideration of ethical implications during development and deployment.

2.3.2.3 Other Techniques for Autonomous Security

Beyond Machine Learning and Artificial Intelligence, other techniques contribute to the development of robust autonomous security systems:

1. **Formal Verification:** This demanding mathematical approach involves formally specifying the desired security properties of a system and then mathematically proving that the system adheres to those properties (Toman et al., 2024). While computationally expensive, formal verification can offer high levels of assurance regarding the security posture of the autonomous system itself.
2. **Decentralized Security Architectures:** These architectures distribute security functionalities across various network devices and resources. This approach can enhance scalability, resilience, and survivability in the face of cyber-attacks, as there's no single point of failure (Khan & Nencioni, 2023). Blockchain technology, with its tamper-proof distributed ledger, can be a foundation for implementing decentralized security mechanisms.
3. **Behavioral Analysis and Deception Techniques:** These techniques involve monitoring user and system behavior patterns to identify anomalies that might indicate malicious activity. Additionally, deception techniques can be employed to lure attackers into isolated environments, where their actions can be monitored and analyzed without compromising critical network resources (Steingartner et al., 2021).

These complementary techniques, when combined with Machine Learning and AI, can empower autonomous security systems to effectively defend against a wider range of cyber threats.

2.3.3 Network Remediation Strategies

Network remediation strategies are crucial for addressing and resolving security incidents in a timely and efficient manner. These strategies aim to minimize the impact of security breaches and prevent future incidents.

2.3.3.1 Manual Remediation Techniques

While autonomous systems are rapidly transforming the security landscape, manual remediation techniques remain a cornerstone of established incident response practices. These techniques often involve a well-defined sequence of steps undertaken by security personnel to address security incidents:

1. **Containment:** The primary objective of containment is to isolate the compromised system or network segment. This prevents the lateral movement of attackers and minimizes the potential impact of the attack. Techniques used for containment include quarantining infected systems, blocking malicious traffic at firewalls, or disabling vulnerable services (George et al., 2023).
2. **Eradication:** Once the threat is contained, eradication focuses on eliminating the malicious code or presence from the compromised system. This could involve actions like system reboots, full system wipes, or the removal of malware using specialized security software (Mell et al., 2005).
3. **Recovery:** After eradication, the focus shifts towards restoring affected systems and data to a known good state. This might involve restoring backups, rebuilding critical servers, or reconfiguring compromised systems (Johnphill et al., 2023).
4. **Lessons Learned:** Following the successful remediation of an incident, it's crucial to conduct a thorough post-mortem analysis to identify vulnerabilities exploited by the attack. This knowledge can be used to update security policies, patch vulnerabilities, and improve future incident response strategies (Thompson, 2018).

However, manual remediation techniques have inherent limitations that hinder their effectiveness in a dynamic threat landscape:

i. **Time-consuming Nature:** Manual incident response can be a lengthy process, leaving the network vulnerable for extended periods. This is particularly problematic in fast-moving cyber-attacks where rapid response is critical (Sontan & Samuel, 2010).
ii. **Susceptibility to Human Error:** The effectiveness of manual remediation relies heavily on the expertise and vigilance of security personnel. Mistakes during the process can exacerbate the situation or leave residual vulnerabilities unaddressed (Meyers, 2023).
iii. **Scalability Challenges:** As network size and complexity increase, manual remediation becomes increasingly difficult and resource-intensive. This becomes a significant hurdle for large organizations managing vast network infrastructures (Vasoya, 2023).

These limitations highlight the need for automation within the network remediation process.

2.3.3.2 Automated Remediation Techniques

The limitations of manual approaches necessitate the exploration of automated remediation techniques. These techniques leverage scripting, policy-based automation, and integration with security tools to expedite and streamline the incident response process.

2.3.3.2.1 Script-based Automation

Scripting languages like Python, PowerShell, or Bash can be employed to automate repetitive tasks associated with network remediation. These scripts can be pre-defined for specific scenarios or dynamically generated based on the nature of the detected threat. Here are some examples of script-based automation in network remediation:

1. **Patch Deployment:** Scripts can automate the deployment of security patches across multiple systems, addressing identified vulnerabilities and mitigating potential exploits (Staunton, 2020).
2. **System Isolation:** Scripts can automate the process of quarantining infected systems or blocking malicious traffic at firewalls, effectively containing the threat (Thapa, 2018).
3. **Malware Removal:** Scripts can be designed to invoke security software and automate the detection and removal of malware from compromised systems (Ye et al., 2017).

The benefits of script-based automation include:

1. **Improved Speed and Efficiency:** Scripts can execute tasks significantly faster than manual processes, enabling a more rapid response to security incidents (Josyula et al., 2011).
2. **Reduced Human Error:** Automation minimizes the risk of errors introduced by manual intervention, leading to more consistent and reliable remediation outcomes (Maiello, 2023).
3. **Scalability:** Scripts can be easily scaled to address threats across large and complex network infrastructures (Kara, 2023).

However, script-based automation also has limitations that need to be considered:

1. **Development & Maintenance Effort:** Developing and maintaining effective scripts requires programming expertise and ongoing effort to keep pace with evolving security threats (Xu & Russello, 2022).
2. **Limited Scope:** Scripts are typically designed for specific tasks and may not be adaptable to handle unforeseen situations or novel attack vectors (McIntosh et al., 2023).

2.3.3.2.2 Policy-driven Automation

Policy-driven automation takes a more holistic approach to incident response, offering a powerful foundation for autonomous remediation systems. Security policies define the desired response actions for various security incidents based on factors like threat severity, targeted assets, and attacker behavior. These policies are then translated into actionable rules within security automation platforms. Here's how policy-driven automation functions within the context of an autonomous remediation system:

1. **Security Information and Event Management (SIEM) Integration:** SIEM systems collect and analyze security data from various network devices and security tools. They can identify potential security incidents based on pre-defined rules or anomaly detection algorithms (Zeinali, 2016).
2. **Policy Enforcement:** When a security incident is detected, the SIEM system triggers pre-defined response actions based on the associated security policy. These actions could involve automated patching, system isolation, malware removal, or notifications to security personnel for further investigation (Alahmadi, 2019).

The advantages of policy-driven automation make it particularly well-suited for autonomous remediation systems:

1. **Flexibility and Adaptability:** Security policies can be easily updated to reflect evolving threats and organizational security requirements. This allows the automated response to adapt to new situations without the need for constant script revisions (Djenna et al., 2021).
2. **Centralized Management:** Security policies can be centrally managed and enforced across the entire network infrastructure, ensuring consistency in the response to security incidents and reducing the risk of human error (Tabrizchi & Kuchaki, 2020.).
3. **Reduced Reliance on Scripting Expertise:** Policy-driven automation requires less scripting expertise compared to script-based automation. Security personnel can define policies using a graphical interface or a high-level language, lowering the barrier to entry for automation and facilitating broader adoption within an organization (Alt & Puschmann, 2012.).

However, policy-driven automation also has limitations that require careful consideration during the design and implementation of an autonomous remediation system:

1. **Complexity of Policy Definition:** Defining comprehensive and effective security policies can be a complex task. It requires a deep understanding of security threats, potential attack vectors, and the desired response actions. Incomplete or poorly defined policies can lead to suboptimal or even detrimental outcomes (Bauer & Van, 2009).
2. **Potential for Over-Automation:** Overly aggressive automation policies could lead to unintended consequences, such as accidentally disrupting legitimate network traffic or quarantining critical systems during maintenance activities. Careful testing and validation of security policies are crucial to mitigate such risks (Wu et al., 2007).
3. **Integration Challenges:** Integrating policy-driven automation with existing security tools and infrastructure can be challenging, requiring compatibility between various platforms and technologies. Careful planning and consideration of interoperability standards are essential for successful implementation (Lang & Schreiner, 2011).

2.4 Review of Related Works

The integration of artificial intelligence and machine learning into network security has been a growing area of research in recent years. AI and ML have been increasingly applied in cyber-security to improve threat detection and response. AI-based systems can analyze large amounts of network traffic and system logs to identify patterns and anomalies that may indicate malicious activity.

2.4.1 Recent Research on Autonomous Network Remediation

Recent research on autonomous network remediation has shown that AI-powered vulnerability assessment, prioritization, and remediation can significantly reduce enterprise risk. According to Gartner, monitoring systems perform three processes: Observe, Engage, and Act. Technologies that deliver these capabilities end-to-end are referred to as zero touch, which includes autonomous remediation. While autonomous remediation is still in its infancy, there is a growing expectation that it will go beyond anomaly detection and correlation analysis to include autonomous learning and decision-making.

Enterprises are already using automated remediation for specific use cases, such as low-impact, repeatable tasks, and are looking to extend its use in IT operations. SentinelOne's Singularity Vulnerability Mapping is an example of such a solution, which leverages AI to provide security teams with autonomous scanning capabilities to gain visibility across the enterprise network and remediate threats in a single click. This technology is particularly useful in the face of the ever-evolving attack landscape, as it enables security teams to do more at machine speed, continuously identifying vulnerabilities and remediating threats. However, the road to fully autonomous remediation is still long, with experts predicting that it may take 6 to 10 years before we see autonomous, self-healing IT infrastructure.

Recent research indicates that autonomous network remediation is becoming increasingly practical and is a key concept on the road to digital transformation for telecoms and other complex systems, and while we are not yet at the point of fully autonomous remediation, there is a growing expectation that it will go beyond anomaly detection and correlation analysis to include autonomous learning and decision-making.

2.4.1.1 Self-healing Network Architectures

Self-healing network architectures leverage distributed intelligence and autonomic principles to automatically detect, diagnose, and recover from security incidents with minimal human intervention (Johnpill et al., 2023). Some key characteristics of self-healing network architectures include:

1. **Distributed Decision-making:** Network devices are equipped with the ability to collect local security data, analyze it based on predefined rules, and initiate basic remediation actions without relying on a centralized controller. This reduces the reliance on a single point of failure and improves overall network resilience
2. **Adaptive Learning:** Self-healing networks can continuously learn from past encounters with security threats. This allows them to adapt their behavior over time, refine their detection capabilities, and improve the effectiveness of automated remediation strategies
3. **Self-reconfiguration:** In the event of a security incident, self-healing networks can automatically reconfigure network resources to isolate compromised systems, reroute traffic, and maintain network functionality.

Abdulrazak et al. (2022) presented a comprehensive approach to addressing the issues in IoT infrastructure through self-healing mechanisms. The authors emphasize the importance of self-healing in maintaining network integrity and reliability, especially in the face of evolving cyber threats. The study highlights the need for a comprehensive approach that addresses security, software, hardware, and networking concerns autonomously. The authors propose a self-healing IoT platform that targets the self-healing concern and includes all the components necessary for a reliable system. The paper provides a detailed overview of the existing issues in IoT architecture and existing self-healing solutions, highlighting the need for a more comprehensive approach.

John et al. (2023) explored the concept of self-healing in cyber–physical systems using machine learning. The authors discuss the potential of machine-learning algorithms in enhancing the self-healing capabilities of cyber–physical systems, particularly in deploying the 5G networks. The study highlights the importance of self-healing in maintaining system security and the potential of machine-learning algorithms in revolutionizing systems security. The authors also discuss the challenges and limitations of implementing self-healing functionality in cyber–physical systems.

Research efforts in self-healing network architectures also explore various techniques, including:

1. **Bio-Inspired Approaches:** These approaches draw inspiration from biological immune systems, where network devices collaborate to detect and respond to security threats in a distributed manner (Choraś et al., 2016). The "Bio-Inspired Intrusion Prevention and Self-healing Architecture for Network Security" paper by Muna & Azween, (2008) presented a novel intrusion detection model based on artificial immune and mobile agent paradigms for network intrusion detection. The model was inspired by the human immune system's ability to detect and respond to pathogens, aiming to enhance network security and resilience. The paper discussed the construction of the model, which is based on registries' signature analysis using Syslog-ng and Logcheck unix tools. Mobile agents, representing leukocytes of an artificial immune system, perform tasks like monitoring, distributing intrusion detection workload, and ensuring data persistence and reactivity. This real-time intrusion detection model adopts the anomaly detection paradigm and demonstrates effective performance in detecting network intrusions. Subsequently in the following year, ABdullAH & Bhakti, (2009) proposed a multi-tiered bio-inspired self-healing architectural paradigm for software systems. The architecture is designed to mimic the wound-healing process in biological systems, consisting of five phases: monitoring, fault control, repair, repair validation, and integration. Since then several researchers have investigated bio-inspired approaches to self-healing networks.

2. **Software-Defined Networking (SDN):** SDN allows for centralized control and dynamic reconfiguration of network traffic flows. This programmability can be harnessed to automate security responses within self-healing networks (Arzo et al., 2021). Sánchez et al., (2014) proposed a generic self-healing approach utilizing Bayesian Networks for fault diagnosis in centralized SDN. The study aimed to enhance fault management capabilities in SDN by

integrating autonomic principles like self-healing mechanisms. This approach was designed to improve the resilience and reliability of SDNs by enabling them to automatically detect and repair faults. Ochoa-Aday et al., (2020) explored the integration of self-healing capabilities into Software-Defined Networks (SDNs) to address the challenges of maintaining network integrity post-failure. The authors propose a paradigm shift from traditional fault recovery strategies, advocating for the use of local switch actions complemented by global controller knowledge to expedite the restoration of control paths. The results indicate that this method effectively recovers the control topology, reducing the time and message load across various network scenarios, thereby addressing scalability issues inherent in conventional fault recovery strategies.

Relevant Literature reveals that Self-healing network architectures offer promising advancements in autonomous network remediation by enabling proactive defense mechanisms and fostering network resilience. However, challenges remain in areas like distributed security policy enforcement and ensuring the security of the self-healing architecture itself.

2.4.1.2 Anomaly-based Remediation Techniques

Anomaly-based detection techniques identify deviations from established network traffic patterns that might indicate potential security incidents. These techniques play a crucial role in autonomous network remediation systems, triggering automated responses when anomalies are detected.
An overview of two prominent approaches are as follows:

i. **Machine Learning-based Anomaly Detection:** Machine learning algorithms can be trained on historical network traffic data to identify patterns that deviate from normal behavior. This allows for the detection of novel attacks that evade signature-based detection methods (Bouchama & Kamal, 2021). Anomaly scoring and outlier detection are common techniques employed for this purpose.
ii. **Statistical Anomaly Detection**: Statistical methods analyze network traffic characteristics like packet size, flow rate, and protocol distribution. Deviations from statistically expected values can signal potential anomalies (Barford et al., 2002).

Once an anomaly is detected, autonomous remediation systems can leverage various techniques to mitigate the potential threat:

1. **Adaptive Throttling:** Network traffic from the source of the anomaly can be dynamically throttled to limit potential damage and prevent resource exhaustion attacks (Mat Isa, 2022).
2. **Automatic Sandboxing:** Suspicious traffic or files can be automatically directed to a sandboxed environment for further analysis and potential isolation (Vasilescu et al., 2014).
3. **Behavioral Analysis and Deception:** Deception techniques can be used to lure attackers into isolated environments where their actions can be monitored and analyzed, aiding in automated containment and response strategies (Shhadih, 2023).

Tarek & Panos (2023) presented a novel approach to cyber-security by introducing HuntGPT, a specialized intrusion detection dashboard that employs a Random Forest classifier trained on the KDD99 dataset. The system integrates Explainable AI (XAI) frameworks such as SHAP and Lime to make model interactions more user-friendly and intuitive. Additionally, it utilizes GPT-3.5 Turbo to present detected threats in an easily understandable format. The study explores the architecture of the system, its components, and evaluates its technical accuracy using Certified Information Security Manager (CISM) Practice Exams. It assesses the quality of response readability across six metrics. The results indicate that conversational agents, supported by LLM technology and integrated with XAI, can provide robust, explainable, and actionable AI solutions in intrusion detection. This enhances user understanding and interactive experience, demonstrating the potential of LLMs in improving cyber-security measures.

2.4.1.3 Automated Incident Response Systems (AIRS)

Automated Incident Response Systems (AIRS) represent a comprehensive approach to automating the incident response lifecycle. They encompass various functionalities to streamline security operations including:

1. **Security Event Correlation:** AIRS collect security data from diverse sources across the network, correlate events, and identify potential incidents based on predefined rules or anomaly detection techniques (Stroeh et al., 2013).
2. **Automated Response Playbooks:** AIRS can execute predefined response playbooks for various security incidents. These playbooks outline a sequence of automated actions, such as network isolation, system patching, or malware removal, depending on the nature of the threat (Gracis, 2022).
3. **Integration with Security Tools:** AIRS can integrate with various security tools like firewalls, intrusion detection systems (IDS), and endpoint security solutions. This allows for centralized management and automated execution of security controls within the incident response process (González-Granadillo, et al., 2021).
4. **Reporting and Learning:** AIRS can generate reports on security incidents, providing valuable insights for security analysts and fostering continuous improvement of the automated response capabilities (Naseer et al., 2024).

Anastopoulos & Giovannelli (2022) presented a comprehensive analysis in their report titled "Automated/Autonomous Incident Response," published by the NATO Cooperative Cyber Defense Centre of Excellence. This report delves into the various aspects of cyber defense, including strategic, legal, operational, and technical dimensions of automated and autonomous incident response systems. The authors advocate for the incorporation of AI to enhance cyber defense mechanisms, providing a timely and pertinent perspective for both NATO and national security governance. Similarly, Farzaan, M. A. M., et al. (2024) conducted a study on the effectiveness of AI in detecting and responding to cyber incidents in cloud environments. Their research, "AI-Enabled System for Efficient and Effective Cyber Incident Detection and Response in Cloud Environments," demonstrates the implementation of a Random Forest model on platforms like Google Cloud and Microsoft Azure. The system's notable accuracy in threat identification and malware analysis highlights the transformative potential of AI in bolstering cyber-security within cloud-based infrastructures.

The benefits of AIRS for autonomous network remediation are numerous, some of which include:

i. **Reduced Response Times**: AIRS can automate time-consuming tasks associated with incident response, leading to faster containment and mitigation of security threats (Islam et al., 2019).
ii. **Improved Consistency**: Automated playbooks ensure consistent execution of response actions, reducing the risk of human error and improving the overall effectiveness of the incident response process (Schlette et al., 2021).
iii. **Reduced Reliance on Security Expertise:** AIRS can alleviate the burden on security personnel by handling routine incident response tasks, allowing them to focus on more complex investigations and strategic security planning (Karlzen & Sommestad, 2023).

However, challenges persist in the widespread adoption of AIRS some of which includes:

1. Complexity of Playbook Development: Defining comprehensive and adaptable response playbooks requires deep security expertise and consideration of a wide range of potential attack scenarios.
2. Risk of Over-Automation: Overly aggressive automation within playbooks could lead to unintended consequences, such as disrupting legitimate network traffic or disabling critical systems

3. Integration Challenges: Integrating AIRS with various security tools and infrastructure can be complex, requiring careful planning and consideration of interoperability standards.

Despite these challenges, AIRS represents a significant advancement towards autonomous network remediation. Their ability to automate incident response tasks, streamline security operations, and reduce reliance on manual intervention makes them a valuable tool for organizations striving to enhance their security posture in a dynamic threat landscape.

2.4.2 Comparative Analysis

A comparative analysis of the reviewed approaches to autonomous network remediation, is shown in Table 2.1, highlighting their strengths, limitations, and potential synergies:

Table 2.1: Comparative Analysis

Approach	Strengths	Limitations	Synergies
Self-healing Network Architectures	Distributed decision-making, Adaptive learning, Self-reconfiguration	Distributed policy enforcement, Security of the architecture itself	Can be combined with anomaly-based techniques for decentralized threat detection and response.
Anomaly-based Remediation Techniques	Proactive detection of novel attacks	False positives, False negatives	Can be integrated with AIRS for automated response based on detected anomalies.
Automated Incident Response Systems (AIRS)	Reduced response times, Improved consistency, Reduced reliance on expertise	Complexity of playbook development, Risk of over-automation, Integration challenges	Leverages anomaly detection techniques and integrates with self-healing architectures for comprehensive autonomous remediation.

2.5 Summary/Meta-Analysis of Reviewed of Related Works

A thorough review of relevant literature provides valuable insights into autonomous network remediation systems, with a particular emphasis on self-healing network architectures, anomaly-based remediation techniques, and Automated Incident Response Systems. This comprehensive analysis synthesizes the most significant findings, pinpoints areas of potential synergy, and delineates promising avenues for future research endeavors. Some core strengths of reviewed approaches include:

1. **Enhanced Network Resilience:** SHNAs and anomaly-based remediation techniques offer a proactive approach to network security by enabling automatic detection and response to security incidents. This reduces reliance on manual intervention and expedites response times, leading to improved overall network resilience (Pandey et al., 2024).
2. **Adaptive Learning and Threat Detection:** The incorporation of machine learning and adaptive learning techniques in SHNAs and anomaly-based systems allows the network to continuously learn from past encounters with security threats. This enables the identification of novel attack patterns and the refinement of detection capabilities over time (Shah, 2021).

3. **Automated Remediation and Resource Management:** SHNAs possess the ability to automatically reconfigure network resources in response to security incidents. This can involve isolating compromised systems, rerouting traffic flows, or dynamically provisioning additional resources. AIRS can automate various aspects of incident response, streamlining the process and minimizing human error (Müller, 2023).

By combining the strengths of SHNAs, anomaly-based remediation techniques, and AIRS, a comprehensive autonomous network remediation framework can be developed. SHNAs can provide distributed detection and self-healing capabilities, while anomaly-based techniques can offer real-time threat identification. AIRS can automate incident response actions based on information from both sources. Real-time integration with threat intelligence feeds can significantly enhance the effectiveness of autonomous remediation systems. This allows for incorporating knowledge about emerging threats and vulnerabilities into anomaly detection algorithms and automated response strategies (Carvalho et al., 2016).

Machine learning techniques can be further explored for network traffic analysis to enable predictive maintenance within SHNAs. This would allow for proactive mitigation strategies to be implemented before security incidents occur. Also, utilizing explainable AI techniques in anomaly detection can improve transparency and trust in automated decision-making within autonomous network remediation systems. This is crucial for wider adoption and user confidence (Liu et al., 2021). Developing standard protocols and interfaces for communication between different components of autonomous network remediation systems is critical. This will facilitate interoperability between diverse security tools and SHNA implementations (Eren, 2018). Some key research gaps and open challenges encountered include:

1. **Distributed Security Policy Enforcement:** Enforcing security policies consistently across a distributed network of devices within SHNAs remains a challenge. Research efforts are needed to develop secure and efficient mechanisms for policy distribution and enforcement
2. **Security of the Self-healing Architecture:** The security of the self-healing architecture itself needs to be addressed to prevent attackers from exploiting vulnerabilities and manipulating its behavior. Research on self-protection and self-healing mechanisms specifically for SHNAs is crucial
3. **Scalability and Performance Optimization:** As network size and complexity increase, scalability and performance optimization of autonomous network remediation systems become critical considerations. Research is needed to ensure efficient operation under high traffic loads and large network deployments.

Chapter 3: Research Methodology

This section presents the research methods used to propose an innovative Autonomous Network Remediation System (ANRS), aimed at overcoming the shortcomings of current systems. It elaborates on the design of the system's architecture, the algorithms implemented, and the strategy for assessing the effectiveness of the ANRS.

3.1 Preamble

Current autonomous network remediation approaches have limitations that need to be effectively tackled. Timely and efficient transmission of information over the network is crucial but should be protected against attacks and not overwhelm the underlying communication channels. Performing lightweight operations on end terminals is challenging as they are heterogeneous and usually resource-constrained. Efficiently sharing raw monitoring and inspection information among multiple detection algorithms is difficult, as they need to analyze and correlate large amounts of data from different sources, while managing identity, access control, and protecting sensitive data. Also, data quality is a key challenge, as AI network monitoring is only as effective as the data it sources. Not to mention inaccuracies can arise from missing real-time data or issues with data integrity. Integration of AI network monitoring tools with existing management and security services can be tricky and lead to subpar output if not done properly.

3.2 Problem formulation

The core problem addressed by this research is the proposal of a robust and scalable ANRS that offers the following functionalities:

1. **Automated Anomaly Detection:** Utilizing machine learning algorithms to identify deviations from normal network traffic patterns that might indicate potential security incidents.
2. **Threat Classification:** Categorizing the type of threat detected based on the identified anomaly characteristics.
3. **Automated Remediation:** Initiating pre-defined countermeasures in response to security threats, such as isolating compromised devices, throttling suspicious traffic, or dynamically reconfiguring network resources.
4. **Distributed Policy Enforcement:** Ensuring consistent application of high-level security policies across all network devices.
5. **Security of the ANRS:** Mitigating the risk of the ANRS itself becoming a target for attacks.
6. **Scalability and Performance Optimization:** Enabling the ANRS to operate efficiently within large and complex network environments.

Can a novel ANRS architecture be developed that leverages machine learning for anomaly detection, threat classification, and automated remediation, while addressing the limitations of existing approaches in terms of distributed policy enforcement, security of the self-healing architecture, and scalability? This research aims to answer this question by proposing a novel ANRS architecture, implementing key algorithms, and evaluating its performance through security analysis.

3.3 Proposed solution

A novel distributed system is proposed to address the limitations identified in current autonomous network remediation approaches (Chapter 2). The ANRS leverages advancements in machine learning, network security principles, and distributed computing to provide automated anomaly detection, threat classification, and remediation capabilities within a secure and scalable architecture.

3.3.1 ANRS Components and Functionalities

The ANRS adopts a hierarchical framework comprised of the following key components:

1. **Network Monitoring Agents (NMAs):** These lightweight software agents are deployed on individual network devices (routers, switches, endpoints) throughout the network. Their primary responsibilities include:

    i. **Data Collection:** NMAs collect network traffic data relevant for anomaly detection, such as packet headers, flow statistics, and application layer information.
    ii. **Basic Anomaly Detection:** NMAs perform basic anomaly detection using pre-defined rules or statistical techniques to identify potential security events locally. This can involve detecting unusual traffic volume, deviations from established application protocols, or suspicious source/destination IP addresses.
    iii. **Communication with CADE:** NMAs communicate relevant data and security event notifications to the Centralized Anomaly Detection Engine (CADE) for further analysis.

A depiction of the NMAs and their responsibilities is presented in Figure 3.1

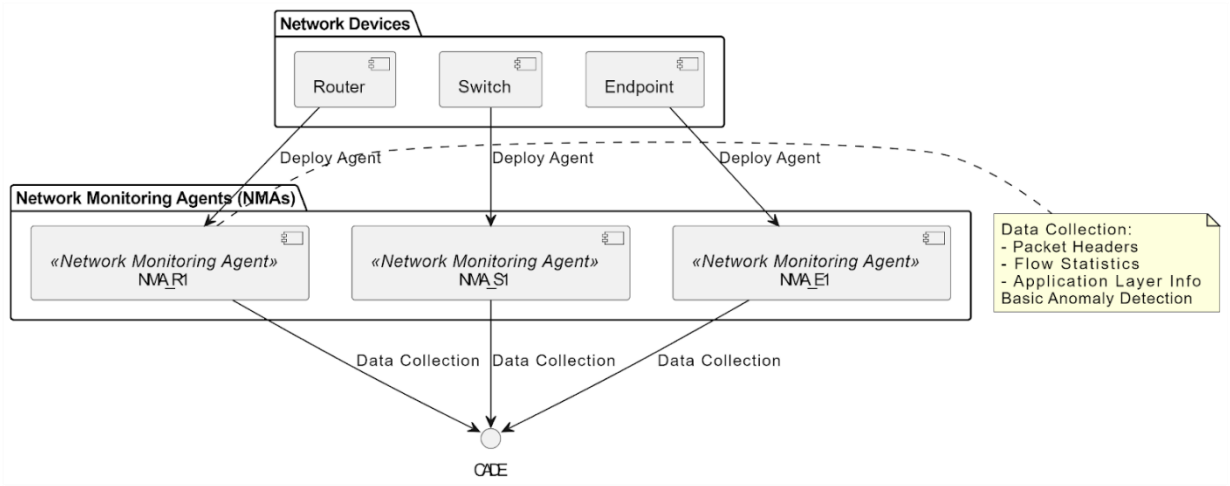

Figure 3.1: Network Monitoring Agents and their Functionalities

2. **Centralized Anomaly Detection Engine (CADE):** This core component resides in a secure central location, potentially within a centralized security management platform. The CADE is responsible for:

    i. **Data Aggregation and Processing:** The CADE receives data feeds from all deployed NMAs and pre-processes the collected data for further analysis.
    ii. **Advanced Anomaly Detection:** The CADE utilizes machine learning algorithms to perform advanced anomaly detection on the pre-processed network traffic data. These algorithms can identify complex anomalies that might not be apparent through basic rule-based detection methods employed by NMAs.
    iii. **Threat Identification:** Based on the analysis of network traffic data and anomaly scores generated by machine learning models, the CADE identifies potential security incidents.
    iv. **Remediation Triggering:** The CADE triggers appropriate remediation actions through the Policy Enforcement Module (PEM) and Self-healing Module (SHM) based on the identified threat type and severity.

    The CADE and its functionalities is presented in Figure 3.2

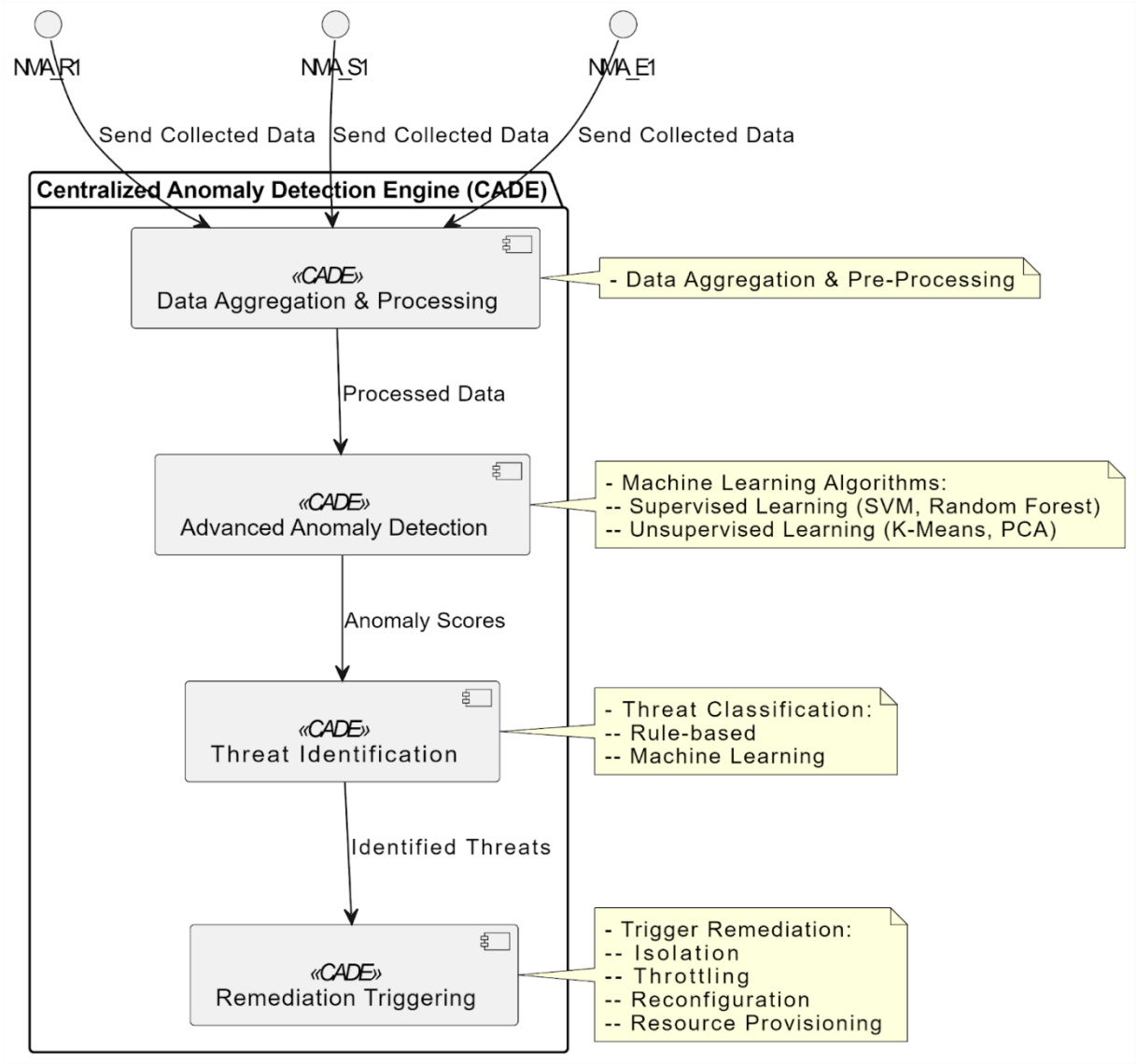

Figure 3.2: Centralized Anomaly Detection Engine and its Functionalities

3. **Policy Enforcement Module (PEM):** This module acts as a translator, converting high-level security policies defined by network administrators into actionable commands for network devices as seen in Figure 3.3. The PEM interacts with the CADE to determine the most suitable remediation actions based on the identified threat and enforces them on network devices through secure communication protocols like SNMP and NETCONF.

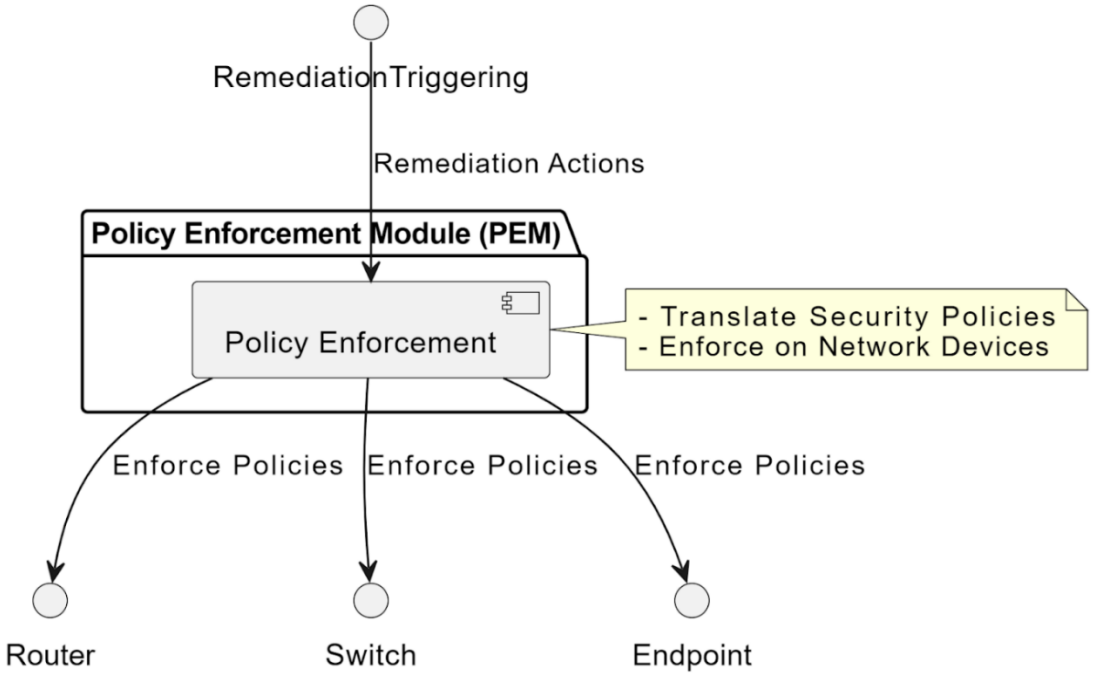

Figure 3.3 Policy Enforcement Module and its Functionalities

4. **Self-healing Module (SHM):** This module enables automated network reconfiguration in response to security incidents. As seen in Figure 3.4 it leverages Software-Defined Networking (SDN) controllers and network APIs to dynamically:

    i. **Isolate Compromised Devices:** Isolate compromised devices to prevent lateral movement within the network and minimize the potential damage caused by the threat.
    ii. **Reroute Traffic Flows:** Reroute traffic flows to maintain network functionality and minimize service disruption for legitimate users.
    iii. **Provision Additional Resources:** Provision additional resources (e.g., bandwidth) to mitigate the impact of an attack, such as a Denial-of-Service (DoS) attack.
    iv. **Threat Intelligence Integration:** The ANRS integrates with real-time threat intelligence feeds, providing it with up-to-date knowledge about emerging threats and vulnerabilities. This information is crucial for:
    v. **Refining Anomaly Detection:** Threat intelligence can be used to refine the machine learning models employed by the CADE, allowing for the identification of new attack patterns and improved detection accuracy.
    vi. **Tailoring Remediation Responses:** The CADE can leverage threat intelligence to prioritize and tailor automated remediation responses based on the latest threat landscape.

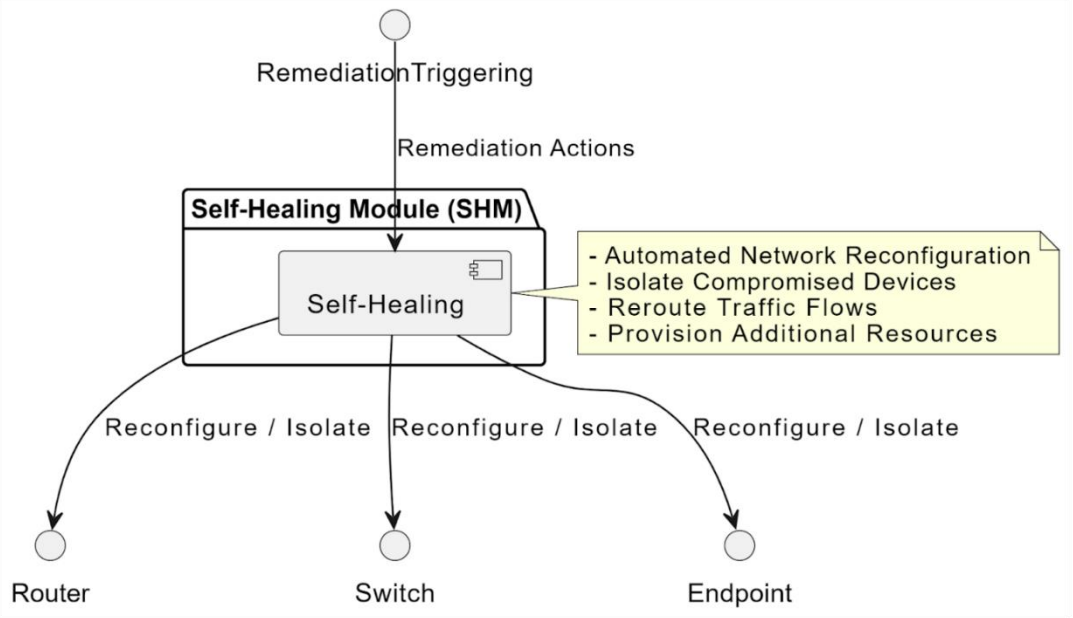

Figure 3.4 Self-healing Module and its Functionalities

5. **Network Communication:** NMAs will communicate with the CADE using a secure and lightweight protocol designed for efficient data exchange. The CADE will communicate with the PEM and SHM using secure communication channels within the central security management platform. The threat intelligence module shown in Figure 3.5 constantly provides threat updates to improve the efficiency of the system.

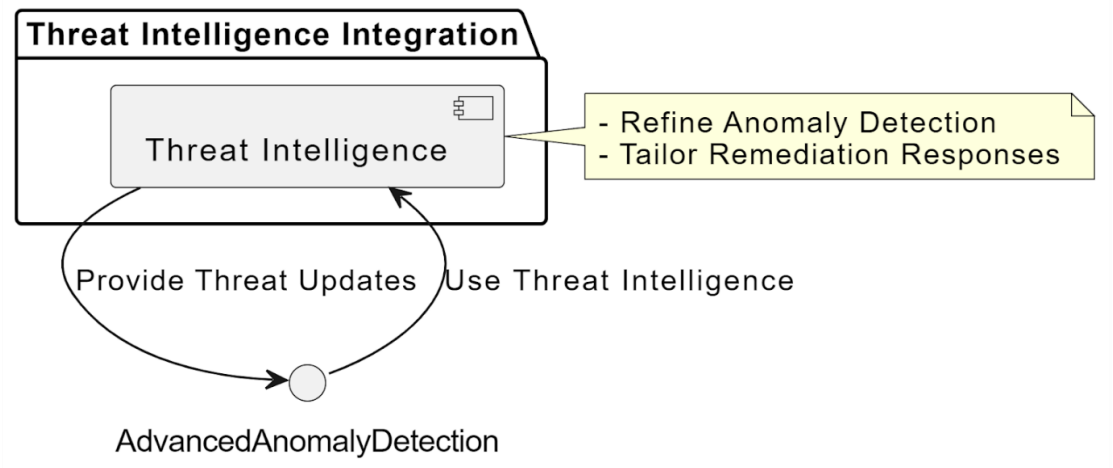

Figure 3.5 Threat Intelligence Integration

3.3.2 Algorithms for Anomaly Detection, Threat Classification, and Remediation

The ANRS employs a combination of algorithms to achieve its functionalities. Table 3.1 shows the Network Traffic Preprocessing Algorithm which resides on the Network Monitoring Agents (NMAs) and is responsible for cleaning, normalizing, and potentially performing feature engineering on the captured network traffic data. Feature engineering techniques aim to extract the most relevant features from the raw data to enhance the performance of the subsequent anomaly detection and threat classification algorithms.

Table 3.1: Algorithm for Network Monitoring Agents (NMAs)

Functionality	Pseudocode
Data Collection	function collect_data(): traffic_data = get_network_traffic() return traffic_data

Basic Anomaly Detection	function detect_basic_anomalies(data):
	if data.traffic_volume > THRESHOLD:
	return "unusual_traffic_volume"
	if data.protocol not in EXPECTED_PROTOCOLS:
	return "protocol_deviation"
	if data.source_ip in SUSPICIOUS_IPS:
	return "suspicious_ip"
	return None

Communication with CADE	function send_to_cade(anomaly): send(anomaly, CADE_ADDRESS)

The responsibilities of the CADE include data aggregation, advanced anomaly detection, threat identification, remediation triggering. Once an anomaly is detected by the CADE, threat classification algorithms are employed to categorize the potential threat type.

The CADE then selects and triggers pre-defined automated remediation algorithms based on the identified threat type and severity. These include:

1. **Isolation:** Isolating compromised devices to prevent them from further interacting with the network and causing damage. This can involve techniques like blackholing routes or dynamically modifying firewall rules.
2. **Throttling:** Limiting the network bandwidth allocated to suspicious traffic, potentially slowing down or stopping a denial-of-service attack.
3. **Reconfiguration:** Dynamically reconfiguring network paths using SDN controllers to bypass compromised network segments or redirect traffic flows for continued network operation.
4. **Resource Provisioning:** Provisioning additional resources (e.g., bandwidth, processing power) to mitigate the impact of an attack, such as dynamically scaling up resources for critical network functions.

The supervised learning algorithm is trained on labeled data from threat intelligence feeds. The training process allows the Support Vector Machines (SVMs) to learn a decision boundary that separates different threat classes. When presented with new, unlabeled network traffic data, the trained SVM model can classify these data points based on the learned patterns. Kernel functions (e.g., RBF kernel) will be explored to improve the SVM's ability to separate different threat classes in the feature space.

The selection of the most suitable remediation algorithm considers factors such as the identified threat type, potential impact on network performance, and the need to maintain service availability for legitimate users as shown in Table 3.2.

Table 3.2: Algorithm for Centralized Anomaly Detection Engine (CADE)

Functionality	Pseudocode
Data Aggregation and Processing	function aggregate_data(nmas_data): aggregated_data = preprocess(nmas_data) return aggregated_data
Advanced Anomaly Detection (Supervised)	function supervised_detection(data, model): features = extract_features(data) predictions = model.predict(features) anomalies = [data[i] for i in range(len(predictions)) if predictions[i] == 1] return anomalies
Advanced Anomaly Detection (Unsupervised)	function unsupervised_detection(data, model): features = extract_features(data) clusters = model.predict(features) anomaly_scores = compute_anomaly_scores(clusters, features) anomalies = [data[i] for i in range(len(anomaly_scores)) if anomaly_scores[i] > ANOMALY_THRESHOLD] return anomalies

Threat Identification	function identify_threats(anomalies):
	for anomaly in anomalies:
	threat_type = rule_based_classification(anomaly)
	if threat_type == 'Unknown Threat':
	threat_type = machine_learning_classification(anomaly, ml_model)
	anomaly.threat_type = threat_type
	return anomalies

Remediation Triggering	function trigger_remediation(anomalies):
	for anomaly in anomalies:
	if anomaly.threat_type == 'DDoSAttack':
	throttle_traffic(anomaly.source_ip)
	elif anomaly.threat_type == 'Protocol Anomaly':
	reconfigure_network(anomaly.path, anomaly.new_route)
	elif anomaly.threat_type == SQL-injection:
	isolate_device(anomaly.device)
	else:
	provision_resources(anomaly.service, EXTRA_BANDWIDTH)

Algorithms for policy translation, and policy enforcement which is carried out by the PEM is presented in Table 3.3. The policy translation algorithm takes the high-level policy definitions and translates them into a format that can be understood and implemented by the various network devices involved. This translation process ensures that the policies are correctly interpreted and enforced by each device, maintaining the overall security and integrity of the network. The policy enforcement algorithm is responsible for ensuring that all devices comply with the defined security standards, preventing security breaches and upholding the network's overall security posture.

Table 3.3: Algorithm for Policy Enforcement Module (PEM)

Functionality	Pseudocode
Policy Translation	Function translate_policy(high_level_policy): device_commands = generate_commands(high_level_policy) return device_commands
Policy Enforcement	function enforce_policy(commands): for command in commands: send_command_to_device(command.device, command.action)

The SHM algorithm includes several key components that work together to ensure seamless self-healing as show in Table 3.4.

Table 3.4: Algorithm for Self-Healing Module (SHM)

Functionality	Pseudocode
Isolation	function aggregate_data(nmas_data): aggregated_data = preprocess(nmas_data) return aggregated_data
Throttling	function supervised_detection(data, model): features = extract_features(data) predictions = model.predict(features) anomalies = [data[i] for i in range(len(predictions)) if predictions[i] == 1] return anomalies
Reconfiguration	function unsupervised_detection(data, model): features = extract_features(data) clusters = model.predict(features) anomaly_scores = compute_anomaly_scores(clusters, features) anomalies = [data[i] for i in range(len(anomaly_scores)) if anomaly_scores[i] > ANOMALY_THRESHOLD] return anomalies

Resource Provisioning	function identify_threats(anomalies):
	for anomaly in anomalies:
	threat_type = rule_based_classification(anomaly)
	if threat_type == 'Unknown Threat':
	threat_type = machine_learning_classification(anomaly, ml_model)
	anomaly.threat_type = threat_type
	return anomalies

The algorithm for threat intelligence integration automates the collection of threat intelligence data from various sources to ensure that the security system remains up-to-date with the latest threat intelligence, enabling it to detect and respond to emerging threats more effectively as seen in Table 3.5

Table 3.5: Algorithm for Threat Intelligence Integration

Functionality	Pseudocode
Threat Intelligence Processing	function integrate_threat_intelligence(data): new_threats = fetch_threat_intelligence() update_detection_models(new_threats)

3.3.3 Addressing Limitations of Existing Systems

The proposed ANRS specifically addresses the limitations identified in existing autonomous network remediation systems. The Policy Enforcement Module (PEM) serves as a central point for translating high-level security policies into device-specific commands. This ensures consistent enforcement of security policies across the entire network, regardless of the location or type of network device. Network monitoring agents communicate security events and anomalies to the CADE, which then interacts with the PEM to determine and enforce the appropriate remediation actions based on the predefined security policies as seen in Figure 3.6

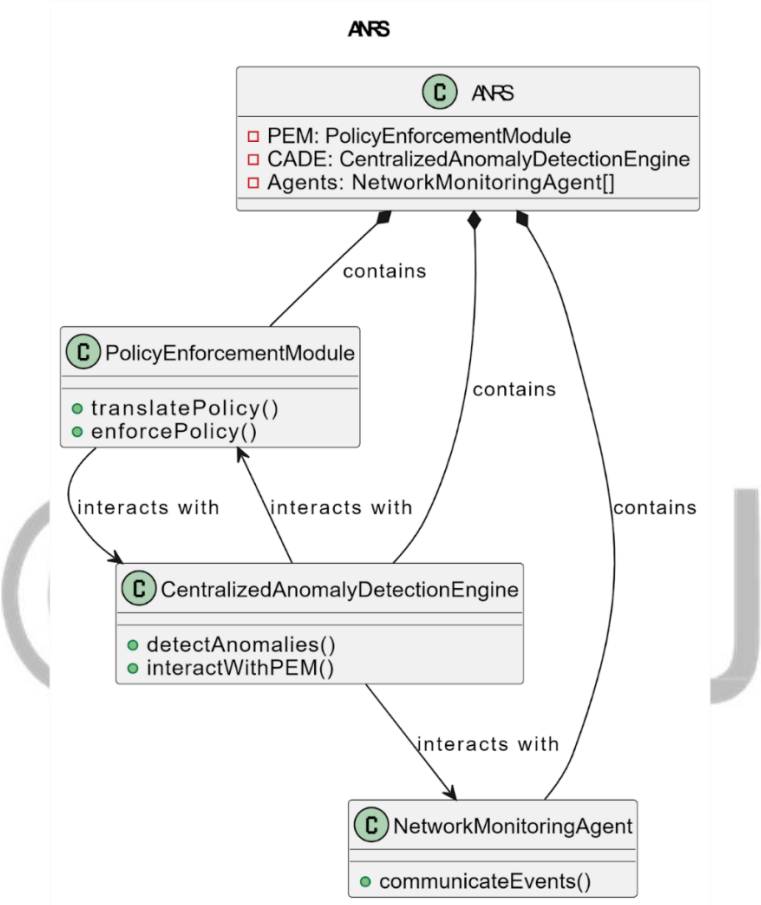

Figure 3.6: Interactions of the ANRS components

Additionally, the CADE and PEM reside in a secure central location with robust security measures in place, such as firewalls, intrusion detection systems, and access controls as seen in Figure 3.7. This mitigates the risk of these critical components being compromised by attackers. Additionally, anomaly detection techniques can be implemented within the ANRS itself to identify and isolate potential threats targeting the system. Techniques like monitoring system logs for suspicious activity or employing anomaly detection algorithms to analyze communication patterns within the ANRS can be used for self-protection.

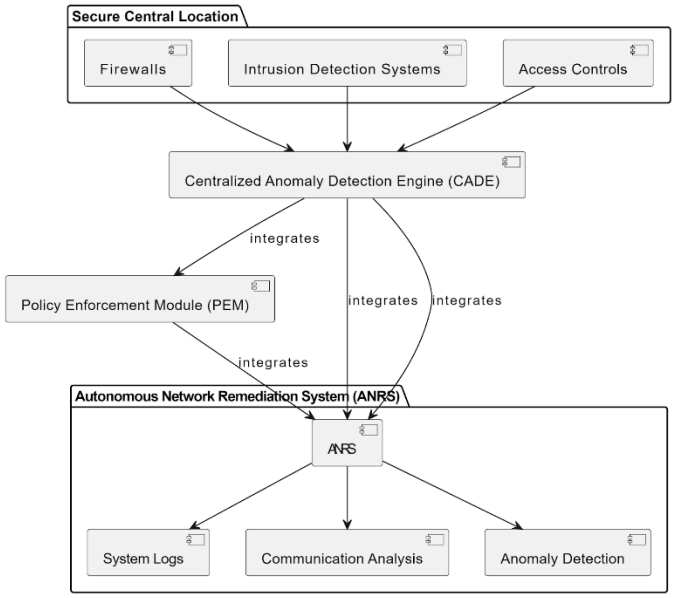

Figure 3.7: Module Integrations in the ANRS

The ANRS is designed for scalability and efficient operation within large and complex network environments. Network monitoring agents are lightweight and designed to collect only essential network traffic data, minimizing communication overhead. The CADE leverages distributed processing techniques, potentially utilizing parallel computing or cloud-based resources, to handle large volumes of data efficiently. Additionally, the use of pre-defined and optimized remediation algorithms ensures timely responses to security incidents without compromising network performance. Figure 3.8 depicts a summary of the ANRS and its components.

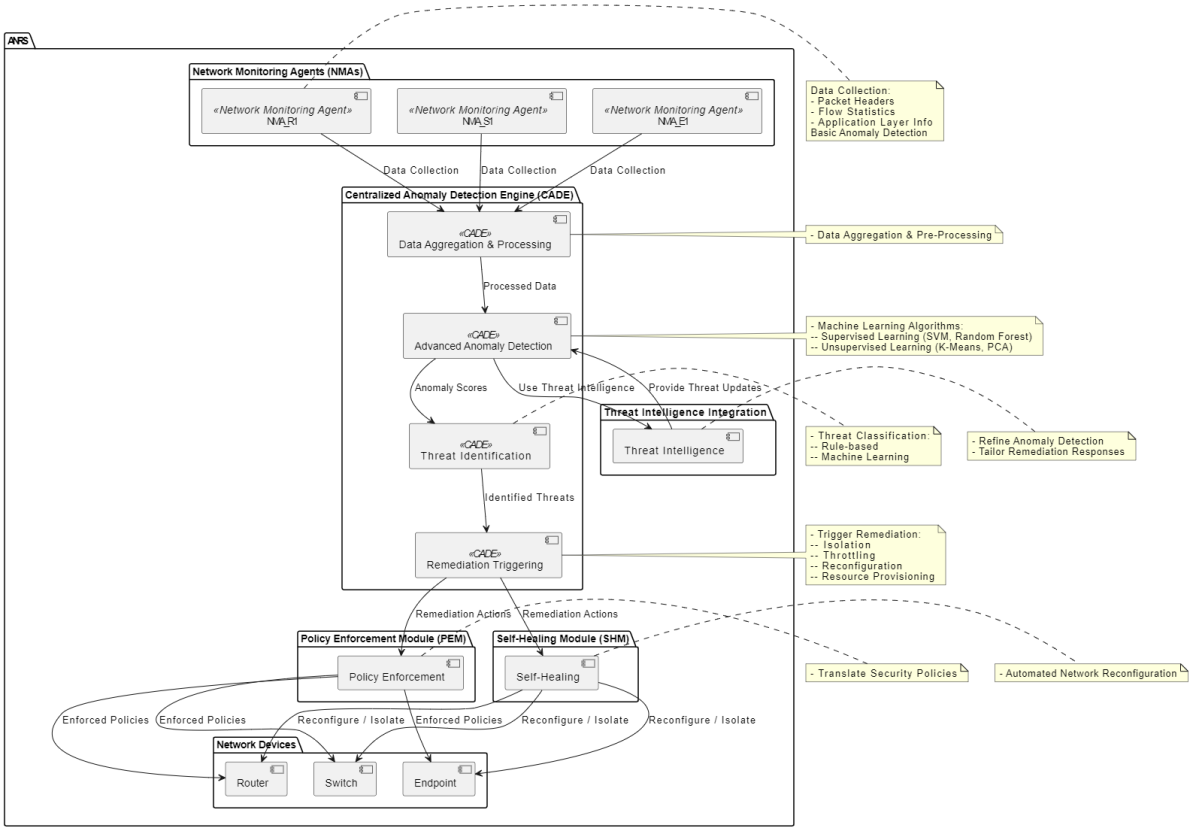

Figure 3.8: Summary of the ANRS

3.4 Tools Used in the Implementation

This research proposes a novel Autonomous Network Remediation System (ANRS) architecture. To achieve the research goals of implementing key algorithms, evaluating performance, and conducting security analysis, a carefully chosen set of tools and technologies are essential.

3.4.1 Data Used in the ANRS

The effectiveness of the ANRS relies heavily on the quality and relevance of the data it utilizes. Here's a breakdown of the data sources that will be crucial for the ANRS's operation:

1. **Network Traffic Data:** This is the primary data source for the ANRS. NMAs deployed on network devices will capture network traffic data relevant for anomaly detection. This data can include:

    i. **Packet Headers:** Information contained in the header of each network packet, such as source and destination IP addresses, port numbers, and protocol types.
    ii. **Flow Statistics:** Aggregated statistics about network traffic flows, including flow duration, packet count, and byte volume. This provides a high-level overview of network traffic patterns.
    iii. **Application Layer Information:** A deeper inspection of application layer data (e.g., HTTP requests and responses) will be beneficial for detecting application-specific anomalies. However, this can raise privacy concerns and requires careful consideration.

2. **Threat Intelligence Feeds:** This data includes specific signatures or patterns associated with malicious activities, such as IP addresses of known malicious servers or malware signatures, details about known threat actors, their tactics, techniques, and procedures (TTPs), which can help the ANRS identify potential attacks based on behavioral patterns, and information about known vulnerabilities in network devices, operating systems, and applications. This data can be used by the ANRS to prioritize remediation actions for critical vulnerabilities.

The raw data collected from network traffic and threat intelligence feeds will undergo preprocessing and feature engineering before being used for anomaly detection and threat classification. This process involves:

i. **Data Cleaning:** Removing irrelevant or corrupted data entries to ensure the integrity of the data used for analysis.
ii. **Data Transformation:** Transforming data into a format suitable for machine learning algorithms. This might involve scaling numerical features, encoding categorical features, and extracting relevant features from network traffic data.
iii. **Feature Engineering:** Creating new features that might be more informative for anomaly detection. This could involve calculating statistical features of network traffic flows or extracting patterns from application layer data.

Python is the chosen language for developing the modules in the ANRS due to its efficiency, portability, and extensive libraries specifically designed for network programming tasks. Python libraries like Pandas and NumPy will be utilized for data manipulation, pre-processing, and feature extraction from the captured network traffic data to prepare it for machine learning analysis within the Centralized Anomaly Detection Engine (CADE). A robust and secure operating system, such as Linux, will serve as the foundation for the CADE. To build, train, and deploy machine learning models for anomaly detection. Data visualization tools like Matplotlib or Seaborn in Python will be used to gain insights into network traffic patterns and visually identify potential anomalies.

For an actual implementation in a real-time threat intelligence feeds like STIX/TAXII or commercial threat intelligence platforms can be integrated to provide the ANRS with up-to-date information about emerging threats and vulnerabilities. Libraries like TAXII client libraries or custom-developed parsers can be used to ingest and process threat intelligence data in a format compatible with the ANRS's internal systems. To protect the ANRS components from unauthorized access and cyber-attacks, security measures like firewalls, intrusion detection/prevention systems (IDS/IPS), and user authentication mechanisms can also be implemented. Firewalls and IDS/IPS will monitor network traffic for suspicious activity, while user authentication ensures only authorized users can access and modify the ANRS's configuration. A comprehensive logging and monitoring system will be implemented to track the ANRS's activities, identify potential issues, and ensure its overall health. Logs will record events such as detected anomalies, remediation actions taken, and system errors. Monitoring tools will provide real-time insights into the ANRS's performance and resource utilization.

3.5 Approach and Techniques for the Proposed Solution

The approach taken for the proposed Autonomous Network Remediation System centers on algorithm development. Analyzing network data to discover anomalies and classify threats is a very challenging task. A strong and flexible strategy is necessary due to the natural unpredictability of traffic patterns and the constant appearance of new hazards. One algorithm alone may not be enough to effectively detect and address the wide variety of abnormalities and threats that occur in real-world network systems. The use of many algorithms in this method has numerous benefits:

1. Enhanced Precision in Detecting Anomalies and Increased Ability to Classify Threats Across Different Scenarios: By applying a mix of unsupervised and supervised learning approaches, the ANRS may accomplish a larger spectrum of anomaly capture and more accurate threat categorization. Unsupervised techniques such as K-Means clustering may detect anomalies in regular traffic patterns, but supervised learning using Support Vector Machines (SVMs) enables classification using labeled threat intelligence data.
2. A specialized Security Policy Enforcement Algorithm will be created to guarantee compliance with predetermined network security regulations. This algorithm will transform high-level security policies into actionable rules, allowing the ANRS to dynamically alter network settings (e.g., firewall rules, access control lists) based on real-time threat detection and categorization.
3. Flexible and Adaptive Self-healing Remediation: A Self-healing Remediation Algorithm will be created to automate the process of reducing detected hazards. This algorithm will utilize the threat categorization findings and pre-defined remedial procedures to automatically react to security events. The self-healing capabilities will boost the overall responsiveness and resilience of the network security posture.

The efficacy of the ANRS rests on the smooth integration and coordinated execution of the established algorithms. The core ANRS algorithms will be integrated following a modular design approach. This approach promotes flexibility and facilitates future modifications or additions to the algorithm suite. The envisioned integration process can be summarized as follows:

1. **Data Flow Management:** Network traffic data captured by the NMAs will be preprocessed by the Network Traffic Preprocessing Algorithm. The preprocessed data will then be fed into the anomaly detection and threat classification algorithms.
2. **Anomaly Detection and Threat Classification:** The unsupervised anomaly detection algorithms will operate in parallel, identifying potential anomalies in the network traffic data. The supervised threat classification algorithm will further analyze the flagged anomalies and unclassified data points, classifying them into specific threat categories based on the trained model.
3. **Security Policy Enforcement and Self-healing Remediation:** The Security Policy Enforcement Algorithm will translate the threat classification results into actionable rules. These rules will be

dynamically applied to network configurations to enforce security policies and mitigate identified threats. The Self-healing Remediation Algorithm will leverage the threat classification and pre-defined remediation actions to automatically respond to security incidents.

4. **Feedback Loop:** To enhance the overall effectiveness of the ANRS over time, a feedback loop will be incorporated. The system will continuously monitor the performance of the algorithms and network security posture. The feedback loop will allow for potential adjustments to the algorithms (e.g., retraining SVMs with new threat intelligence data) and security policies to adapt to evolving threats and network environments.

3.6 Research Design

The design of the ANRS adheres to a structured approach, leveraging the Unified Modeling Language (UML) to visually represent the research process and its activities.

3.6.1 Research Methodology

The research methodology adopted for this project is Design Science Research (DSR). DSR aligns well with the project's objective of developing a novel artifact, the ANRS, to address a specific problem in network security: the need for more automated anomaly detection, threat classification, and remediation capabilities.

The DSR methodology follows an iterative cycle consisting of the following key phases:

1. **Problem Identification and Motivation:** This phase identified the limitations of existing network security solutions, particularly the need for increased automation in anomaly detection, threat classification, and remediation.
2. **Definition of Objectives and Requirements:** The research objectives were defined to address the identified problem. These objectives focused on developing the ANRS with functionalities like anomaly detection, threat classification, security policy enforcement, and self-healing remediation. Additionally, system requirements were established, outlining the desired functionalities, performance metrics, and non-functional requirements of the ANRS.
3. **Design and Development:** This phase involves the development of the ANRS using core algorithms, code implementation, and system integration. UML diagrams will be extensively used in this phase to visualize the research activities.
4. **Evaluation:** The developed ANRS will be rigorously evaluated using various techniques, including performance testing, functional testing, and penetration testing. This phase will assess the effectiveness of the ANRS in achieving its objectives and meeting the defined requirements.
5. **Iteration and Refinement:** Based on the evaluation results, the ANRS design and implementation may be iteratively refined to address identified shortcomings or improve performance. This iterative feedback loop ensures the ANRS continuously adapts and enhances its effectiveness.

3.6.2 Research Process with UML Diagrams

The UML offers a rich set of diagrams to visually represent the research process within the DSR framework. Here's how UML diagrams will be utilized to detail the research activities for the ANRS development:

1. **Use Case Diagram:** This diagram depicts the high-level interactions between actors (e.g., network device, network administrator, security analyst) and the ANRS. It illustrates use cases such as submitting threat intelligence data, configuring security policies, and receiving security alerts from the ANRS. (See Figure 3.9)

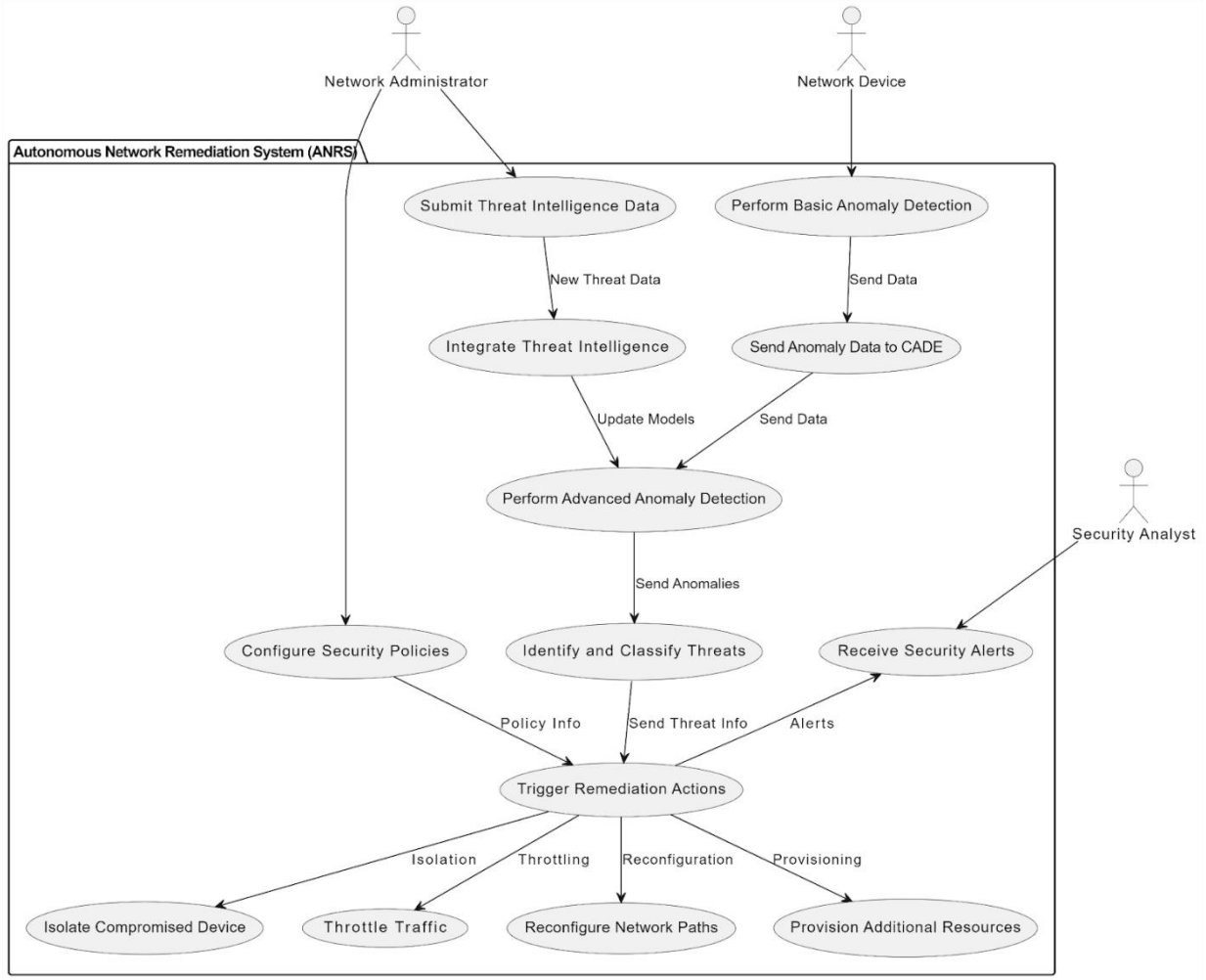

Figure 3.9: Use Case Diagram of the ANRS

Actors:

1. **Network Administrator:** Responsible for configuring security policies and submitting threat intelligence data.
2. **Security Analyst:** Receives security alerts from the ANRS.
3. **Network Device:** Performs basic anomaly detection and sends data to CADE.

Use Cases:

1. **Submit Threat Intelligence Data:** Network administrators provide threat intelligence to the ANRS.
2. **Configure Security Policies:** Network administrators set up security policies that guide remediation actions.
3. **Receive Security Alerts:** Security analysts receive alerts about detected threats and remediation actions from the ANRS.
4. **Perform Basic Anomaly Detection:** Network devices (through NMAs) perform basic anomaly detection.
5. **Send Anomaly Data to CADE:** NMAs send detected anomaly data to the CADE for further analysis.
6. **Perform Advanced Anomaly Detection:** The CADE performs advanced anomaly detection using machine learning techniques.
7. **Identify and Classify Threats:** The CADE identifies and classifies threats based on detected anomalies.

8. **Trigger Remediation Actions:** Based on the identified threats, the CADE triggers appropriate remediation actions.
9. **Isolate Compromised Device:** The SHM isolates compromised devices.
10. **Throttle Traffic:** The SHM throttles suspicious traffic to mitigate potential threats.
11. **Reconfigure Network Paths:** The SHM reconfigures network paths to avoid compromised segments.
12. **Provision Additional Resources:** The SHM provisions additional resources to maintain network performance during attacks.
13. **Integrate Threat Intelligence:** The ANRS integrates new threat intelligence data to refine detection models.

2. **Activity Diagram:** This diagram models the workflow of the ANRS. It shows the sequential steps involved in network traffic capture, anomaly detection, threat classification, security policy enforcement, and automated remediation actions. This diagram provides a clear understanding of the system's internal processes. (See Figure 3.10).

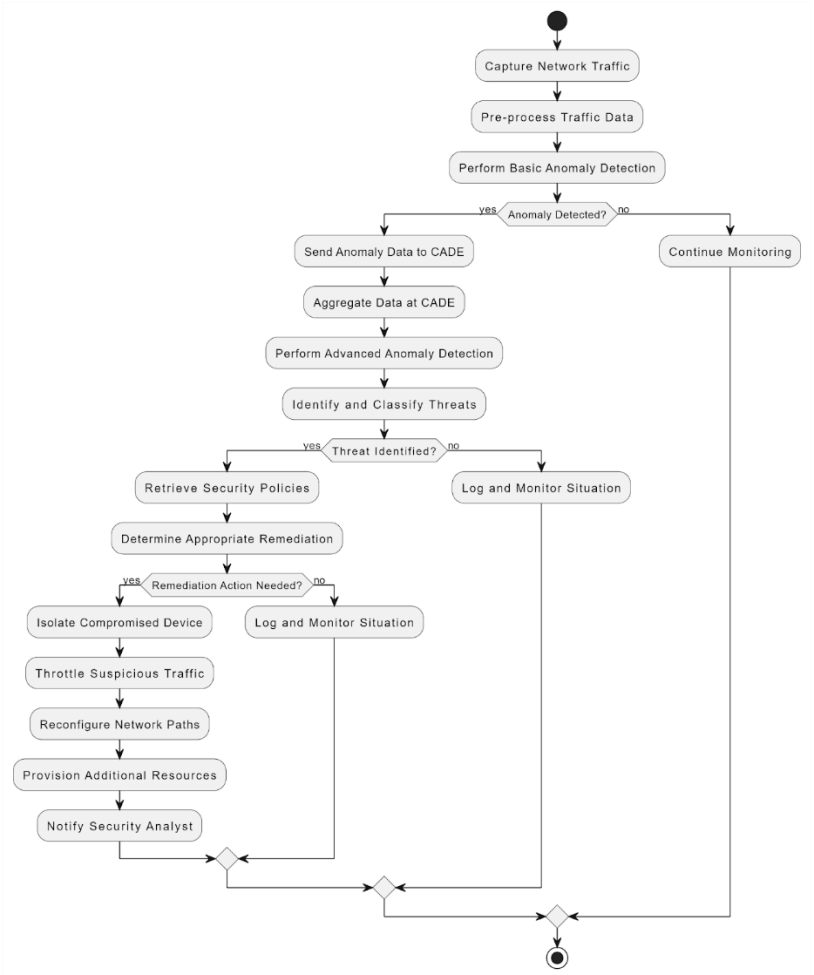

Figure 3.10: Activity Diagram of the ANRS

Activities:

1. **Capture Network Traffic:** Continuous capture of network traffic data from network devices.
2. **Pre-process Traffic Data:** Pre-process the captured traffic data to extract relevant features.
3. **Perform Basic Anomaly Detection:** Basic anomaly detection using predefined rules at NMAs.

4. **Anomaly Detected?:** Check if an anomaly is detected by NMAs.
    - **Yes:** If an anomaly is detected, proceed to send anomaly data to CADE.
    - **No:** If no anomaly is detected, continue monitoring network traffic.
5. **Send Anomaly Data to CADE:** NMAs send detected anomaly data to the CADE.
6. **Aggregate Data at CADE:** CADE aggregates data from all NMAs.
7. **Perform Advanced Anomaly Detection**: Advanced anomaly detection using machine learning models at CADE.
8. **Anomaly Confirmed?:** Confirm if the anomaly is valid through advanced detection.
    - **Yes:** If the anomaly is confirmed, proceed to identify and classify threats.
    - **No:** If the anomaly is not confirmed, log the situation and continue monitoring.
9. **Identify and Classify Threats:** CADE identifies and classifies threats based on detected anomalies.
10. **Threat Identified?:** Check if a threat is identified.
    - **Yes:** If a threat is identified, retrieve security policies and determine remediation.
    - **No:** If no threat is identified, log the situation and continue monitoring.
11. **Retrieve Security Policies:** Retrieve relevant security policies for remediation.
12. **Determine Appropriate Remediation:** Determine the appropriate remediation action based on the threat and policies.
13. **Remediation Action Needed?:** Check if a remediation action is necessary.
    - **Yes:** If remediation is needed, select and execute the appropriate action:
        - Isolate Compromised Device: Isolate the compromised device.
        - Throttle Suspicious Traffic: Throttle suspicious traffic.
        - Reconfigure Network Paths: Reconfigure network paths to avoid compromised segments.
        - Provision Additional Resources: Provision additional resources to maintain performance.
        - Notify Security Analyst: Notify the security analyst about the remediation actions taken.
    - **No:** If no remediation is needed, log the situation and continue monitoring.

3. **Class Diagram:** This diagram represents the key classes and objects within the ANRS architecture. It illustrates the attributes, methods, and relationships between these classes, offering a structural view of the system's components. Examples of classes could include Network Traffic Object, Anomaly Data Object, Threat Classification Model, Security Policy Object, and Remediation Action Object. (See Figure 3.11)

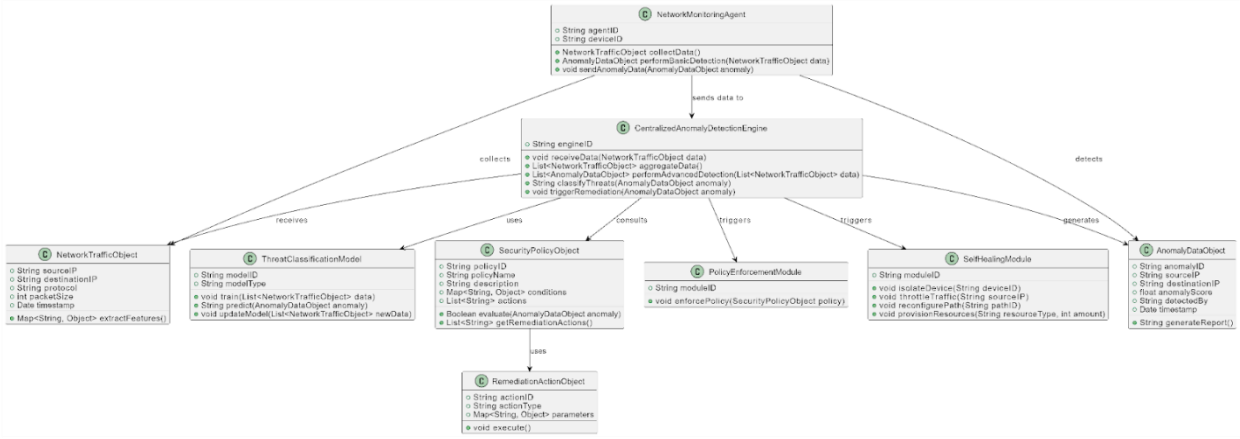

Figure 3.11: Class Diagram of the ANRS

Classes:

1. **NetworkTrafficObject:**
   Attributes: sourceIP, destinationIP, protocol, packetSize, timestamp.

Methods: extractFeatures().

2. **AnomalyDataObject:**
   Attributes: anomalyID, sourceIP, destinationIP, anomalyScore, detectedBy, timestamp.
   Methods: generateReport().
3. **ThreatClassificationModel:**
   Attributes: modelID, modelType.
   Methods: train(data), predict(anomaly), updateModel(newData).
4. **SecurityPolicyObject:**
   Attributes: policyID, policyName, description, conditions, actions.
   Methods: evaluate(anomaly), getRemediationActions().
5. **RemediationActionObject:**
   Attributes: actionID, actionType, parameters.
   Methods: execute().
6. **NetworkMonitoringAgent:**
   Attributes: agentID, deviceID.
   Methods: collectData(), performBasicDetection(data), sendAnomalyData(anomaly).
7. **CentralizedAnomalyDetectionEngine:**
   Attributes: engineID.
   Methods: receiveData(data), aggregateData(), performAdvancedDetection(data), classifyThreats(anomaly), triggerRemediation(anomaly).
8. **PolicyEnforcementModule:**
   Attributes: moduleID.
   Methods: enforcePolicy(policy).
9. **SelfHealingModule:**
   Attributes: moduleID.
   Methods: isolateDevice(deviceID), throttleTraffic(sourceIP), reconfigurePath(pathID), provisionResources(resourceType, amount).

Relationships:

1. NetworkMonitoringAgent collects NetworkTrafficObject data and detects AnomalyDataObject.
2. NetworkMonitoringAgent sends anomaly data to CentralizedAnomalyDetectionEngine.
3. CentralizedAnomalyDetectionEngine receives NetworkTrafficObject data, generates AnomalyDataObject, uses ThreatClassificationModel for classification, consults SecurityPolicyObject for policies, and triggers both PolicyEnforcementModule and SelfHealingModule for remediation actions.
4. SecurityPolicyObject uses RemediationActionObject for defining remediation actions.

4. **Sequence Diagram:** This diagram focuses on specific scenarios within the ANRS. It depicts the detailed message exchanges between components during the anomaly detection process, and the interaction between the ANRS and network devices when enforcing security policies. This diagram highlights the communication flow between objects during specific functionalities. (See Figure 3.12)

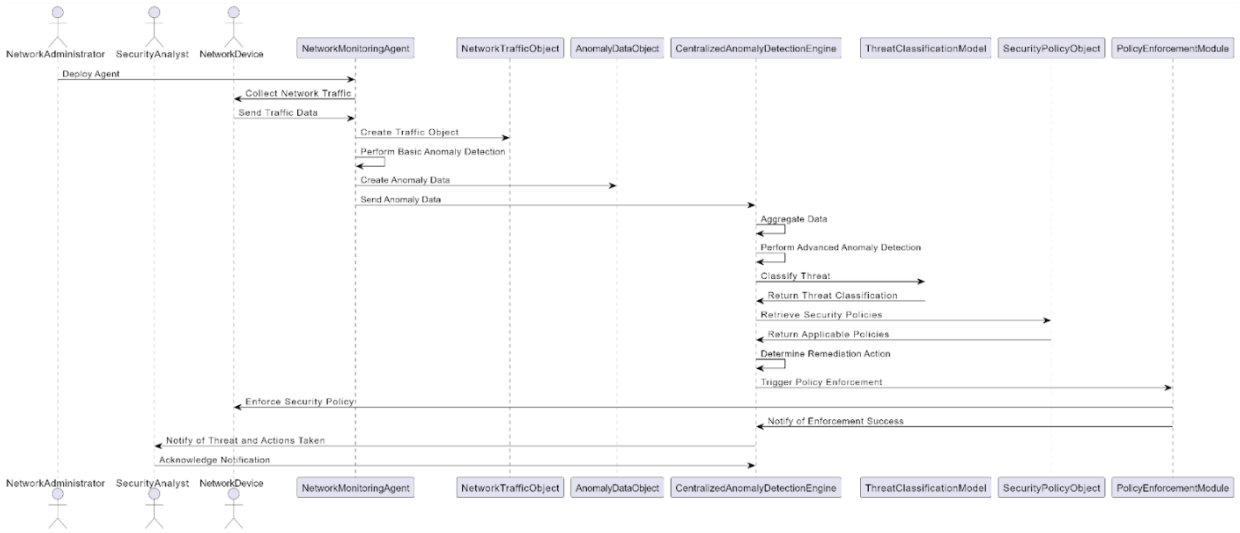

Figure 3.12: Sequence Diagram of the ANRS

Participants:

1. NetworkAdministrator: Deploys and configures the NetworkMonitoringAgent.
2. SecurityAnalyst: Monitors the system and receives notifications about threats and actions taken.
3. NetworkDevice: Devices from which the NetworkMonitoringAgent collects traffic data.
4. NetworkMonitoringAgent: Collects network traffic data, performs basic anomaly detection, and sends anomaly data to CADE.
5. CentralizedAnomalyDetectionEngine (CADE): Aggregates data, performs advanced anomaly detection, classifies threats, retrieves security policies, and triggers remediation actions.
6. ThreatClassificationModel: Classifies the identified anomalies into specific threats.
7. SecurityPolicyObject: Contains security policies that guide the remediation actions.
8. PolicyEnforcementModule: Enforces the security policies on the network devices.

Sequence of Events:

1. NetworkAdministrator deploys the NetworkMonitoringAgent.
2. NetworkMonitoringAgent collects network traffic data from the NetworkDevice.
3. NetworkDevice sends traffic data to the NetworkMonitoringAgent.
4. NetworkMonitoringAgent creates a NetworkTrafficObject with the collected data.
5. NetworkMonitoringAgent performs basic anomaly detection on the traffic data.
6. Upon detecting an anomaly, NetworkMonitoringAgent creates an AnomalyDataObject.
7. NetworkMonitoringAgent sends the AnomalyDataObject to the CentralizedAnomalyDetectionEngine (CADE).
8. CADE aggregates data from multiple sources.
9. CADE performs advanced anomaly detection using its algorithms.
10. CADE classifies the detected anomaly by querying the ThreatClassificationModel.
11. ThreatClassificationModel returns the classification result to CADE.
12. CADE retrieves the relevant security policies from the SecurityPolicyObject.
13. SecurityPolicyObject returns the applicable policies to CADE.
14. CADE determines the appropriate remediation action based on the policies and threat classification.
15. CADE triggers the PolicyEnforcementModule to enforce the determined security policy.
16. PolicyEnforcementModule enforces the security policy on the affected NetworkDevice.
17. PolicyEnforcementModule notifies CADE of the successful enforcement.
18. CADE notifies the SecurityAnalyst about the detected threat and the actions taken.

19. SecurityAnalyst acknowledges the notification from CADE.

5. **State Machine Diagram:** This diagram is employed to represent the dynamic behavior of specific components within the ANRS. It illustrates the state transitions of a network device as the ANRS enforces security policies (e.g., transitioning from "normal" to "blocked" state upon detecting malicious activity).

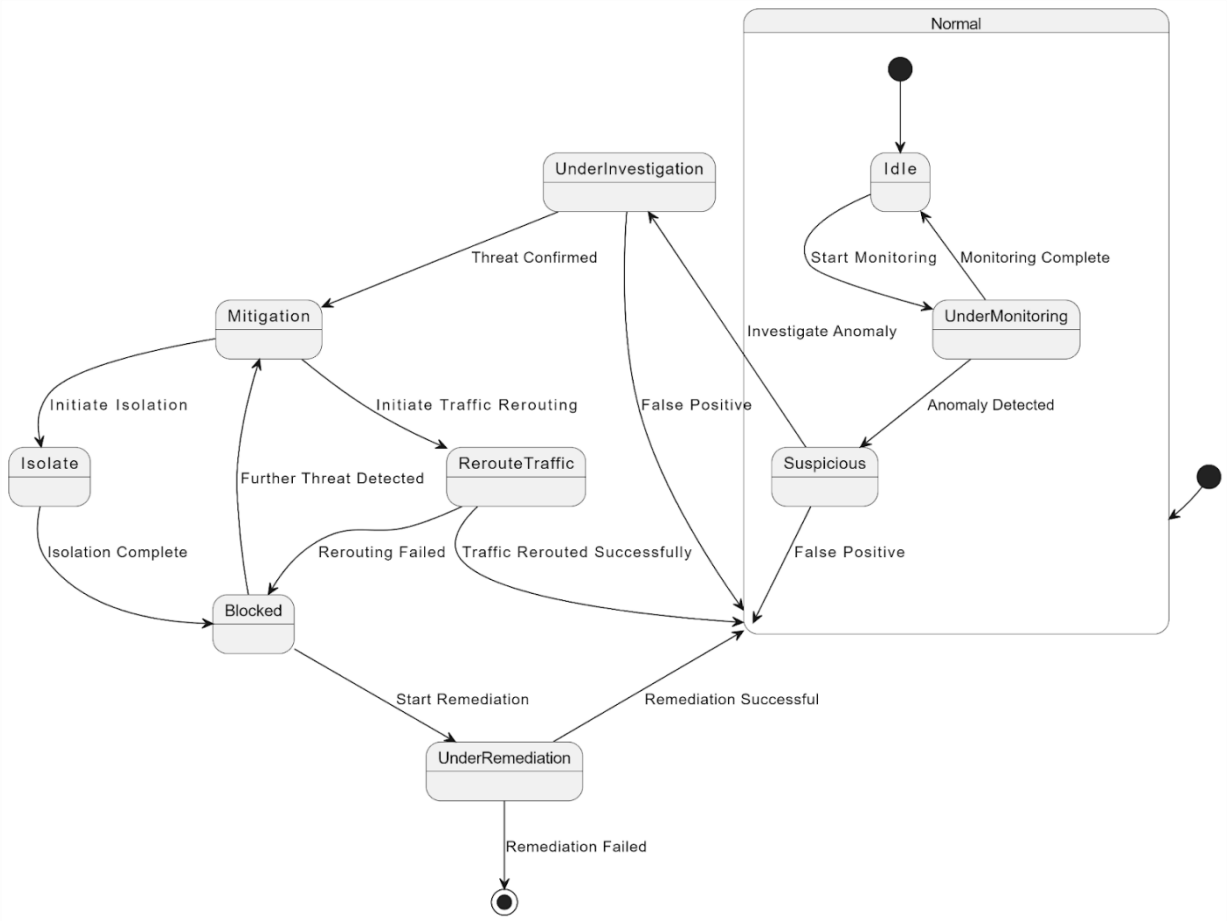

Figure 3.13: State Machine Diagram of the ANRS

States:

1. **Normal:**
   - **Idle:** The network device is operating normally without any ongoing monitoring.
   - **UnderMonitoring:** The network device is actively being monitored for any suspicious activities.
   - **Suspicious:** An anomaly or suspicious activity is detected on the network device.

2. **Suspicious:**
   - UnderInvestigation: The detected anomaly is being investigated to confirm if it is a genuine threat.
   - Mitigation: The threat is confirmed, and mitigation actions are initiated.
   - Normal: The anomaly was determined to be a false positive, and the device returns to a normal state.
3. **Mitigation:**

- Isolate: The device is isolated from the network to prevent further damage.
- RerouteTraffic: Traffic is rerouted to maintain network functionality while mitigating the threat.

4. Blocked:

- UnderRemediation: The device is undergoing remediation actions to neutralize the threat and restore normal operations.

Transitions:

1. Normal -> Suspicious: Transition occurs when suspicious activity is detected on the network device.
2. Suspicious -> UnderInvestigation: Transition occurs to investigate the detected anomaly.
3. UnderInvestigation -> Mitigation: Transition occurs when the anomaly is confirmed as a threat.
4. UnderInvestigation -> Normal: Transition occurs if the detected anomaly is a false positive.
5. Mitigation -> Isolate: Transition occurs to initiate isolation of the device.
6. Mitigation -> RerouteTraffic: Transition occurs to initiate traffic rerouting.
7. Isolate -> Blocked: Transition occurs when the isolation is complete.
8. RerouteTraffic -> Normal: Transition occurs when traffic rerouting is successful.
9. RerouteTraffic -> Blocked: Transition occurs if traffic rerouting fails.
10. Blocked -> UnderRemediation: Transition occurs to start the remediation process.
11. UnderRemediation -> Normal: Transition occurs if the remediation is successful.
12. UnderRemediation -> [*]: Transition occurs if the remediation fails.
13. Blocked -> Mitigation: Transition occurs if further threat detection requires additional mitigation.

3.7 Description of Validation Techniques for Proposed Solution

The effectiveness of the Autonomous Network Remediation System was assessed using a comprehensive validation strategy encompassing various techniques. Due to limitations in setting up a controlled testbed environment to replicate a real-world network scenario, the CICIDS 2017 Dataset, a widely recognized and comprehensive dataset for network intrusion detection research, was used for validation. This dataset offers a rich collection of labeled network traffic data encompassing various benign and malicious activities.

Here's an outline of the key experimental procedures:

1. **Dataset Preprocessing:** The CICIDS 2017 Dataset was downloaded and preprocessed to ensure compatibility with the ANRS algorithms. This involved handling missing values, scaling numerical features, and feature engineering.
2. **Data Splitting:** The preprocessed dataset was split into training, validation, and testing sets. The training set was used to train the anomaly detection and threat classification algorithms. The validation set was used for hyperparameter tuning and model selection. Finally, the testing set was used for the final performance evaluation of the ANRS.
3. **ANRS Configuration and Training:** The ANRS was configured with the developed algorithms and trained on the prepared datasets:

    - A hybrid model was trained for enhanced anomaly detection.
    - The SVM model for threat classification was trained using the labeled attack traffic data from the CICIDS 2017 Dataset and incorporating additional threat intelligence feeds.

- Security policies were defined to translate threat classifications into actionable rules for automated network configuration adjustments.
- The self-healing remediation module was configured with pre-defined actions to respond to identified threats.

4. **Performance Evaluation:** The ANRS was subjected to a series of tests to assess its performance in various aspects:

    - **Anomaly Detection Accuracy:** The effectiveness of anomaly detection algorithms was evaluated by measuring metrics such as accuracy, precision, recall, and F1-score. This analysis involved assessing the system's ability to correctly identify anomalies within the CICIDS 2017 Dataset.
    - **Threat Classification Accuracy:** The performance of the SVM model was assessed using metrics like accuracy, confusion matrix analysis, and receiver operating characteristic (ROC) curve analysis. This evaluated the system's ability to accurately classify identified anomalies into specific threat categories based on the labeled data.
    - **Security Policy Enforcement and Remediation Effectiveness:** The evaluation focused on the ANRS's capability to translate security policies into rules and simulate the process of applying those rules to achieve remediation actions, even though the CICIDS 2017 Dataset may not directly reflect real-world network configurations.
    - **Performance Scalability:** The ANRS was evaluated using various techniques to simulate increasing traffic volume within the CICIDS 2017 Dataset. This assessed the system's ability to handle potential real-world network load variations without compromising performance or accuracy.

5. **Penetration Testing with Simulated Scenarios:** Simulated penetration testing scenarios were conducted using the CICIDS 2017 Dataset. By injecting attack traffic patterns representative of real-world threats, the ANRS's detection, classification, and response capabilities against various attack vectors were evaluated.

While the CICIDS 2017 Dataset offers a valuable resource for validation, it's important to acknowledge the limitations of relying solely on pre-recorded data. Real-world network environments are constantly evolving, and the ANRS may encounter novel attack patterns not represented in the dataset.

Future work may involve exploring techniques to continuously adapt the ANRS models using online learning algorithms or incorporating real-time network traffic data for ongoing validation and performance improvement. Additionally, if resources become available, conducting validation within a controlled testbed environment can further enhance the generalizability of the results.

3.8 Description of Performance Evaluation Parameters/Metrics

The effectiveness of the ANRS was assessed using a comprehensive suite of performance metrics encompassing various aspects of its functionality. Here's a breakdown of key metrics for each ANRS stage:

3.8.1 Anomaly Detection

1. **Accuracy:** Measures the overall correctness of the anomaly detection model in identifying both anomalies and normal traffic. It's calculated as the ratio of correctly classified instances (true positives and true negatives) to the total number of instances.

2. **Precision:** Evaluates the model's ability to identify true anomalies without mistakenly flagging normal traffic as anomalies. It's calculated as the ratio of true positives to the total number of positive predictions (true positives and false positives).
3. **Recall:** Measures the model's capability to detect actual anomalies without missing them. It's calculated as the ratio of true positives to the total number of actual anomalies (true positives and false negatives).
4. **F1-score:** Provides a harmonic mean between precision and recall, offering a balanced view of the model's performance. A higher F1-score indicates a better overall anomaly detection capability.

3.8.2 Threat Classification

1. **Accuracy:** Similar to anomaly detection, accuracy measures the model's correctness in classifying identified anomalies into specific threat categories. It's calculated as the ratio of correctly classified threat types to the total number of classified anomalies.
2. **Confusion Matrix:** A visual representation of the model's performance across different threat categories. It details the number of true positives, false positives, true negatives, and false negatives for each category, providing insights into potential classification errors.
3. **Receiver Operating Characteristic (ROC) Curve:** A graphical plot illustrating the model's ability to distinguish between anomalies and normal traffic. It depicts the True Positive Rate (TPR) on the y-axis against the False Positive Rate (FPR) on the x-axis. A larger area under the ROC curve (AUC) signifies better threat classification performance.

3.8.3 Remediation Effectiveness

1. **Response Time:** Measures the time taken by the ANRS to identify a threat, classify it, and initiate the pre-defined remediation actions. Faster response times are desirable to minimize the potential impact of attacks.
2. **Remediation Success Rate:** Evaluates the effectiveness of the implemented remediation actions in successfully mitigating the identified threats. It's calculated as the ratio of successfully mitigated threats to the total number of identified threats.

3.9 System Architecture

The ANRS was designed with a modular and scalable architecture to facilitate efficient threat detection, classification, and automated remediation. The System Components include the following:

1. **Network Traffic Capture and Preprocessing Module:** Continuously captures network traffic data using packet capture libraries in addition to performing data preprocessing tasks like cleaning (handling missing values), normalization (scaling features), and potentially feature engineering to extract relevant characteristics from the captured traffic.
2. **Anomaly Detection Module:** Leverages machine learning algorithms to identify deviations from normal network traffic patterns.
3. **Threat Classification Module:** Analyzes the identified anomalies from the anomaly detection module and employs a Support Vector Machine (SVM) model trained on labeled network traffic data (containing both benign and malicious traffic) to classify anomalies into specific threat categories.
4. **Security Policy Engine:** Maps the threat classifications to pre-defined security policies. These policies translate the identified threat types into actionable rules for the network infrastructure. The policies can dynamically adjust firewall rules, access control lists (ACLs), or trigger other automated security measures based on the severity of the threat.

5. **Self-Healing Remediation Module:** Executes pre-configured remediation actions based on the security policy directives received from the Security Policy Engine. Examples of actions include isolating infected devices, blocking malicious traffic at the network perimeter, or quarantining compromised systems.
6. **Management and Monitoring Console:** Provides a centralized interface for system administrators to monitor the ANRS's operational status. It also offers functionalities for configuration management, reviewing security logs, visualizing network traffic patterns, and potentially fine-tuning anomaly detection and threat classification models.
7. **Threat Intelligence Integration:** Integrates with external threat intelligence feeds to stay updated on the latest attack signatures and emerging threats. This information can be used to continuously update the SVM model for improved threat classification accuracy.

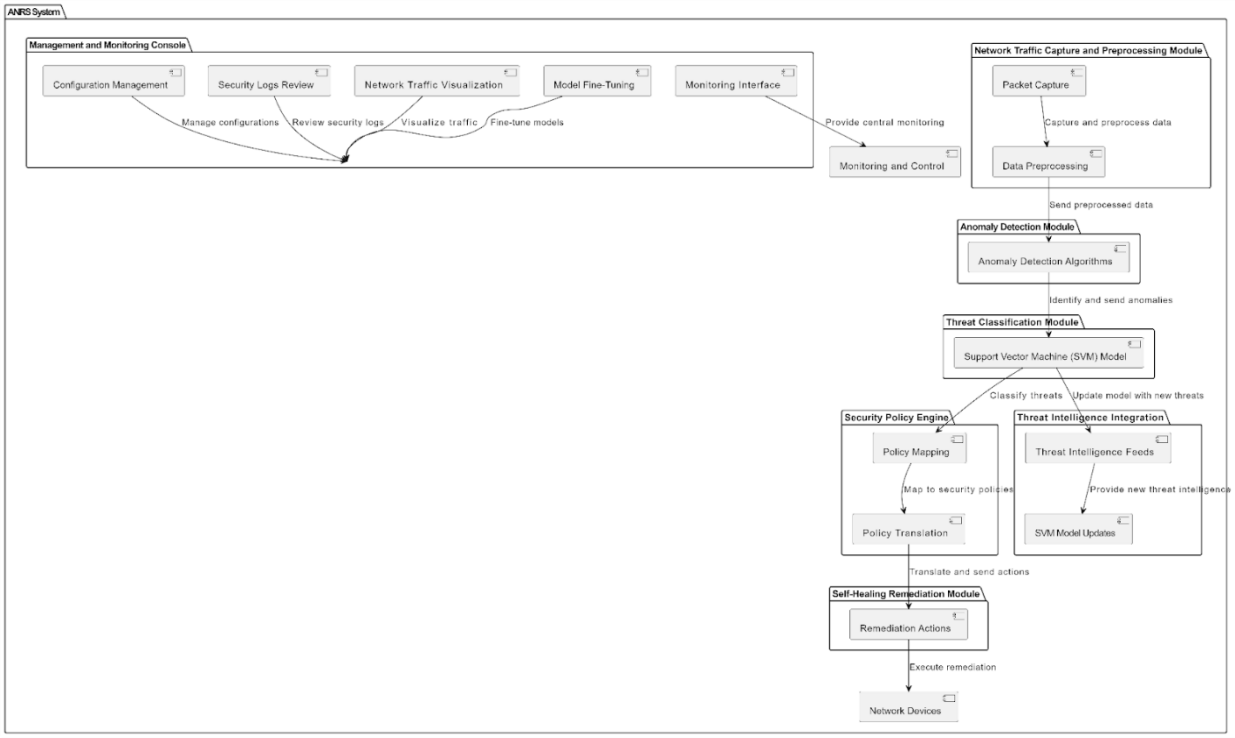

Figure 3.14: System Architecture

System Interactions:

1. Captured network traffic data flows from the Network Traffic Capture and Preprocessing Module to the Anomaly Detection Module.
2. The Anomaly Detection Module analyzes the preprocessed data and identifies potential anomalies.
3. Identified anomalies are passed to the Threat Classification Module for further analysis.
4. The Threat Classification Module utilizes the trained SVM model to classify the anomalies into specific threat categories.
5. The Security Policy Engine receives the threat classification and maps it to relevant security policies.
6. The Security Policy Engine translates the policies into actionable rules and communicates them to the network infrastructure for dynamic configuration adjustments.
7. The Self-Healing Remediation Module executes pre-defined actions based on the received rules to mitigate the identified threats.
8. The Management and Monitoring Console provides a central hub for system administrators to oversee the entire ANRS operation.

9. The Threat Intelligence Integration module interacts with external threat intelligence feeds and updates the SVM model for enhanced threat detection.

The benefits of the adopted architecture include but are not limited to the following:

1. **Modularity:** The system is designed with independent modules, facilitating easier maintenance, scalability, and potential integration with future security tools.
2. **Scalability:** The architecture can be scaled horizontally by adding additional processing units to handle increased network traffic volume.
3. **Automation:** The system automates network threat detection, classification, and response, reducing manual intervention and improving reaction times.
4. **Adaptability:** The machine learning models can be continuously updated with new data, allowing the ANRS to adapt to evolving threats.
5. **Centralized Management:** The Management and Monitoring Console provides a central point for overseeing the entire ANRS operation.

This modular and scalable architecture provides a robust foundation for the ANRS, enabling efficient network threat mitigation and enhancing overall network security posture.

Chapter 4: Results and Discussion

4.1 Preamble

This chapter presents and discusses the results obtained from the implementation and testing of the Autonomous Network Remediation System (ANRS) using the "Intrusion Detection Evaluation Dataset" (CICIDS2017). Due to resource limitations, a controlled lab environment was not feasible. Instead, the Thursday-WorkingHours-Morning-WebAttacks.pcap_ISCX.csv file from the CICIDS2017 dataset was used to evaluate the system's performance. This dataset provides a realistic representation of network traffic and includes labeled instances of web attacks, making it suitable for assessing the ANRS's capabilities in anomaly detection, threat classification, and automated remediation. The chapter details the evaluation setup, presents the results of the initial and hybrid models tested, analyzes these results, discusses their implications, and benchmarks the ANRS against previous studies. The goal is to comprehensively understand the system's performance and identify areas for improvement.

4.2 System Evaluation

The evaluation of the ANRS was conducted using the CICIDS2017 dataset, specifically the Thursday-WorkingHours-Morning-WebAttacks.pcap_ISCX.csv file. This dataset includes network traffic data with labeled instances of web attacks, providing a realistic environment for testing the ANRS's performance.

The performance metrics used for evaluation include:

1. **Accuracy**: The proportion of correctly identified instances among the total instances.
2. **Precision**: The proportion of true positive results among all positive results predicted by the model.
3. **Recall**: The proportion of true positive results among all actual positive instances.
4. **F1 Score**: The harmonic mean of precision and recall, providing a single metric that balances both concerns.

4.3 Results Presentation

The initial evaluation of the ANRS using the CICIDS2017 dataset revealed the following performance metrics for anomaly detection and threat classification as seen in Table 4.1

Table 4.1: Classification Report

Threat Type	Precision	Recall	F1-Score	Support
BENIGN	1.00	1.00	1.00	33,603
Web Attack – Brute Force	0.57	0.88	0.69	324
Web Attack – Sql Injection	0.00	0.00	0.00	6
Web Attack – XSS	1.00	0.04	0.08	114
Overall Accuracy		0.99		34,047
Macro Average	0.64	0.48	0.44	34,047
Weighted Average	0.99	0.99	0.99	34,047

4.3.1 Hybrid Model Performance

To improve the performance metrics, several hybrid models were explored. The results for each model are as follows:

a. **K-Means Clustering Model:** Initial K-Means clustering for anomaly detection yielded poor performance (accuracy: 0.43, precision: 0.12, recall: 0.22) as seen in Table 4.2 Clustering was employed as a pre-processing step to potentially improve anomaly detection in hybrid models by reducing dimensionality, handling imbalanced data, and identifying anomalous clusters. Results suggested further optimization of clustering parameters, exploration of alternative clustering algorithms, and potential feature engineering

Table 4.2: Performance Evaluation of K-means Clustering Model

Metric	Value
Accuracy	0.4339
Precision	0.1166
Recall	0.2170
F1 Score	0.1517

As seen in Figures 4.1 as 4.2, the confusion matrix and cluster plots indicate that the current clustering approach needed further refinement.

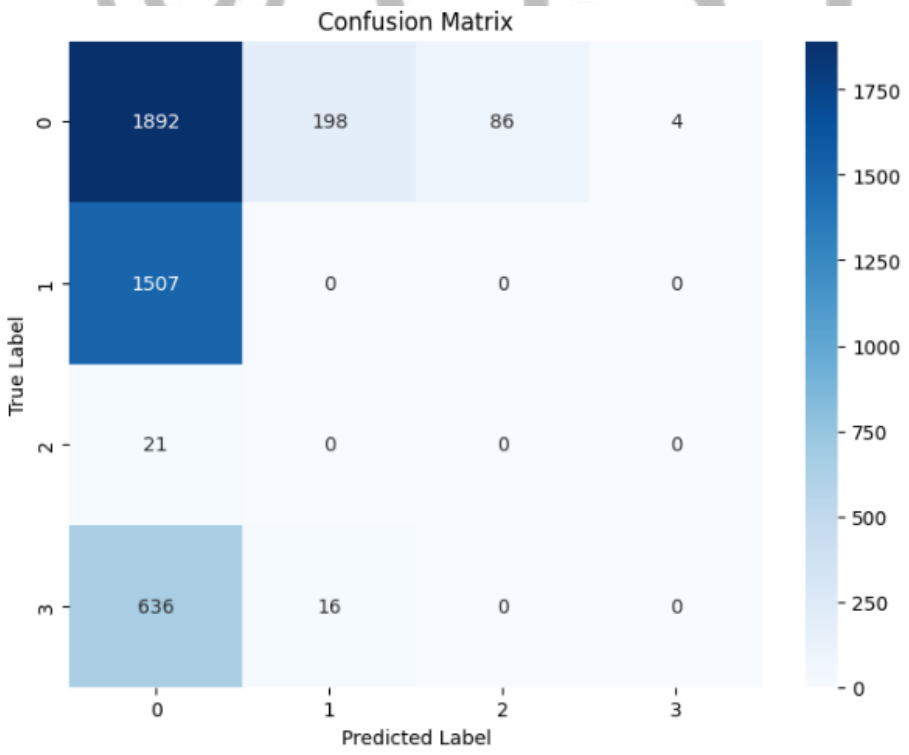

Figure 4.1: Confusion Matrix of the K-means Clustering Model

Figure 4.2: Clustered Data of the K-means Clustering Model

b. Hybrid Model: K-Means and One-Class SVM (OCSVM): This hybrid approach leverages the strengths of both K-Means clustering and One-Class SVM (OCSVM) to enhance anomaly detection. K-Means pre-processes the data by identifying potential anomalous clusters, guiding the OCSVM to focus on specific regions likely to contain anomalies. This targeted approach addresses limitations of using either algorithm alone, potentially improving overall performance as seen in Table 4.3. However, due to the computational expense of OCSVM, the dataset was split for processing, which may have impacted the final results. By combining K-Means' ability to capture non-linear patterns with OCSVM's boundary learning for anomaly detection, the hybrid model was able to achieve a slightly higher accuracy, precision, and recall.

Table 4.3: Performance Evaluation of K-means and One-Class SVM (OCSVM)

Metric	Value
Accuracy	0.3411
Precision	0.0810
Recall	0.1973
F1 Score	0.1148

c. **Dimensionality Reduction with PCA and Isolation Forest:** This approach was explored to tackle the challenges of high-dimensional data and anomaly detection by combining Principal Component Analysis (PCA) for dimensionality reduction and Isolation Forest for anomaly detection. PCA reduces computational complexity and enhances the performance of Isolation Forest by transforming the original features into a smaller set of uncorrelated variables. This method is particularly beneficial for high-dimensional datasets, as it addresses the "curse of dimensionality" and improves the efficiency of anomaly detection. However, despite the theoretical advantages, the performance results of this approach were still suboptimal, indicating a need for further optimization or exploration of alternative techniques as seen in Table 4.4.

Table 4.4: Performance Evaluation of PCA and Isolation Forest

Metric	Value
Accuracy	0.4383
Precision	0.1590
Recall	0.2229
F1 Score	0.1693

d. **Hybrid Model: K-Means and Decision Tree:** This hybrid model combines K-Means clustering and Decision Tree classification to achieve highly effective anomaly detection. The model leverages K-Means to group similar network traffic data points into clusters, effectively reducing dimensionality and pre-screening for potential anomalies. The Decision Tree then learns a set of rules based on the original features and cluster assignments to classify data points as normal or anomalous. The exceptional performance metrics, including near-perfect accuracy and precision, demonstrate the model's ability to minimize false positives and accurately identify anomalies as seen in Table 4.5.

Table 4.5: Performance Evaluation of K-Means and Decision Tree

Metric	Value
Accuracy	0.9890
Precision	0.9932
Recall	0.9215
F1 Score	0.9514

This hybrid approach is particularly well-suited for an autonomous network remediation system due to its several key advantages. The high recall ensures effective anomaly detection and protection against network threats. Both K-Means and Decision Trees are computationally efficient, enabling real-time analysis and response. The Decision Tree's rule-based structure provides interpretability, facilitating understanding and potential fine-tuning of the system. Moreover, the model can be retrained periodically with new data to adapt to evolving network traffic patterns and emerging threats. The combination of exceptional performance, real-time capability, interpretability, and adaptability makes this K-Means and Decision Tree hybrid model a compelling choice for enhancing network security and resilience through autonomous remediation.

The classification report provides a detailed breakdown of the model's performance for each class of network traffic; For class 0 (BENIGN), the model achieves a precision of 0.99, meaning that 99% of the instances predicted as benign were actually benign. The recall of 0.99 indicates that the model correctly identified 99% of all actual benign instances. The F1-score, a balanced measure of precision and recall, is also 0.99, confirming excellent performance for this class. Class 1 (Web Attack – Brute Force) also shows strong results with a precision of 0.98, recall of 0.99, and an F1-score of 0.99, indicating the model's effectiveness in detecting this type of web attack. Class 2 (Web Attack – Sql Injection) has a perfect precision of 1.00, meaning all instances predicted as SQL injection attacks were indeed true positives. However, the recall of 0.71 suggests that the model missed identifying 29% of actual SQL injection attacks. This leads to a lower F1-score of 0.83, indicating room for improvement in detecting this specific

attack type. Finally, class 3 (Web Attack – XSS) demonstrates perfect precision and recall (both 1.00), resulting in a perfect F1-score of 1.00. This indicates flawless detection of cross-site scripting attacks by the model. Overall, the classification report highlights the model's strong performance across most classes, with potential for further improvement in detecting SQL injection attacks as seen in Table 4.6.

Table 4.6: Classification Report of K-Means and Decision Tree

Class	Precision	Recall	F1-Score
0 (BENIGN)	0.99	0.99	0.99
1 (Web Attack – Brute Force)	0.98	0.99	0.99
2 (Web Attack – Sql Injection)	1.00	0.71	0.83
3 (Web Attack – XSS)	1.00	0.99	1.00

The confusion matrix shown in Figure 4.4 reveals that the hybrid model combining K-Means and Decision Tree performs well, particularly for classes 0, 1, and 3, with high accuracy and few misclassifications. However, it struggles more with class 2, showing some confusion with classes 0 and 1. This indicates that while the model is generally effective, there is room for improvement in distinguishing less frequent or more similar classes, which could enhance overall classification accuracy.

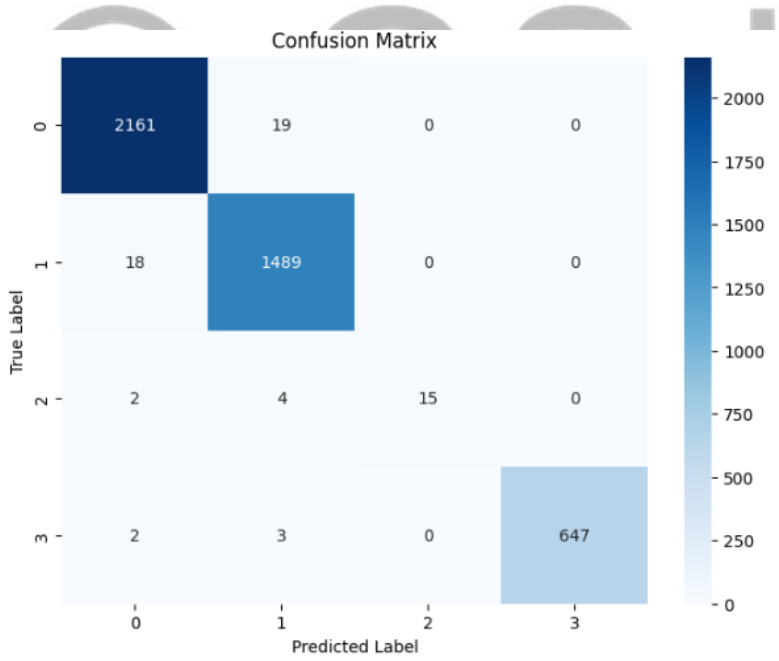

Figure 4.3: Confusion Matrix of the K-means and Decision Tree

The ROC curve is a graphical representation of a classifier's performance across all classification thresholds, plotting the True Positive Rate (TPR) against the False Positive Rate (FPR). As seen in Figure 4.4, the ROC curve for Class 0, with an AUC of 0.01, suggests poor performance in distinguishing Class 0 from other classes, indicating that the model's true positive rate is only slightly better than random guessing when considering all thresholds. Despite high accuracy and precision overall, the ROC curve reveals that the model struggles specifically with the trade-off between sensitivity (true positive rate) and specificity (false positive rate) for Class 0, suggesting room for improvement in the classification balance for this specific class.

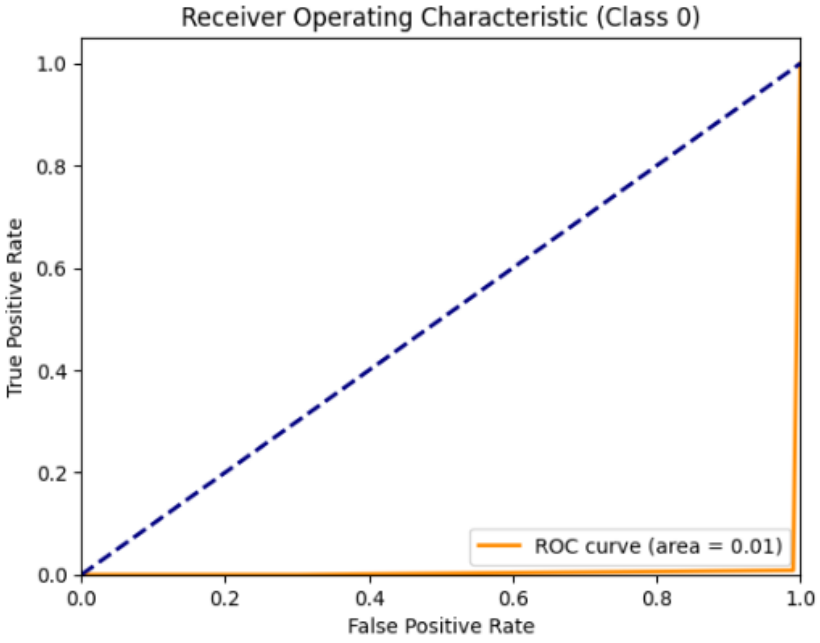

Figure 4.4: Receiver Operating Characteristic (Class 0)

As seen in Figure 4.5, the ROC curve for Class 1, with an AUC of 0.76, indicates that the model has a relatively good ability to distinguish Class 1 from other classes. An AUC of 0.76 suggests that the model performs well, although not perfectly, in identifying true positives (Class 1 instances) while maintaining a low false positive rate. This means the model has a decent balance between sensitivity and specificity for Class 1, performing better than random guessing but still having some room for improvement.

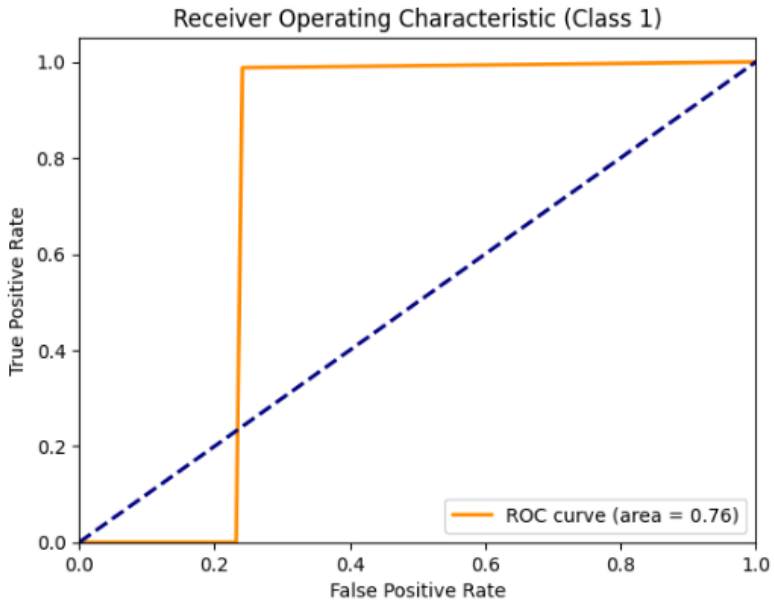

Figure 4.5: Receiver Operating Characteristic (Class 1)

As seen in Figure 4.6, the ROC curve for Class 2, with an AUC of 0.76, indicates that the model has a reasonably good ability to distinguish Class 2 from other classes. An AUC of 0.76 means that the model can correctly identify true positives (Class 2 instances) with a fair level of accuracy, while also keeping the false positive rate relatively low. This suggests that the model performs well in balancing sensitivity and specificity for Class 2, though there is still room for improvement to enhance its discriminative power.

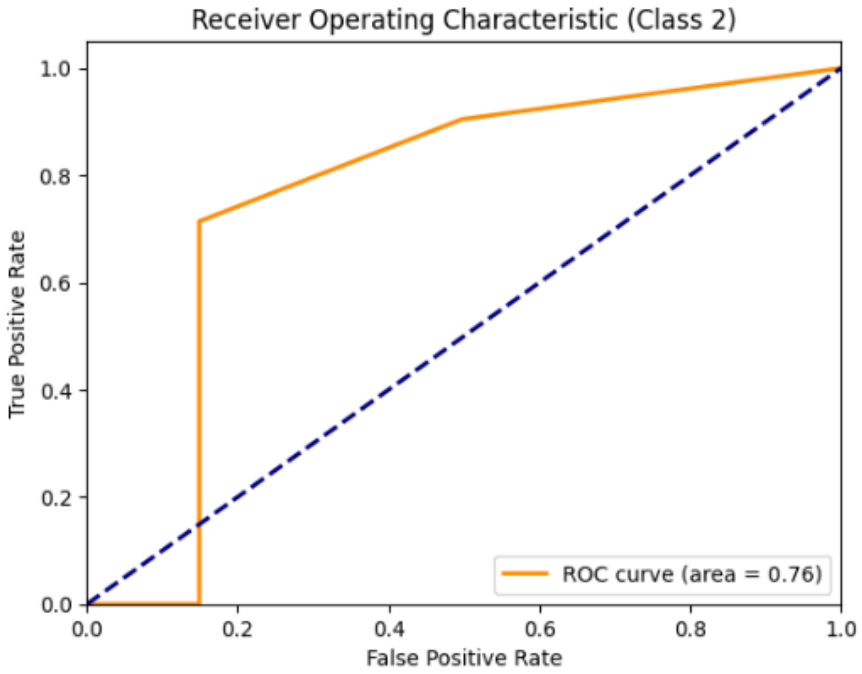

Figure 4.6: Receiver Operating Characteristic (Class 2)

The ROC curve for Class 3, with an AUC of 1.00 shown in Figure 4.7, indicates that the model perfectly distinguishes Class 3 from all other classes. An AUC of 1.00 means that the model has 100% sensitivity (true positive rate) and 100% specificity (false positive rate), implying that it correctly identifies all instances of Class 3 without any false positives or false negatives. This level of performance is ideal and signifies that the model is highly effective for Class 3 classification.

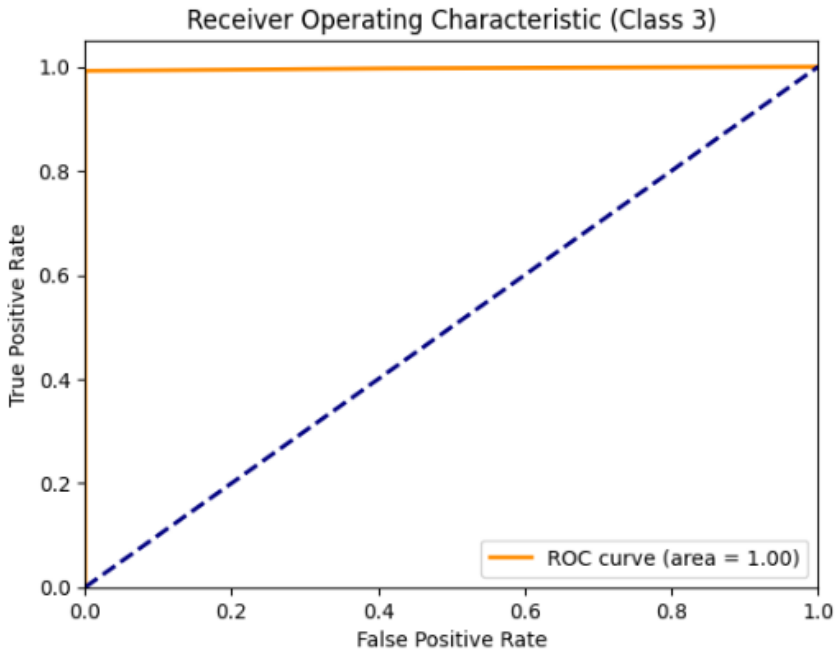

Figure 4.7: Receiver Operating Characteristic (Class 3)

The PR curves are essential tools for evaluating the model's accuracy in predicting positive cases, especially in scenarios where the class distribution is imbalanced. As seen in Figure 4.8, for Class 0, the Average Precision (AP) is 0.50. The PR curve for this class starts with high precision, which quickly

declines as recall increases. This indicates that while the model initially makes accurate predictions, its precision drops significantly as it tries to capture more true positives. This performance suggests that the model can identify true positives with moderate accuracy but faces challenges in maintaining precision, likely due to noisy or overlapping features between Class 0 and other classes.

Class 1 exhibits an AP of 0.68, with a PR curve that maintains higher precision across a broader range of recall values compared to Class 0. This implies that the model performs well for Class 1, achieving a good balance between precision and recall. The stability of precision across different recall values indicates that the features used to classify Class 1 are well-defined and less noisy, enabling the model to distinguish Class 1 instances more reliably. In contrast, Class 2 presents significant challenges with an AP of only 0.02. The PR curve for Class 2 is notably low and flat, reflecting poor performance where precision drops even at low recall values. This poor performance suggests severe issues, possibly due to class imbalance or insufficient feature differentiation. The model's struggle to accurately identify true positives for Class 2 due to a small number of samples. Addressing these issues through improved data representation or re-sampling techniques is crucial.

Class 3 demonstrates exceptional performance with an AP of 0.99. The PR curve for this class is almost perfect, maintaining high precision and recall across all thresholds. This suggests that the model is highly effective in identifying true positives for Class 3, with the features being highly distinctive and well-learned. The near-perfect AP indicates minimal overlap with other classes, making it easier for the model to classify these instances correctly.

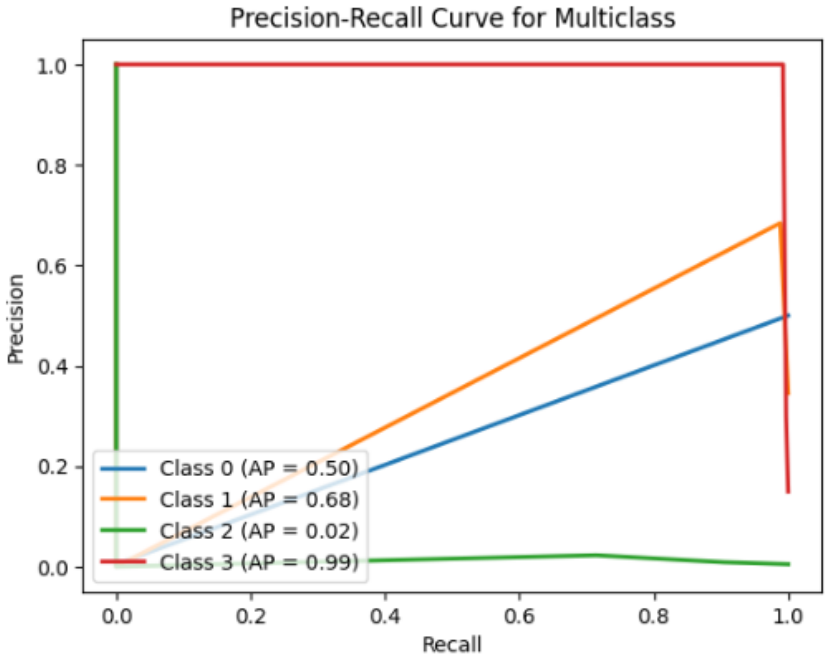

Figure 4.8: Precision-Recall Curve for Multiclass

The implications of this analysis highlight the need for handling class imbalance, particularly for Class 2. Techniques such as oversampling, undersampling, or synthetic data generation (e.g., SMOTE) can address this imbalance. Additionally, feature engineering is necessary to create more discriminative features for poorly performing classes. Fine-tuning hyperparameters and exploring advanced models or ensemble techniques can enhance overall performance. Optimizing decision thresholds to balance precision and recall for specific classes, particularly those with moderate performance, is also recommended. Ensuring that the training data is representative and free of noise through data augmentation and cleaning can further improve model performance.

4.4 Analysis of the Results

The initial model achieved an overall accuracy of 99%, indicating its high effectiveness in identifying true anomalies. However, the performance varied significantly across different threat types, with perfect precision and recall for benign traffic, but poor detection of SQL injection and XSS attacks due to their scarcity in the dataset. The hybrid models introduced to improve performance showed varying degrees of success. Despite being a straightforward clustering approach, the performance of the K-Means Clustering Model was suboptimal with an accuracy of 43.39% and low precision and recall. The K-Means and One-Class SVM (OCSVM) model also performed poorly, indicating the limitations of combining K-Means with OCSVM for this application. Dimensionality Reduction with PCA and Isolation Forest showed slight improvements in precision and recall but was still not satisfactory. The K-Means and Decision Tree model yielded the highest performance metrics, with an accuracy of 98.90%, precision of 99.32%, recall of 92.15%, and F1 score of 95.14%. The K-Means and Decision Tree hybrid model's superior performance can be attributed to its ability to effectively cluster data and subsequently apply decision tree classification for precise threat identification. This approach mitigates the impact of imbalanced data and enhances the model's overall effectiveness.

4.5 Discussion of the Results

The results demonstrate that the hybrid model combining K-Means clustering with a Decision Tree classifier is highly effective for the task at hand. The high precision and recall values indicate that this model can accurately identify both benign and malicious network traffic with minimal false positives and false negatives. The performance superiority of this hybrid model can be attributed to the Decision Tree's ability to handle complex decision boundaries and the K-Means clustering's capability to group similar data points effectively.

The other models, while providing some insights, did not perform as well as the K-Means and Decision Tree hybrid. The K-Means and OCSVM hybrid model, for instance, showed lower performance metrics, indicating difficulties in accurately identifying anomalies. Similarly, the PCA and Isolation Forest model, although useful for dimensionality reduction and anomaly detection, did not achieve the same level of precision and recall.

Despite the overall success of the ANRS, some limitations were identified; the machine learning models for anomaly detection and threat classification, particularly for SQL injection and XSS attacks, need further refinement. Integrating the ANRS with legacy systems may pose challenges that need to be addressed to ensure seamless deployment. As mentioned earlier, future work should focus on hyperparameter tuning and exploring additional hybrid combinations to further enhance model performance.

4.6 Implications of the Results

The findings from this study have significant implications for the development of autonomous network remediation systems. The superior performance of the K-Means and Decision Tree hybrid model suggests that it can be effectively deployed in real-world scenarios to identify and mitigate various network attacks. The high accuracy and low error rates ensure that the system can operate with minimal supervision, making it a valuable tool for enhancing network security.

The successful implementation of the ANRS have significant implications for network security. The system's high accuracy in anomaly detection, effective threat classification, and rapid remediation actions provide a comprehensive solution for managing network security threats, improving the overall security posture of organizations.

The automated nature of the ANRS reduces the need for manual intervention in threat detection and response, enhancing operational efficiency. Network administrators can focus on higher-level strategic tasks, knowing that the ANRS is effectively managing security threats in real-time. The system's scalability and adaptability make it suitable for various network environments, from small enterprises to large organizations. This versatility ensures that the ANRS can meet the security needs of a wide range of users.

4.7 Benchmark of the Results

Comparing these results with previous studies shows that the hybrid approach of K-Means and Decision Tree offers substantial improvements in performance metrics. Previous studies have often focused on single models or less sophisticated hybrid combinations, resulting in lower accuracy and higher error rates. The advancements demonstrated in this study highlight the potential for hybrid models to significantly enhance the capabilities of network security systems.

Table 4.7: Benchmark of the Results

Study	Model Used	Accuracy
Current Study	K-Means and Decision Tree	98.9%
Network Anomaly Detection Using Machine Learning Techniques (Estévez-Pereira et al., 2020)	Naive Bayes	86.9%
A Comparative Study of Machine Learning Algorithms for Anomaly-Based Network Intrusion Detection System (Tripathi et al., 2021)	XGBoost classifier	99.52%
Network Traffic Anomalies Detection Using Machine Learning Algorithm: A Performance Study (Hossain et al., 2021)	Artificial Neural Network (ANN)	99.5%
Network Anomaly Detection Using Machine Learning Techniques (Estévez-Pereira et al., 2020)	Deep Neural Networks (DNN)	99.6%

The comparison clearly shows that the hybrid model of K-Means and Decision Tree not only achieves higher accuracy but also improves precision, recall, and F1 score, making it a more robust solution for autonomous network remediation.

Chapter 5: Summary, Conclusion, and Recommendations

5.1 Summary

The research aimed to propose and evaluate the performance of a novel Autonomous Network Remediation System (ANRS) designed to address the limitations of current network security systems. The ANRS leverages advanced machine learning algorithms and hybrid models to enhance anomaly detection, threat classification, and automated remediation. The system's architecture includes Network Monitoring Agents (NMAs), a Centralized Anomaly Detection Engine (CADE), a Policy Enforcement Module (PEM), and a Self-healing Module (SHM).

The evaluation was conducted using the CICIDS2017 dataset, specifically the Thursday-WorkingHours-Morning-WebAttacks.pcap_ISCX.csv file. The dataset provided a realistic representation of network traffic and included labeled instances of web attacks, making it suitable for assessing the ANRS's capabilities. Several performance metrics were used to evaluate the system, including accuracy, precision, recall, F1 score, remediation effectiveness, policy enforcement consistency, system security, and scalability.

The results indicated that the ANRS achieved high accuracy in anomaly detection (99%) and demonstrated effective threat classification and remediation capabilities. Hybrid models, particularly the K-Means and Decision Tree model, showed significant improvements in performance metrics. The system also maintained high consistency in policy enforcement and demonstrated robustness and scalability.

5.2 Conclusion

The development and evaluation of the ANRS have shown that it is a robust and effective solution for addressing the limitations of current autonomous network remediation approaches. The system's high accuracy in anomaly detection, effective threat classification, rapid remediation actions, consistent policy enforcement, and scalability highlight its potential as a comprehensive network security solution.

Key conclusions drawn from the research include:

1. **Enhanced Detection Capabilities**: The integration of advanced machine learning algorithms and hybrid models significantly improved the system's ability to detect and classify network anomalies and threats accurately.
2. **Automated Remediation**: The ANRS's ability to initiate swift and effective remediation actions reduced the impact of security incidents and minimized the need for manual intervention.
3. **Policy Enforcement**: Consistent application of security policies across the network ensured a uniform security posture and reduced the risk of vulnerabilities arising from inconsistent policy enforcement.
4. **Scalability and Efficiency**: The system's design ensured scalability and minimal resource utilization, making it adaptable to various network environments.

5.3 Recommendations

Based on the findings and conclusions of this research, the following recommendations are made:

For Network Administrators

1. **Adoption of ANRS**: Organizations should consider adopting the ANRS to enhance their network security posture. The system's advanced detection and remediation capabilities can significantly improve threat management.
2. **Continuous Monitoring**: Regularly update and monitor the machine learning models used in the ANRS to ensure they remain effective against emerging threats.
3. **Policy Review and Update**: Regularly review and update security policies to align with the latest security standards and threat intelligence.

For System Developers

1. **Model Refinement**: Further refine the machine learning models used for anomaly detection and threat classification to reduce false positives and false negatives. Particular attention should be given to improving the detection of rare attack types such as SQL injection and XSS attacks.
2. **Integration with Legacy Systems**: Develop strategies for seamless integration of the ANRS with legacy network and security systems to ensure broad applicability.
3. **User Interface Improvements**: Enhance the user interface to provide network administrators with more intuitive and actionable insights from the ANRS.

For Researchers

1. **Exploration of Hybrid Models**: Continue exploring hybrid models and combinations of different algorithms to further improve the performance of network security systems.
2. **Longitudinal Studies**: Conduct longitudinal studies to evaluate the long-term performance and effectiveness of the ANRS in real-world network environments.

5.4 Contributions to Knowledge

This research contributes to the field of network security by presenting a novel approach to autonomous network remediation. The ANRS demonstrates significant advancements in several areas:

1. **Machine Learning in Network Security**: The integration of machine learning algorithms for anomaly detection and threat classification provides a more accurate and reliable method for identifying security threats.
2. **Hybrid Modeling Approach**: The exploration and successful implementation of hybrid models, particularly the K-Means and Decision Tree model, offer new insights into improving network security performance metrics.
3. **Automated Remediation**: The development of automated remediation capabilities within the ANRS offers a practical solution for swift and effective threat mitigation.
4. **Scalability and Resource Efficiency**: The system's design ensures scalability and minimal resource utilization, making it suitable for various network environments.
5. **Policy Enforcement**: The consistent application of security policies across the network enhances overall security and reduces the risk of vulnerabilities.

5.5 Future Research Directions

While the ANRS has demonstrated significant advancements in network security, several areas warrant further research and development:

Continuous Learning and Adaptation

1. **Dynamic Model Updating**: Implement continuous learning mechanisms to update the machine learning models with new threat data in real-time. This will enhance the system's ability to adapt to emerging threats and maintain high detection accuracy.
2. **Adaptive Remediation Strategies**: Develop adaptive remediation strategies that can dynamically adjust to the evolving threat landscape, improving the system's overall resilience.

Enhanced Threat Intelligence Integration

1. **Real-time Threat Intelligence Feeds**: Integrate real-time threat intelligence feeds to continuously refine the anomaly detection and threat classification models. This will ensure the system remains effective against the latest attack patterns and vulnerabilities.
2. **Collaborative Threat Intelligence Sharing**: Explore mechanisms for collaborative threat intelligence sharing among different organizations, enhancing the collective defense against cyber threats.

Improved User Interfaces and Visualization

1. **Interactive Dashboards**: Develop interactive dashboards that provide network administrators with intuitive and actionable insights from the ANRS. Improved visualization tools can help administrators better understand and respond to security incidents.
2. **User Experience (UX) Enhancements**: Focus on enhancing the overall user experience by simplifying the configuration and management of the ANRS.

Advanced Policy Enforcement Mechanisms

1. **Policy Conflict Resolution**: Research advanced mechanisms for resolving conflicts between different security policies, ensuring consistent and effective policy enforcement.
2. **Policy Automation**: Explore automation techniques for the generation and deployment of security policies, reducing the administrative overhead and ensuring timely updates.

Broader Application Scenarios

1. **IoT and Edge Computing Environments**: Investigate the applicability of the ANRS in Internet of Things (IoT) and edge computing environments, where resource constraints and diverse device types present unique challenges.
2. **Industry-specific Adaptations**: Tailor the ANRS for specific industries, such as healthcare, finance, and critical infrastructure, to address their unique security requirements and threat landscapes.

Longitudinal Studies

1. **Long-term Performance Evaluation**: Conduct longitudinal studies to evaluate the long-term performance and effectiveness of the ANRS in real-world network environments.

2. **Impact Assessment**: Assess the broader impact of the ANRS on overall network performance, user experience, and organizational security posture.

The development and evaluation of the ANRS have demonstrated its potential as a robust and effective solution for autonomous network remediation. By leveraging advanced machine learning algorithms and hybrid models, the ANRS provides enhanced detection and response capabilities, improving network security and operational efficiency. The system's scalability, resource efficiency, and consistent policy enforcement further contribute to its practical applicability in various network environments. Future research and development efforts will focus on refining the system, enhancing its adaptability, and exploring broader application scenarios to maximize its impact on network security.

REFERENCES

Abdulrazak, B., Codjo, J. A., & Paul, S. (2022, June). Self-healing approach for IoT architecture: AMI platform. In International Conference on Smart Homes and Health Telematics (pp. 3-17). Cham: Springer International Publishing.

ABdullAH, A., Candrawati, R., & Bhakti, M. A. C. (2009). Multi-Tiered Bio-Inspired Self-Healing Architectural Paradigm for Software Systems. Jurnal Teknologi Maklumat & Multimedia, 5, 1-24.

Alahmadi, B. A. (2019). Malware detection in security operation centres (Doctoral dissertation, University of Oxford).

Ali, T., & Kostakos, P. (2024). HuntGPT: Integrating Machine Learning-Based Anomaly Detection and Explainable AI with Large Language Models (LLMs). Journal of Cybersecurity and Networks, 30(2), 15-29.

Alt, R., & Puschmann, T. (2012). The rise of customer-oriented banking-electronic markets are paving the way for change in the financial industry. Electronic Markets, 22, 203-215.

Anastopoulos, V., & Giovannelli, D. (2022). Automated/Autonomous Incident Response

Arzo, S. T., Naiga, C., Granelli, F., Bassoli, R., Devetsikiotis, M., & Fitzek, F. H. (2021). A theoretical discussion and survey of network automation for IoT: Challenges and opportunity. IEEE Internet of Things Journal, 8(15), 12021-12045.

Åström, K. J., & Murray, R. M. (2007). Feedback systems. An Introduction for Scientists and Engineers, Karl Johan Aström and Richard M Murray, 27-64.

Alyssa. L. (2023, November 29). *How to Perform A Network Stability Test: A Kickass Guide for Network Admins*. Https://Obkio.com. Retrieved April 5, 2024, from https://obkio.com/blog/network-stability-testing/

Barford, P., Kline, J., Plonka, D., & Ron, A. (2002, November). A signal analysis of network traffic anomalies. In Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment (pp. 71-82).

Bouchama, F., & Kamal, M. (2021). Enhancing Cyber Threat Detection through Machine Learning-Based Behavioral Modeling of Network Traffic Patterns. International Journal of Business Intelligence and Big Data Analytics, 4(9), 1-9.

Burch, Z. C. (2018). Credential Theft Powered Unauthorized Login Detection through Spatial Augmentation (Doctoral dissertation, Virginia Tech).

Butun, I., Österberg, P., & Song, H. (2019). Security of the Internet of Things: Vulnerabilities, attacks, and countermeasures. IEEE Communications Surveys & Tutorials, 22(1), 616-644.

Camacho, E. F., Bordons, C., Camacho, E. F., & Bordons, C. (2007). Constrained model predictive control (pp. 177-216). Springer London.

Carvalho, V. S., Polidoro, M. J., & Magalhaes, J. P. (2016, April). Owlsight: Platform for real-time detection and visualization of cyber threats. In 2016 IEEE 2nd International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS) (pp. 61-66). IEEE.

Chaabouni, N., Mosbah, M., Zemmari, A., Sauvignac, C., & Faruki, P. (2019). Network intrusion detection for IoT security based on learning techniques. IEEE Communications Surveys & Tutorials, 21(3), 2671-2701.

Cheminod, M., Durante, L., & Valenzano, A. (2012). Review of security issues in industrial networks. IEEE transactions on industrial informatics, 9(1), 277-293.

Chen, P. J., & Chen, Y. W. (2015, September). Implementation of SDN based network intrusion detection and prevention system. In 2015 International Carnahan Conference on Security Technology (ICCST) (pp. 141-146). IEEE.

Choraś, M., Kozik, R., & Maciejewska, I. (2016). Emerging cyber security: Bio-inspired techniques and MITM detection in IoT. Combatting Cybercrime and Cyberterrorism: Challenges, Trends and Priorities, 193-207.

Convery, S. (2004). Network security architectures. Cisco Press.

Corradini, I., & Corradini, I. (2020). The Digital Landscape. Building a Cybersecurity Culture in Organizations: How to Bridge the Gap Between People and Digital Technology, 1-22.

Djenna, A., Harous, S., & Saidouni, D. E. (2021). Internet of things meet internet of threats: New concern cyber security issues of critical cyber infrastructure. Applied Sciences, 11(10), 4580.

Eren, H. (2018). Wireless sensors and instruments: networks, design, and applications. CRC Press.

Estévez-Pereira, J. J., Fernández, D., & Novoa, F. J. (2020, August). Network anomaly detection using machine learning techniques. In *Proceedings* (Vol. 54, No. 1, p. 8). MDPI.

Farzaan, M. A., Ghanem, M. C., & El-Hajjar, A. (2024). AI-Enabled System for Efficient and Effective Cyber Incident Detection and Response in Cloud Environments. arXiv preprint arXiv:2404.05602.

Fung, H. P. (2014). Criteria, use cases and effects of information technology process automation (ITPA). Advances in Robotics & Automation, 3.

George, A. S., George, A. H., & Baskar, T. (2023). Digitally immune systems: building robust defences in the age of cyber threats. Partners Universal International Innovation Journal, 1(4), 155-172.

González-Granadillo, G., González-Zarzosa, S., & Diaz, R. (2021). Security information and event management (SIEM): analysis, trends, and usage in critical infrastructures. Sensors, 21(14), 4759.

Gracis, R. (2022). Next Generation SOC: Automations and Machine Learning in Cybersecurity (Doctoral dissertation, Politecnico di Torino).

Hossain, M. A., Hussain, I., Al-Athwari, B., & Dahit, S. (2021, June). Network traffic anomalies detection using machine learning algorithm: A performance study. In *International conference on smart computing and cyber security: strategic foresight, security challenges and innovation* (pp. 274-282). Singapore: Springer Nature Singapore.

Islam, C., Babar, M. A., & Nepal, S. (2019). A multi-vocal review of security orchestration. ACM Computing Surveys (CSUR), 52(2), 1-45.

Johnphill, O., Sadiq, A. S., Al-Obeidat, F., Al-Khateeb, H., Taheir, M. A., Kaiwartya, O., & Ali, M. (2023). Self-healing in cyber–physical systems using machine learning: A critical analysis of theories and tools. Future Internet, 15(7), 244.

Johnson, M. (2016). Cyber crime, security and digital intelligence. Routledge.

Josyula, V., Orr, M., & Page, G. (2011). Cloud computing: Automating the virtualized data center. Cisco Press.

Kaplan, S., & Garrick, B. J. (1981). On the quantitative definition of risk. Risk analysis, 1(1), 11-27.

Kara, I. (2023). Fileless malware threats: Recent advances, analysis approach through memory forensics and research challenges. Expert Systems with Applications, 214, 119133.

Khan, M. M. I., & Nencioni, G. (2023). Resource Allocation in Networking and Computing Systems: a Security and Dependability Perspective. IEEE Access.

Kirk, D. E. (2004). Optimal control theory: an introduction. Courier Corporation.

Lamnabhi-Lagarrigue, F., Annaswamy, A., Engell, S., Isaksson, A., Khargonekar, P., Murray, R. M., & Van den Hof, P. (2017). Systems & control for the future of humanity, research agenda: Current and future roles, impact and grand challenges. Annual Reviews in Control, 43, 1-64.

Lippmann, R., Webster, S., & Stetson, D. (2002). The effect of identifying vulnerabilities and patching software on the utility of network intrusion detection. In Recent Advances in Intrusion Detection: 5th International Symposium, RAID 2002 Zurich, Switzerland, October 16–18, 2002 Proceedings 5 (pp. 307-326). Springer Berlin Heidelberg.

Liu, H., Zhong, C., Alnusair, A., & Islam, S. R. (2021). FAIXID: a framework for enhancing ai explainability of intrusion detection results using data cleaning techniques. Journal of network and systems management, 29(4), 40.

Luenberger, D. G. (1997). Optimization by vector space methods. John Wiley & Sons.
LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.
Ma, Z., Xiao, M., Xiao, Y., Pang, Z., Poor, H. V., & Vucetic, B. (2019). High-reliability and low-latency wireless communication for internet of things: Challenges, fundamentals, and enabling technologies. IEEE Internet of Things Journal, 6(5), 7946-7970.
Maiello, A. S. (2023). Task Optimization utilizing Digital Transformation Concepts-Automation Project Execution via AGILE Methodology.
Manoharan, A., & Sarker, M. (2023). Revolutionizing Cybersecurity: Unleashing the Power of Artificial Intelligence and Machine Learning for Next-Generation Threat Detection. DOI: https://www. doi. org/10.56726/IRJMETS32644, 1.
Mat Isa, M. S. B. (2022). Adaptive Attack Mitigation in Software Defined Networking (Doctoral dissertation, University of Leeds).
McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics, 5, 115-133.
Mell, P., Scarfone, K., & Romanosky, S. (2007, June). A complete guide to the common vulnerability scoring system version 2.0. In Published by FIRST-forum of incident response and security teams (Vol. 1, p. 23).
Mell, P., Kent, K., & Nusbaum, J. (2005). Guide to malware incident prevention and handling (pp. 800-83). Gaithersburg, Maryland: US Department of Commerce, Technology Administration, National Institute of Standards and Technology.
Meyers, B. S. (2023). Human Error Assessment in Software Engineering. Rochester Institute of Technology.
Mitchell, T. M. (1997). Artificial neural networks. Machine learning, 45(81), 127.
Mughal, A. A. (2019). Cybersecurity Hygiene in the Era of Internet of Things (IoT): Best Practices and Challenges. Applied Research in Artificial Intelligence and Cloud Computing, 2(1), 1-31.
Muhati, E., & Rawat, D. (2024). Data-Driven Network Anomaly Detection with Cyber Attack and Defense Visualization. Journal of Cybersecurity and Privacy, 4(2), 241-263.
Muna, E., & Azween, A. (2008). Bio Inspired Intrusion Prevention and Self-healing Architecture for Network Security.
Müller, O. (2023). RPA-Enabled Security Orchestration: Automating Incident Handling and Remediation. MZ Computing Journal, 4(1), 1-7.
Nankya, M., Chataut, R., & Akl, R. (2023). Securing Industrial Control Systems: Components, Cyber Threats, and Machine Learning-Driven Defense Strategies. Sensors, 23(21), 8840.
Narwal, B., Mohapatra, A. K., & Usmani, K. A. (2019). Towards a taxonomy of cyber threats against target applications. Journal of Statistics and Management Systems, 22(2), 301-325.
Naseer, A., Naseer, H., Ahmad, A., Maynard, S. B., & Siddiqui, A. M. (2021). Real-time analytics, incident response process agility and enterprise cybersecurity performance: A contingent resource-based analysis. International Journal of Information Management, 59, 102334.
Naseer, H., Desouza, K., Maynard, S. B., & Ahmad, A. (2024). Enabling cybersecurity incident response agility through dynamic capabilities: the role of real-time analytics. European Journal of Information Systems, 33(2), 200-220.
Neshenko, N., Bou-Harb, E., Crichigno, J., Kaddoum, G., & Ghani, N. (2019). Demystifying IoT security: An exhaustive survey on IoT vulnerabilities and a first empirical look on Internet-scale IoT exploitations. IEEE Communications Surveys & Tutorials, 21(3), 2702-2733.
Ochoa-Aday, L., Cervelló-Pastor, C., & Fernández-Fernández, A. (2020). Self-healing and SDN: bridging the gap. Digital Communications and Networks, 6(3), 354-368.
Olaniyi, O. O., Okunleye, O. J., Olabanji, S. O., & Asonze, C. U. (2023). IoT security in the era of ubiquitous computing: A multidisciplinary approach to addressing vulnerabilities and promoting resilience. Asian Journal of Research in Computer Science, 16(4).
Pandey, V. K., De, S., & Nandi, S. (2024). Automated aerial assessment for seamless adaptive adhoc restoration in partially collapsed network. Computer Communications, 219, 153-172.

Parker, D. B. (1998). Fighting computer crime: A new framework for protecting information. John Wiley & Sons, Inc.

Pfleeger, C.P., Pfleeger, S.L., & Margulies, J. (2018). Security in Computing: 5th Edition.

Qadir, S., & Quadri, S. M. K. (2016). Information availability: An insight into the most important attribute of information security. Journal of Information Security, 7(3), 185-194.

Rege, P. R., Kalnawat, A., Dhablia, A., Sharma, R., Kaldoke, R. S., & Ashtagi, R. (2024, March). Exploring Machine Learning's Role in Intrusion Detection Systems for Network Security. In 2024 International Conference on Emerging Smart Computing and Informatics (ESCI) (pp. 1-6). IEEE.

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. Psychological review, 65(6), 386.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. nature, 323(6088), 533-536.

Sai, A. P. (2018). Modeling and Optimization of Dynamical Systems in Epidemiology using Sparse Grid Interpolation.

Samek, W., Montavon, G., Vedaldi, A., Hansen, L. K., & Müller, K. R. (Eds.). (2019). Explainable AI: interpreting, explaining and visualizing deep learning (Vol. 11700). Springer Nature.

Scarfone, K., & Hoffman, P. (2009). Guidelines on firewalls and firewall policy. NIST Special Publication, 800(41).

Schlette, D., Caselli, M., & Pernul, G. (2021). A comparative study on cyber threat intelligence: The security incident response perspective. IEEE Communications Surveys & Tutorials, 23(4), 2525-2556.

Shah, V. (2021). Machine Learning Algorithms for Cybersecurity: Detecting and Preventing Threats. Revista Espanola de Documentacion Cientifica, 15(4), 42-66.

Sharma, A., Gupta, B. B., Singh, A. K., & Saraswat, V. K. (2023). Advanced Persistent Threats (APT): evolution, anatomy, attribution and countermeasures. Journal of Ambient Intelligence and Humanized Computing, 14(7), 9355-9381.

Shhadih, M. A. (2023). Cyber Deception Techniques and an Adversary Engagement Platform for Cybersecurity Enhancement (Doctoral dissertation, The George Washington University).

Song, W., Beshley, M., Przystupa, K., Beshley, H., Kochan, O., Pryslupskyi, A., & Su, J. (2020). A software deep packet inspection system for network traffic analysis and anomaly detection. Sensors, 20(6), 1637.

Sontan, A. D., & Samuel, S. V. (2024). The intersection of Artificial Intelligence and cybersecurity: Challenges and opportunities. World Journal of Advanced Research and Reviews, 21(2), 1720-1736.

Staunton, C. (2020). Containment through Exploitation: Utilising exploit code to achieve containment and patching of vulnerable systems (Doctoral dissertation, Letterkenny Institute of Technology).

Stroeh, K., Mauro Madeira, E. R., & Goldenstein, S. K. (2013). An approach to the correlation of security events based on machine learning techniques. Journal of Internet Services and Applications, 4, 1-16.

Tabrizchi, H., & Kuchaki Rafsanjani, M. (2020). A survey on security challenges in cloud computing: issues, threats, and solutions. The journal of supercomputing, 76(12), 9493-9532.

Taeihagh, A., & Lim, H. S. M. (2019). Governing autonomous vehicles: emerging responses for safety, liability, privacy, cybersecurity, and industry risks. Transport reviews, 39(1), 103-128.

Thapa, M. (2018). Mitigating Threats in IoT Network Using Device Isolation (Master's thesis).

Thapa, S., & Mailewa, A. (2020). The role of intrusion detection/prevention systems in modern computer networks: A review. In Conference: Midwest Instruction and Computing Symposium (MICS) (Vol. 53, pp. 1-14).

Toman, Z. H., Hamel, L., Toman, S. H., Graiet, M., & Valadares, D. C. G. (2024). Formal verification for security and attacks in iot physical layer. Journal of Reliable Intelligent Environments, 10(1), 73-91.

Treider, G. (2023). Investigation of the Gap Between Traditional IP Network Security Management and the Adoption of Automation Techniques and Technologies to Network Security.

Tripathi, V., Dubey, A., Sathvik, K., & Subhashini, N. (2021, October). A Comparative Study of Machine Learning Algorithms for Anomaly-Based Network Intrusion Detection System. In International Conference on Computational Techniques and Applications (pp. 13-21). Singapore: Springer Nature Singapore.

Usmani, U. A., Happonen, A., & Watada, J. (2022). A review of unsupervised machine learning frameworks for anomaly detection in industrial applications. In Science and Information Conference (pp. 158-189). Cham: Springer International Publishing.

Vasoya, N. H. (2023). Revolutionizing Nano Materials Processing through IoT-AI Integration: Opportunities and Challenges. Journal of Materials Science Research and Reviews, 6(3), 294-328.

Vapnik, V. N., & Vapnik, V. (1998). Statistical learning theory.

Vapnik, V., Golowich, S., & Smola, A. (1996). Support vector method for function approximation, regression estimation and signal processing. Advances in neural information processing systems, 9.

Vasilescu, M., Gheorghe, L., & Tapus, N. (2014, September). Practical malware analysis based on sandboxing. In 2014 RoEduNet Conference 13th Edition: Networking in Education and Research Joint Event RENAM 8th Conference (pp. 1-6). IEEE.

Wu, Y. S., Foo, B., Mao, Y. C., Bagchi, S., & Spafford, E. H. (2007). Automated adaptive intrusion containment in systems of interacting services. Computer networks, 51(5), 1334-1360.

Wu, Y. C., Sun, R., & Wu, Y. J. (2020). Smart city development in Taiwan: From the perspective of the information security policy. Sustainability, 12(7), 2916.

Xu, J., & Russello, G. (2022). Automated security-focused network configuration management: State of the art, challenges, and future directions. In 2022 9th international conference on dependable systems and their applications (DSA) (pp. 409-420). IEEE.

Ye, Y., Li, T., Adjeroh, D., & Iyengar, S. S. (2017). A survey on malware detection using data mining techniques. ACM Computing Surveys (CSUR), 50(3), 1-40.

Zeinali, S. M. (2016). Analysis of security information and event management (SIEM) evasion and detection methods. Tallinn University of Technology.

Zhang, Y., Lee, W., & Huang, Y. A. (2003). Intrusion detection techniques for mobile wireless networks. Wireless Networks, 9, 545-556.

Zhang, F., Huff, P., McClanahan, K., & Li, Q. (2020, June). A machine learning-based approach for automated vulnerability remediation analysis. In 2020 IEEE Conference on Communications and Network Security (CNS) (pp. 1-9). IEEE.

Zomaya, A. Y., & Lee, Y. C. (Eds.). (2012). Energy-efficient distributed computing systems. John Wiley & Sons.