T.C.
ISTANBUL GEDİK UNIVERSITY
INSTITUTE OF GRADUATE STUDIES

A NEW COMPUTER SECURITY TECHNIQUE BASED DATA SCIENCE TECHNIQUES

MASTER THESIS

Mohammed Hazim Mutar MUTAR

Engineering Management Department

Engineering Management Master in English Program

NOVEMBER 2023
İSTANBUL

T.C.
ISTANBUL GEDİK UNIVERSITY
INSTITUTE OF GRADUATE STUDIES

A NEW COMPUTER SECURITY TECHNIQUE BASED DATA SCIENCE TECHNIQUES

MASTER THESIS

Mohammed Hazim Mutar MUTAR
(201281013)

Engineering Management Department

Engineering Management Master in English Program

Thesis Advisor: Assist. Prof. Dr. Üyesi Ahmet SARIKAHYA

Istanbul 2023

T.C.
İSTANBUL GEDİK ÜNİVERSİTESİ
Lisansüstü Eğitim Enstitüsü Müdürlüğü

Jüri Tez Onay Formu

28.11.2023

LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ MÜDÜRLÜĞÜ

Bu çalışma 28.11 2023 tarihinde aşağıdaki jüri tarafından Engineering Management Department, Engineering Management (Tezli Yüksek Lisans) Programı Yüksek Lisans Tezi olarak kabul edilmiştir.

TEZ JÜRİSİ

Dr. Öğr. Üyesi Ahmet SARIKAHYA
Danışman
İstanbul Gedik Üniversitesi

Dr. Öğr. Üyesi Tuğbay Burçin GÜMÜŞ
Üye (İmza)
İstanbul Gedik Üniversitesi

Doç. Dr. Mehmet Serdar GÜZEL
Üye (İmza)
Ege Üniversitesi

DECLARATION

I, Mohammed Hazim MUTAR, hereby certify that this thesis entitled “A New Computer Security Technique Based Data Science Techniques" is my original thesis for the award of Master's Degree in Engineering Management at the Faculty of Engineering Management. I further certify that this thesis or any part thereof has not been submitted and presented for any other degree or research thesis at any other university or institution. (28.11.2023)

Mohammed Hazim Mutar MUTAR

DEDICATION

I would like to begin by expressing my thankfulness to Allah (God) for bestowed upon me the knowledge, talents, and opportunities that were essential to carry out and finish this study. I would want to express my heartfelt appreciation to both of my parents. Their love is the driving force behind all of the favorable occurrences and adventures that have materialized in my life over the last several years. I would want to take use of this chance to show my appreciation to my sister and brother for all they have done for me, and I would like to do so by making use of this time.

TABLE OF CONTENT

Page

ABBREVIATIONS

IDS : Intrusion Detection system
RF : Random Forest
DT : Decision Tree
BotNet : Robot Network
WSN : Wireless Sensor Network
IOT : Internet of Things
ML : Machine Learning

LIST OF TABLES

Page

LIST OF FIGURES

Page

A NEW COMPUTER SECURITY TECHNIQUE BASED DATA SCIENCE TECHNIQUES

ABSTRACT

Due to the proliferation and dependence on digital technology, cyber security has emerged as a pressing issue in today's world. The dangers of unauthorized access, data breaches, and network invasions are growing as more people and businesses take use of the perks of networked technology. Intrusion detection is essential in reducing these dangers since it allows for instantaneous response to security breaches. Network traffic, system logs, and other data sources are monitored and analyzed by intrusion detection systems (IDS) in search of indicators of hacking or other malicious activity. In the event of a security breach or other harmful activity, these technologies can detect it and notify the appropriate authorities. The success of an IDS, however, hinges largely on the careful selection of features. The success or failure of an intrusion detection system relies heavily on the quality of the dataset used for detection, and this is where "feature selection" comes in. The objective is to minimize the data's dimensionality while maintaining or improving the features' discriminating ability. High-dimensional datasets, irrelevant or redundant features, and the curse of dimensionality are only some of the obstacles that feature selection in intrusion detection seeks to address. Manual feature selection based on expert knowledge or trial and error are examples of traditional ways; nevertheless, these methods are time-consuming, subjective, and sometimes unproductive.

Keywords: IDS, RF, FE, AI, ML

YENİ BİR BİLGİSAYAR GÜVENLİK TEKNİĞİ TABANLI VERİ BİLİMİ TEKNİKLERİ

ÖZET

Dijital teknolojinin yaygınlaşması ve ona bağımlı hale gelmesi nedeniyle siber güvenlik günümüz dünyasında acil bir konu olarak ortaya çıkmıştır. Yetkisiz erişim, veri ihlalleri ve ağ istilası tehlikeleri, daha fazla kişi ve işletme ağ bağlantılı teknolojinin avantajlarından faydalandıkça artıyor. İzinsiz giriş tespiti, güvenlik ihlallerine anında yanıt verilmesine olanak sağladığı için bu tehlikelerin azaltılması açısından önemlidir. Ağ trafiği, sistem günlükleri ve diğer veri kaynakları, bilgisayar korsanlığı veya diğer kötü amaçlı etkinlik göstergelerinin araştırılması amacıyla izinsiz giriş tespit sistemleri (IDS) tarafından izlenir ve analiz edilir. Bir güvenlik ihlali veya başka bir zararlı faaliyet durumunda, bu teknolojiler bunu tespit edebilir ve ilgili makamlara bildirimde bulunabilir. Ancak bir IDS'nin başarısı büyük ölçüde özelliklerin dikkatli seçimine bağlıdır. Bir izinsiz giriş tespit sisteminin başarısı veya başarısızlığı büyük ölçüde tespit için kullanılan veri kümesinin kalitesine bağlıdır ve "özellik seçimi" burada devreye girer. Amaç, özelliklerin ayırt etme yeteneğini korurken veya geliştirirken verinin boyutsallığını en aza indirmektir. Yüksek boyutlu veri kümeleri, ilgisiz veya gereksiz özellikler ve boyutluluğun laneti, izinsiz giriş tespitinde özellik seçiminin ele almaya çalıştığı engellerden yalnızca birkaçıdır. Uzman bilgisine veya deneme yanılmaya dayalı manuel özellik seçimi, geleneksel yolların örnekleridir; yine de bu yöntemler zaman alıcıdır, özneldir ve bazen verimsizdir.

Anahtar Kelimeler: IDS, RF, FE, AI, ML

1. INTRODUCTION

1.1 Backround

When it comes to the topic of cybersecurity, two of the most critical steps that can be performed to secure sensitive data and ensure that systems continue to function correctly are the detection of potential computer vulnerabilities and the avoidance of those vulnerabilities wherever possible. When it comes to identifying new and developing dangers, traditional methodologies that are based on signatures have their limitations, which is why the creation of more advanced procedures is important [1]. The application of data science techniques, such as behavior analysis, has emerged as a potentially effective tool for spotting possible threats to a company's security. One example of such a strategy is the deployment of data science methods. This extensive background section looks at the concept of behavior analysis, as well as the application of Random Forest (RF) methodologies and Recursive Feature Elimination (RFE), in the context of the detection of potential threats posed by computers. Cybersecurity has become an increasingly critical aspect of modern society due to the rapid growth and reliance on digital technologies. As organizations and individuals embrace the benefits of interconnected systems, they also face escalating risks from malicious activities, such as unauthorized access, data breaches, and network intrusions. Intrusion detection plays a crucial role in mitigating these risks by identifying and responding to potential threats in real-time. Intrusion detection systems (IDS) are designed to monitor and analyze network traffic, system logs, and other data sources to detect signs of unauthorized access or suspicious activities. These systems act as a frontline defense mechanism, capable of identifying and alerting security personnel about potential security breaches or malicious activities. However, the effectiveness of an IDS heavily depends on the selection of appropriate features. Feature selection is the process of identifying the most relevant and informative attributes or variables from the available dataset, which can significantly impact the accuracy and efficiency of intrusion detection. The goal is to reduce the dimensionality of the data while preserving or enhancing

the discriminatory power of the features. Feature selection in intrusion detection aims to overcome challenges such as high-dimensional datasets, irrelevant or redundant attributes, and the curse of dimensionality [2]. Traditional approaches to feature selection include manual selection based on expert knowledge or trial and error, but these methods are time-consuming, subjective, and often ineffective. To address these limitations, researchers have explored various automated feature selection techniques, including filter methods, wrapper methods, and embedded methods. Filter methods assess the relevance of features based on statistical or information-theoretic measures, independent of a specific classification algorithm. Wrapper methods employ a specific classifier to evaluate the subsets of features, optimizing a performance metric such as accuracy or precision. Embedded methods incorporate feature selection directly into the learning algorithm, using regularization techniques or feature importance rankings provided by ensemble models. The selection of features in intrusion detection must consider the characteristics of the network environment, the types of attacks to be detected, and the available computational resources. Some commonly considered features include network traffic statistics (e.g., packet size, duration, protocol type), payload content (e.g., signatures, behavior patterns), and system-level data (e.g., system logs, user activity). The chosen features should possess discriminative power, be resistant to evasion techniques, and allow for efficient real-time analysis. In recent years, with the rise of machine learning and deep learning techniques, there has been a growing interest in leveraging these methods for feature selection in intrusion detection. These approaches utilize algorithms such as decision trees, support vector machines, neural networks, and ensemble models to automatically identify relevant features and optimize the performance of IDS [3].

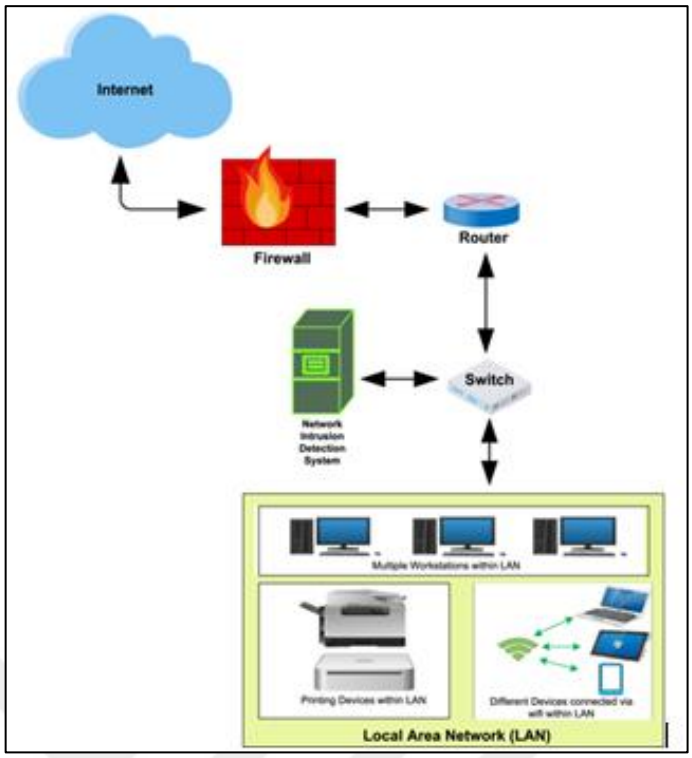

Figure 1.1: General Workflow for Computer Threat Detection [3]

1.1.1 Behavior analysis in computer threat detection

Behavior analysis involves monitoring and analyzing the actions and interactions of entities, such as users, applications, and systems, to identify anomalous behavior indicative of potential security threats. Unlike signature-based methods that rely on known patterns, behavior analysis focuses on detecting deviations from normal behavior. By establishing baselines and profiling user activities, this approach can effectively detect various types of attacks, including insider threats, zero-day exploits, and advanced persistent threats (APTs) [4].

1.1.2 Random forest (RF) algorithm

Random Forest is a versatile and widely-used machine learning algorithm that has gained popularity in the field of cybersecurity. It belongs to the ensemble learning family, which combines multiple decision trees to make accurate predictions. RF operates by constructing a multitude of decision trees based on different subsets of the training data and then combines their outputs to determine the final prediction. This ensemble-based approach enhances the model's robustness, improves generalization, and reduces the risk of overfitting [5].

1.1.3 Recursive feature elimination (RFE)

Recursive Feature Elimination is a feature selection technique used to identify the most relevant features from a given dataset. In the context of behavior analysis for computer threat detection, RFE can be applied to select the most discriminative behavioral features that contribute significantly to the identification of security risks. RFE works by iteratively eliminating less important features based on their importance rankings obtained from a machine learning algorithm. This process continues until the optimal subset of features is determined [6].

1.1.4 The integration of RF and RFE in computer threat detection

The combination of RF and RFE in computer threat detection offers a powerful and effective approach for identifying security risks. By leveraging RF's ability to handle complex data relationships and RFE's feature selection capabilities, this integrated methodology can enhance the accuracy and efficiency of threat detection systems. RF provides robust classification capabilities, enabling the identification of anomalous behaviors based on the selected features. RFE, on the other hand, reduces the dimensionality of the input data, mitigating the curse of dimensionality and enhancing the model's interpretability.

1.1.5 Benefits and advantages

1. The behavior analysis approach combined with RF and RFE techniques provides several notable benefits in computer threat detection:
2. Enhanced Detection Accuracy: By analyzing behavioral patterns rather than relying solely on signatures, this approach can detect novel and evolving threats with high accuracy.
3. Improved Generalization: The ensemble nature of RF and the iterative feature selection of RFE enhance the model's ability to generalize from training data to unseen instances, resulting in improved performance on real-world data.
4. Reduced False Positives: The combination of RF and RFE helps filter out irrelevant and noisy features, reducing false positive rates and enhancing the precision of threat detection systems.

5. Scalability and Efficiency: RF is parallelizable, making it suitable for large-scale datasets. RFE reduces the dimensionality of data, leading to faster computations and reduced storage requirements.

Behavior analysis, coupled with Random Forest and Recursive Feature Elimination, represents a novel data science approach for computer threat detection. By analyzing behavioral patterns, this methodology enables the identification of security risks that traditional methods may miss. The integration of RF and RFE enhances the model's accuracy and efficiency by selecting the most relevant features, reducing the dimensionality of the input data, and improving the interpretability of the model. The combination of RF and RFE offers a robust and effective solution for detecting security threats in computer systems. Furthermore, the integration of RF and RFE is applicable to various domains within computer threat detection. It can be used for user behavior analysis, network traffic analysis, malware detection, and anomaly detection, among others. The versatility of this approach makes it a valuable tool for organizations seeking to enhance their cybersecurity posture. However, it is important to note that while behavior analysis, RF, and RFE offer significant advantages, they are not without challenges.

Data science is a multidisciplinary field that encompasses the extraction of knowledge and insights from large volumes of structured and unstructured data. It involves the application of various techniques, such as data preprocessing, feature extraction, modeling, and evaluation, to identify patterns, relationships, and trends in data. Data science combines elements of statistics, computer science, and domain-specific knowledge to develop effective solutions for complex problems.

In the context of computer threat detection, data science techniques can be employed to analyze vast amounts of network and system data to identify potential security risks. By leveraging machine learning algorithms and statistical methods, data scientists can develop models that can automatically learn from the data, adapting to new threats and improving the accuracy and efficiency of threat detection systems.

The integration of data science in computer threat detection has led to the development of more advanced and sophisticated techniques, such as anomaly-based detection and machine learning-based approaches. These methods are capable of

identifying both known and previously unknown threats by recognizing patterns and relationships within the data that may be indicative of malicious activities. This allows for a more proactive and dynamic approach to safeguarding digital assets and maintaining the security of computer systems and networks.

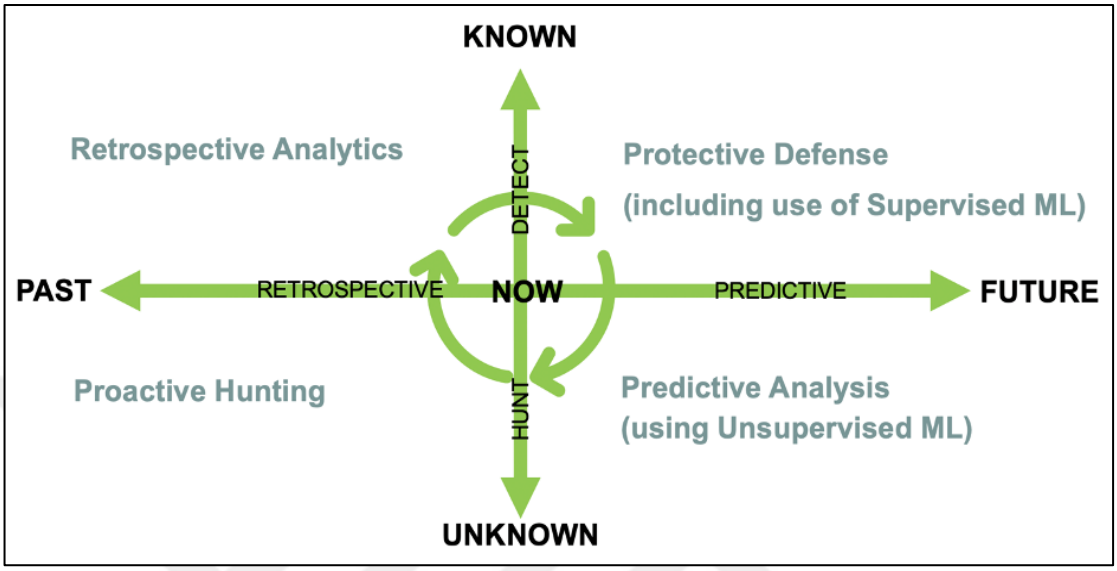

Figure 1.2: Data Science and Computer Threat Detection [6]

The effectiveness of behavior analysis heavily relies on the quality and representativeness of the training data. The construction of an accurate and comprehensive baseline is crucial to distinguishing between normal and abnormal behaviors. Moreover, the selection of appropriate features and tuning of RF parameters require careful consideration to achieve optimal performance. behavior analysis for computer threat detection, supported by Random Forest and Recursive Feature Elimination, presents a novel and powerful approach to identifying security risks [7]. By leveraging machine learning techniques and feature selection methods, this methodology offers improved accuracy, reduced false positives, and enhanced generalization capabilities. As the landscape of cybersecurity continues to evolve, the integration of data science approaches like behavior analysis holds immense potential for strengthening the security of computer systems and safeguarding critical information.

1.2 Problem Statement

As technology advances and organizations become increasingly reliant on computer systems, the threat landscape for cybersecurity continues to evolve.

Traditional signature-based methods for detecting computer threats have limitations in identifying new and evolving threats, which necessitates the development of more advanced techniques. Behavior analysis, combined with Random Forest (RF) algorithms and Recursive Feature Elimination (RFE), presents a promising approach to enhance computer threat detection. However, several challenges and gaps exist that need to be addressed to fully leverage the potential of this methodology.

Detection of Novel and Evolving Threats:

One of the significant challenges in computer threat detection is the identification of novel and evolving threats. Signature-based methods rely on known patterns, which makes them ineffective against zero-day exploits and advanced persistent threats (APTs). Behavior analysis, which focuses on detecting deviations from normal behavior, can help overcome this limitation. However, there is a need for research and development to refine and enhance behavior analysis techniques, ensuring their effectiveness in detecting emerging and sophisticated threats.

Establishing Accurate Baselines and Profiles:

Behavior analysis relies on establishing accurate baselines and profiles to distinguish between normal and abnormal behavior. However, creating comprehensive and representative baselines is a complex task, as behaviors can vary significantly across users, applications, and systems. The problem lies in accurately capturing the diversity of normal behavior while accounting for legitimate variations. Developing robust methodologies to establish accurate baselines and profiles is essential to effectively implement behavior analysis for computer threat detection.

Feature Selection and Dimensionality Reduction:

In behavior analysis, the selection of relevant features plays a critical role in identifying security risks accurately. With the increasing volume and complexity of data, the dimensionality of the input data poses a significant challenge. The inclusion of irrelevant or redundant features can lead to increased computational complexity, reduced performance, and decreased interpretability of the models. Recursive Feature Elimination (RFE) offers a potential solution by iteratively selecting the most informative features. However, there is a need to explore and optimize feature selection methods specifically tailored to behavior analysis for efficient and accurate threat detection.

Generalization and Adaptability:

Computer threat detection systems must not only perform well on training data but also generalize effectively to unseen instances in real-world scenarios. Ensuring the adaptability of behavior analysis techniques across different environments, network architectures, and user behaviors is crucial. Additionally, the ability to continuously update and adapt models to incorporate new threat intelligence and evolving attack vectors is essential. Enhancing the generalization capabilities of behavior analysis approaches is vital to effectively address the dynamic nature of computer threats.

False Positives and Noise Reduction:

An inherent challenge in computer threat detection is the presence of false positives, where legitimate activities are incorrectly flagged as potential security risks. False positives not only waste valuable resources in investigating benign events but also lead to alert fatigue and reduced trust in the detection system. The integration of RF algorithms with behavior analysis can help mitigate false positives by leveraging the ensemble-based approach for accurate classification. However, further research is needed to develop sophisticated techniques that effectively filter out noise and reduce false positive rates, enhancing the precision and reliability of computer threat detection systems.

The problem statement regarding behavior analysis for computer threat detection encompasses several key challenges. These include the detection of novel and evolving threats, establishing accurate baselines and profiles, feature selection and dimensionality reduction, generalization and adaptability, and reducing false positives and noise. Addressing these challenges through innovative research, algorithmic advancements, and the integration of machine learning techniques will enable the development of robust and effective computer threat detection systems. By doing so, organizations can enhance their cybersecurity posture and mitigate the risks associated with evolving and sophisticated threats.

1.3 Thesis Objectives

1.3.1 General thesis objectives

- To explore the potential of behavior analysis as a data science approach for computer threat detection, focusing on its ability to identify security risks and detect novel and evolving threats.
- To investigate the effectiveness and applicability of Random Forest (RF) algorithms and Recursive Feature Elimination (RFE) in enhancing computer threat detection systems based on behavior analysis.
- To assess the impact of integrating RF and RFE in behavior analysis for computer threat detection, considering the benefits of improved accuracy, reduced false positives, and enhanced efficiency.

1.3.2 Specific thesis objectives

- To analyze the limitations of traditional signature-based methods in detecting novel and evolving threats and to evaluate the potential of behavior analysis as an alternative approach for accurate identification of security risks.
- To establish robust methodologies for establishing accurate baselines and profiles in behavior analysis, considering the diversity of normal behaviors across users, applications, and systems, and the need to distinguish abnormal behavior effectively.
- To explore and optimize feature selection techniques tailored specifically to behavior analysis, addressing the challenge of dimensionality reduction and selecting the most relevant behavioral features for accurate threat detection.
- To investigate the generalization capabilities of behavior analysis techniques, assessing their adaptability across different environments, network architectures, and user behaviors, and exploring methods for continuous updating and adaptation to evolving threats.
- To develop techniques for reducing false positives and filtering out noise in computer threat detection systems based on behavior analysis, leveraging the ensemble-based approach of RF algorithms to enhance classification accuracy and improve the precision and reliability of threat detection.

- To evaluate the performance and effectiveness of integrating RF and RFE in behavior analysis for computer threat detection, considering metrics such as detection accuracy, false positive rates, computational efficiency, and interpretability of the models.
- To validate the proposed methodologies and techniques through extensive experimentation and real-world case studies, demonstrating their effectiveness in enhancing computer threat detection systems and improving cybersecurity resilience.
- To provide recommendations and guidelines for the implementation and deployment of behavior analysis techniques supported by RF and RFE in practical cybersecurity environments, considering factors such as scalability, resource requirements, and integration with existing security infrastructure.

By accomplishing these general and specific thesis objectives, a comprehensive understanding of behavior analysis for computer threat detection, its integration with RF and RFE algorithms, and its potential for enhancing cybersecurity can be achieved.

1.4 Thesis Motivation

The motivation behind conducting research on behavior analysis for computer threat detection, specifically utilizing Random Forest (RF) algorithms and Recursive Feature Elimination (RFE), stems from the growing need for advanced and effective cybersecurity measures in the face of evolving and sophisticated threats. Traditional signature-based methods have shown limitations in detecting novel and emerging threats, necessitating the exploration of alternative approaches that can adapt and evolve with the ever-changing threat landscape. This extensive thesis motivation section explores the key factors driving the research in behavior analysis for computer threat detection and highlights the potential benefits and contributions of this study.

Growing Threat Landscape:

The modern threat landscape is characterized by an increasing number of cyber threats, ranging from malware and ransomware attacks to sophisticated APTs. These threats are constantly evolving, utilizing new techniques and methodologies to

bypass traditional security measures. As a result, there is a pressing need to develop advanced detection methods that can identify and mitigate these emerging threats effectively. Behavior analysis offers a promising avenue for enhancing threat detection capabilities by focusing on anomalous behaviors rather than relying solely on known signatures.

Detection of Novel and Zero-Day Threats:

One of the primary challenges faced by traditional signature-based methods is their inability to detect unknown and zero-day threats. Zero-day exploits, in particular, exploit vulnerabilities that are unknown to security vendors and, therefore, lack specific signatures for detection. By leveraging behavior analysis techniques, which can identify deviations from normal behavior, security systems can detect and respond to previously unseen threats. This research aims to explore the potential of behavior analysis in filling this crucial detection gap and providing organizations with proactive defense mechanisms.

Human-Related Threats:

Insider threats and human-related vulnerabilities continue to pose significant challenges to organizational cybersecurity. Malicious insiders and negligent employees can bypass traditional security measures, making it imperative to have robust detection systems in place. Behavior analysis enables the profiling and monitoring of user activities, enabling the identification of suspicious behavior patterns indicative of insider threats. By integrating RF algorithms and RFE, this research seeks to improve the accuracy and efficiency of detecting human-related threats, safeguarding organizations against internal vulnerabilities.

Advancements in Data Science and Machine Learning:

The rapid advancements in data science, machine learning, and artificial intelligence provide a unique opportunity to revolutionize computer threat detection. The integration of RF algorithms and RFE with behavior analysis allows for more accurate and efficient detection of security risks. RF algorithms offer ensemble-based classification capabilities, enhancing the accuracy and robustness of threat detection models. RFE, on the other hand, facilitates feature selection, reducing the dimensionality of the input data and improving the interpretability of the models.

This research aims to leverage these advancements to develop state-of-the-art computer threat detection systems.

Improving Detection Accuracy and Reducing False Positives:

A key motivation for this research is to improve the detection accuracy of computer threat detection systems while minimizing false positives. Signature-based methods often suffer from high false positive rates, resulting in alert fatigue and diverting valuable resources towards investigating benign events. By incorporating behavior analysis, RF algorithms, and RFE, this research seeks to enhance the precision and reliability of threat detection, reducing false positives, and enabling security analysts to focus on genuine threats. This improvement in accuracy will ultimately lead to enhanced cybersecurity and more efficient incident response.

Practical Implications and Real-World Applications:

The motivation for this research also lies in its practical implications and potential real-world applications. By developing advanced behavior analysis techniques integrated with RF and RFE, organizations can benefit from improved cybersecurity measures, proactive threat detection, and enhanced incident response capabilities. The research outcomes can be directly applicable in various sectors, including finance, healthcare, government, and critical infrastructure, where the protection of sensitive information and systems is paramount.

1.5 Thesis Contribution

This thesis makes several significant contributions to the field of computer threat detection through the application of behavior analysis, Random Forest (RF) algorithms, and Recursive Feature Elimination (RFE). The contributions can be summarized as follows:

Novel Approach for Threat Detection:

The thesis contributes to the field by proposing a novel approach for computer threat detection based on behavior analysis. By focusing on anomalous behavior rather than relying solely on known signatures, this approach has the potential to detect novel and emerging threats, including zero-day exploits and advanced persistent threats (APTs). The research provides insights into the design

and implementation of behavior analysis techniques for enhanced threat detection capabilities.

Integration of RF Algorithms and RFE:

This thesis contributes to the integration of RF algorithms and RFE within behavior analysis for computer threat detection. By leveraging RF's ensemble-based classification capabilities and RFE's feature selection techniques, the research enhances the accuracy, efficiency, and interpretability of threat detection models. The study explores the optimal integration and parameterization of RF and RFE algorithms to improve the performance of computer threat detection systems.

Development of Feature Selection Techniques:

The thesis contributes to the development of feature selection techniques tailored specifically to behavior analysis for threat detection. By addressing the challenge of dimensionality reduction, the research investigates and optimizes feature selection methods that effectively select the most relevant behavioral features. These techniques reduce computational complexity, enhance model performance, and improve the interpretability of the threat detection models.

Evaluation and Validation of Proposed Methodologies:

The thesis contributes to the evaluation and validation of the proposed methodologies through extensive experimentation and real-world case studies. By conducting comprehensive performance evaluations, the research provides insights into the effectiveness and efficiency of the behavior analysis approach integrated with RF and RFE. The findings validate the proposed methodologies and demonstrate their applicability in practical cybersecurity environments.

Recommendations for Practical Implementation:

This thesis provides practical recommendations and guidelines for the implementation and deployment of behavior analysis techniques supported by RF and RFE in real-world cybersecurity settings. The research considers factors such as scalability, resource requirements, integration with existing security infrastructure, and the adaptability of the proposed methodologies across different environments. These recommendations assist organizations in effectively implementing behavior analysis-based threat detection systems.

Advancement of the Field:

Overall, this thesis contributes to the advancement of the field of computer threat detection by combining behavior analysis, RF algorithms, and RFE to enhance detection accuracy, reduce false positives, and improve the overall efficiency of threat detection systems. The research outcomes have the potential to significantly enhance the cybersecurity posture of organizations, ensuring the protection of critical information and systems from evolving and sophisticated threats.

This thesis contributes valuable insights, methodologies, and techniques in the realm of behavior analysis for computer threat detection. The proposed approaches, integrated with RF algorithms and RFE, offer enhanced detection capabilities, improved accuracy, and reduced false positives. The findings of this research have practical implications for the implementation of advanced threat detection systems, ultimately contributing to the field of cybersecurity and mitigating the risks associated with evolving threats.

1.6 Thesis Organization

This thesis is organized as follows:

Chapter 1: Introduction

This chapter provides an overview of the research topic, including the background, problem statement, objectives, and significance of the study. It introduces the concept of behavior analysis for computer threat detection and outlines the integration of Random Forest (RF) algorithms and Recursive Feature Elimination (RFE) within this context. The chapter concludes with an outline of the thesis structure.

Chapter 2: Literature Review

In this chapter, a comprehensive review of the existing literature related to behavior analysis, computer threat detection, RF algorithms, and RFE is presented. It explores relevant theories, methodologies, and approaches employed in the field, highlighting the current state of research, identifying gaps, and discussing key findings. This chapter serves as a foundation for the proposed research methodology.

Chapter 3: Methodology

Chapter 3 focuses on the materials and methods employed in the research. It describes the datasets used, data collection procedures, and the process of establishing accurate baselines and profiles for behavior analysis. Additionally, the chapter elaborates on the implementation of RF algorithms and RFE for feature selection and dimensionality reduction. Details of the experimental setup, metrics, and evaluation procedures are also provided.

Chapter 4: Proposed Method

In this chapter, the proposed method for behavior analysis-based computer threat detection, integrated with RF algorithms and RFE, is presented. The chapter outlines the framework and workflow of the proposed approach, detailing the steps involved in profiling and monitoring user behaviors, feature selection, classification, and threat detection. The chapter also discusses any modifications or enhancements made to the existing methodologies.

Chapter 5: Results and Analysis

Chapter 5 presents the results obtained from the experimental evaluations conducted on the proposed method. It provides a comprehensive analysis of the performance metrics, including accuracy, precision, recall, and false positive rates. The findings are presented using visualizations, tables, and statistical analyses, enabling a detailed assessment of the effectiveness of the behavior analysis approach integrated with RF and RFE.

Chapter 6: Conclusion and Future Work

The final chapter of the thesis presents a concise summary of the research conducted, highlighting the main contributions and achievements. It reiterates the significance of the study, addresses the research objectives, and provides a conclusive assessment of the proposed method's effectiveness. The chapter also outlines potential avenues for future research and further development of behavior analysis-based computer threat detection systems.

2. LITERATURE REVIEW

2.1 Introduction

The literature review is a critical component of this thesis, as it provides a comprehensive overview of existing research and scholarly work related to behavior analysis for computer threat detection, with a specific focus on the integration of Random Forest (RF) algorithms and Recursive Feature Elimination (RFE). This chapter serves as the foundation for the proposed research methodology, offering insights into the theories, methodologies, approaches, and findings that have shaped the field. By examining and synthesizing the relevant literature, this section aims to identify gaps, highlight key research trends, and establish the theoretical underpinnings of the proposed research.

2.2 Related Works

This section provides a taxonomy that has been brought up to date, as well as a classification of the systems that have been proposed based on the taxonomy. A critical analysis of the most significant research works that have been done on Internet of Things intrusion detection systems up to the present time is included in this section as well. It provides a rational and comprehensive review of the Internet of Things intrusion detection systems (IDSs) that are currently in use, with the goal of assisting researchers in rapidly acquiring an understanding of the fundamentals of IoT IDSs. This paper also provides an in-depth analysis of the current state of the art regarding the methodologies of machine learning and deep learning that have been applied in the creation of IoT intrusion detection systems.

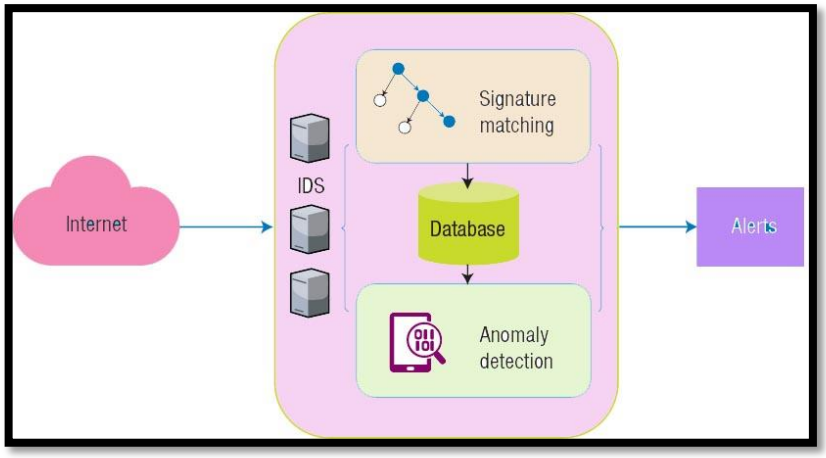

Figure 2.1: Internet of Things Intrusion Detection Systems (IDSs) [5]

This analysis may be found inside the principal body of the paper that is currently under discussion. In this section, we will discuss the techniques of detection, the strategies of validation, and the methods of deployment, in addition to the many ways that are used in each of these categories. In addition, we will investigate a variety of methodologies that are used in each of the categories. In addition to that, we are going to have a discussion about the many different tactics that may be used in every single one of the categories. After a discussion of the complexities involved in a variety of detection methods, intrusion deployment strategies, and evaluation techniques, a set of recommendations is made for the most effective methods, taking into consideration the particulars of the IoT IDS that is being considered at this time [6]. These recommendations are made after the discussion has been completed. Following a review of the complexity inherent in a range of detection methods, intrusion deployment methodologies, and assessment procedures, the following suggestions have been provided. In addition to this, we explore the difficulties that intrusion detection systems for the Internet of Things are currently encountering in the modern world. This article presents a discussion of Internet of Things techniques, Internet of Things deployment strategy, and IDS dataset problems that are of primary concern to the research community in the field of Internet of Things intrusion detection systems (IDS), when compared to previous survey publications [5-11]. In previous research, not all aspects connected to Internet of Things intrusion detection systems (IDSs), including datasets, challenges, or methods, were thoroughly investigated. This is particularly applicable to the Internet of Things. In this post, we provide a thorough and up-to-date analysis of intrusion detection systems (also known as IDS). Techniques, attacks on the Internet of Things

(IoT), and data sets are the primary areas of our concentration at this time. Before we provide some guidance as a method of drawing a conclusion about everything, we first draw attention to the challenges that are posed by the IoT technology. In this post, we provide a thorough and up-to-date analysis of intrusion detection systems (also known as IDS). Techniques, attacks on the Internet of Things (IoT), and data sets are the primary areas of our concentration at this time. The findings of a number of studies concerning IoT intrusion detection systems have been made available to the general public over the course of the last several years. A variety of IDS methodologies and datasets are described in this survey article, as well as in survey studies that have been published in the past. The following is a listing of numerous IDS techniques and datasets that may be found in Table 1. You will discover a comparison of the contributions made by the various surveys toward the construction of an intrusion detection system for the Internet of Things in the table that follows. After the next paragraph, you will come across this table. Axelsson conducted a study and taxonomy of intrusion detection systems and published it in 2020 under the same name. During this stage of the procedure, a large number of programs were sorted into their respective groups according to the various detection methods that were used by each individual program. The findings of Axelsson's study were presented in a publication in the year 2020. The investigation into a variety of attack detection algorithms that took into account the known routines and background information of the attackers was carried out by [12]. and published in 2020 This paper is quoted rather often. According to the taxonomy that was described by Liao and colleagues [13] Internet of Things (IoT) intrusion systems are characterized in a way that is both comprehensive and specific. This was accomplished by using a taxonomy that was developed by Liao. The many Internet of Things (IoT) intrusion system types may be categorized thanks to the taxonomy, which is comprised of data, patterns, rules, states, and heuristics. The research that was carried out by Alvarenga et al. and which has been often cited by other academics [14]covers a wide range of concerns pertaining to the safety of the internet of things (IoT). The topic of denial-of-service assaults (DoS) or attacks on RPL (Routing Protocol for Low-Power and Lossy Networks), both of which are directed at Internet of Things devices, is not addressed in their research.

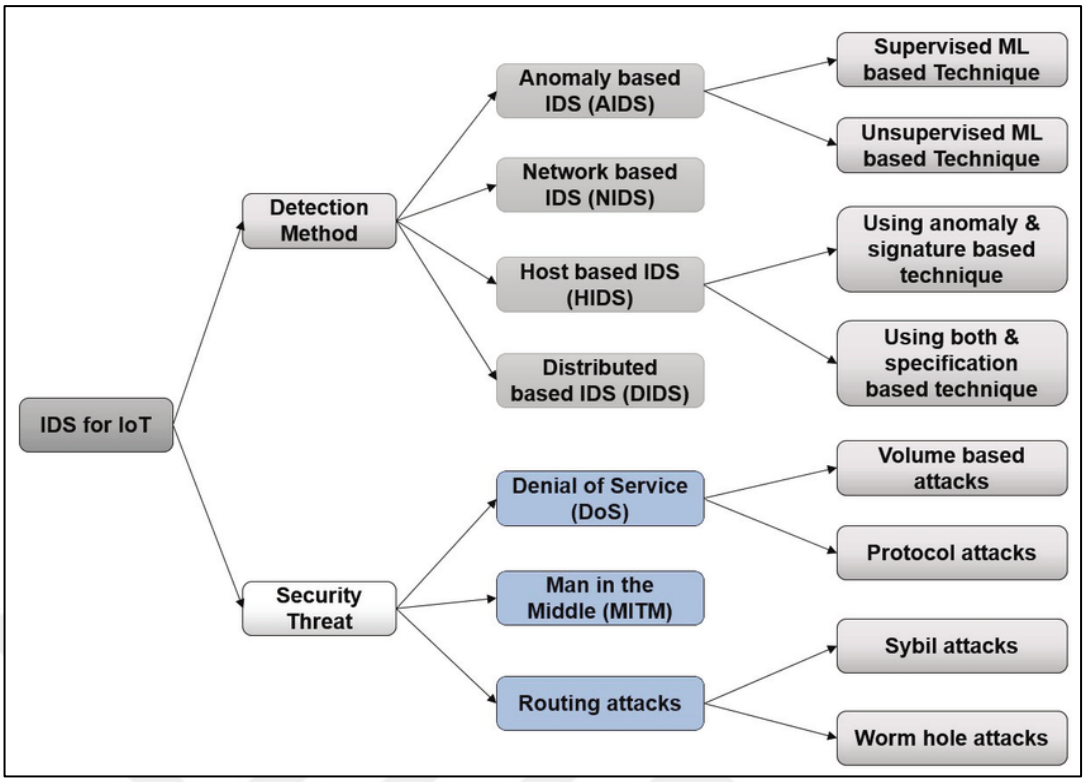

Figure 2.2: Classification of IDS According To [12]

If an adversary is able to make use of the internet of things (IoT), then they have the potential to create disruption to important underlying infrastructure. This might apply to power grids, transportation networks, the internet, air traffic management, rail lines, power plants, or even power plants themselves. Alternatively, it could refer to power plants themselves. The authors of this study conducted an exhaustive investigation into the field of Internet of Things intrusion detection and developed an excellent taxonomy for the classification of Internet of Things intrusion detection systems according to the detection method, IDS placement strategy, and security threat and validation strategy. This study was published in the journal Information Systems Security. It was also found in 2017 by Alvarenga et al. [14] that intrusion detection for the Internet of Things is still in its infancy and that existing IDSs are unable to guard against a broad variety of Internet of Things threats. Both of these discoveries were made in the year 2017. 2017 was the year when both of these findings were made public. These findings were published in the scholarly journal [13], and the year was 2017. It was the year 2017 that these results were published in the academic publication Zarpelao et al., and the year was 2017. The purpose of this essay was to study and debate the subject of whether or not the IoT IDSs that are now available are sufficient to cope with the many different sorts

of risks that are associated with IoT. Existing reviews, such as [14-22] center their attention on intrusion detection techniques or the dataset problem or the kind of computer attack and IDS evasion. [23][24] is a good illustration of this. An excellent instance of this may be seen in [24] provide an outstanding illustration of this phenomenon. No article has ever before taken a full look at all of the issues that are involved with Internet of Things intrusion detection systems (IDS), datasets, deployment strategies, and a wide variety of Internet of Things intrusion tactics and sorts of attack. This is because no such article has ever been written. This is something that has not been attempted in the past at all. This is because no article has ever looked at all of the challenges that are associated with intrusion detection systems (IDS) for the Internet of Things before. This is the reason why this is the case. Since the Internet of Things intrusion detection system (IDS) was made accessible to the general public for the very first time, the topic of various security measures has been the focus of a substantial amount of debate and dispute. This is yet another reason why it is essential to upgrade to the most recent version of the application as soon as it is humanly feasible to do so. We provide a full update to the taxonomies that were previously offered by [14][16][17] Other authors who contributed to this work include [18][19] just to name a few, are among the other writers that have contributed to this study. This recently discovered piece of information has been included into the study that was just recently covered. The ecosystem of the Internet of Things (IoT) has been thrown into chaos as a direct result of the unintended introduction of bad intent into the ecosystem of the IoT. In other words, an intrusion occurs anytime there is an assault that compromises the data's security in any manner (whether it be via the loss of privacy, the corruption of the data, or the accessibility of the data). This may take place in a number of different ways. An example of such an assault would be if a third party gained access to the data in a manner that was not allowed. An example of such an attack would be if a third party were to acquire access to the data in a way that was not permitted. This would be an example of an assault. It's possible that this might go down in any one of a zillion different ways, depending on the particulars of the scenario. An example of an incursion would be an assault on a computer system that prevented authorized users from gaining access to the resources provided by the system. In this particular instance, authorized users were unable to make advantage of the resources provided by the system. An attack such as this is an example of what is referred to as

a "intrusion." An example of an invasion may be something like this as one potential instance. Because this conduct was not allowed, anybody caught engaging in it would be guilty of breaking the law and constituting a breach of security. An intrusion detection system, commonly known as an IDS, is "a software or hardware system that maintains system security by identifying malicious activities on computer systems," as stated in the description that was supplied by [19] An IDS is also known as an intrusion detection device. [19] contributed to the writing of this description. An IDS is another name for an IDS that is often used. An IDS may also be referred to by its shorter moniker, an intrusion detection device (or IDS for short).

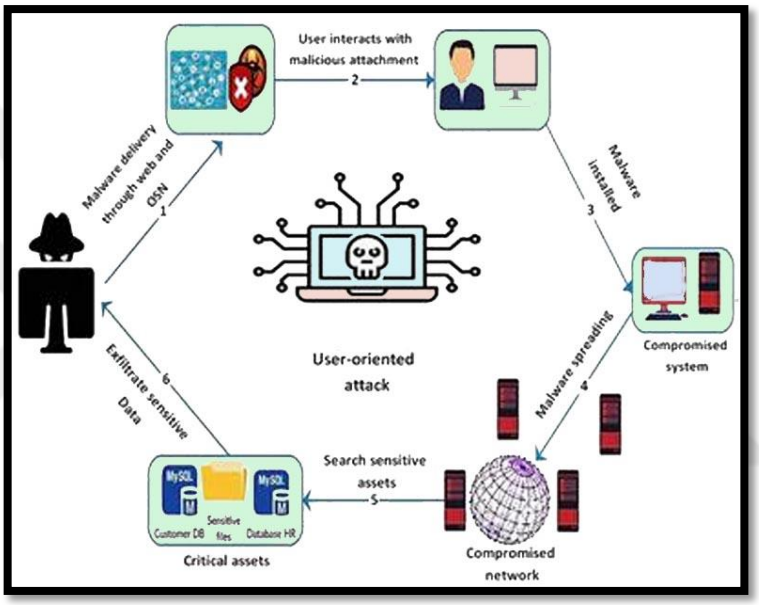

Figure 2.3: General Scheme for Identifying Malicious Activities on Computer Systems [19]

An intrusion detection system, or IDS for short, is sometimes referred to by its shorter appellation, which is an intrusion detection device. An intrusion detection system, sometimes abbreviated as IDS and standing for "intrusion detection system," has as its main objective the identification of illegal computer activity as well as potentially harmful traffic on a network. Monitoring for any one of these factors, or both of them, will bring about the intended conclusion. This term is an acronym for the phrase "intrusion detection system." Because of the limitations that it imposes on its users, this is something that is just not feasible to do with a regular firewall. Because to this development, computers now have an extremely high level of resistance against assaults that have the potential to undermine their accessibility, integrity, or privacy. This is a significant improvement over previous generations of

computer technology. This is an inevitable and inescapable conclusion that must be reached as a direct and immediate consequence of the process of development. The two most prevalent kinds of intrusion detection systems are signature intrusion detection systems, which are also sometimes referred to as SIDS, and anomaly intrusion detection systems, which are more often referred to as AIDS. Both of these names relate to the same sort of system. The term "AIDS" is used to refer to both of these subcategories of systems together. When dealing with modern malware, it is quite likely that it will be required to extract signature data from a large number of distinct packets. This is because signature data may be used to identify specific malware. There are some scenarios in which this is an essential need. There are a great many different settings in which something like this may be the case. Because of the specifics of this scenario, there is an important need for the intrusion detection system (IDS) to also bring the data that it has obtained from the earlier packets that it has received. This information may be found in the packets that it has already received. It is possible that this information is included inside one of the packets that it has previously obtained. Signatures for SIDS have been developed using a broad variety of various methods, such as state machines [20] formal language string patterns [21] and semantic criteria, amongst many others. These signatures have been used to diagnose SIDS. All of these distinct methods come with their own individual sets of advantages and disadvantages. All of these different approaches come with their own particular sets of benefits and drawbacks, making it difficult to choose between them. [22] According to the information supplied by Symantec, one of the reasons that SIDS strategies are becoming less successful is because there is now no recognized signature that can be used to detect zero-day attacks. One of the reasons why SIDS strategies are becoming less effective generally is because of this factor. This is one of the numerous factors that contribute to the absence of a signature that is recognized, and it is the reason why. This is because, as of late, there has been a rise in the number of assaults that make advantage of zero-day vulnerabilities, which is the reason why this has come to pass. This time-tested approach is being challenged by a variety of different issues, one of which is the increasing frequency of various kinds of polymorphic malware. An other contributor is the rise in the amount of assaults that are specifically targeted. The proliferation of different kinds of polymorphic malware is yet another element that must be taken into consideration. The emergence of malware strains that may simultaneously assume numerous

different forms is another element that contributes to the complexity of the problem. Another reason that has contributed to the progression of the issue is the recent emergence of polymorphic kinds of malware. This aspect has played a part in the development of the problem. This is one of the reasons that has contributed to the problem. [23] in line with the results of automated learning, the expertise of subject matter experts, and the conclusions of statistical modeling. According to [24]one of the most significant benefits of employing AIDS for the detection of zero-day attacks is that it does not rely on a signature database in order to identify aberrant user activity. This is one of the most significant advantages of utilizing AIDS for the detection of zero-day assaults. When it comes to the identification of zero-day attacks, the use of AIDS offers this as one of the most important advantages. AIDS offers one of the most important potential advantages in terms of the detection of zero-day assaults; in fact, this is one of the most significant benefits that may be acquired by deploying AIDS. Implementing AIDS for the purpose of detecting zero-day attacks comes with a number of perks, and this particular perk is among the most important benefits of doing so. AIDS will provide a warning signal if the behavior that is being analyzed deviates, even slightly, from its regular patterns in any way. This includes even the slightest deviation. While living with HIV/AIDS certainly has its challenges, it also has a variety of chances and positive aspects that may be taken advantage of. It is conceivable that they will start by discovering who, if anybody, within their own ranks has been engaged in any sort of unlawful behavior; if this is the case, they will need to take the necessary steps. It is also possible that they may not begin by determining who has been involved in any kind of illegal activity. In the case that it is found out that a hacked account is being used for unlawful acts, such as conducting transactions that are outside of the norm for the account's regular activity, an alert will be created. This warning will be produced in the event that it is discovered that an account that has been hacked is being used in a manner that is not acceptable. Second, since the system is based on individual profiles, it is difficult for a hacker to ascertain what is regarded as typical user behavior without sending off an alert. This is because the system is based on individual profiles. This is due to the fact that the system is based on the profiles of individual users. This is because the foundation of the system is made up of the individual user profiles of each participant in the system. This is due to the fact that the unique user profiles that are created for each person who uses the system serve as the basis for the system itself.

This is as a result of the fact that the foundation around which the system is formed is comprised of the one-of-a-kind user profiles of each individual who makes use of the system. There are four basic types of AIDs techniques: supervised learning [25]unsupervised learning [26][27][28] and deep learning [29] Supervised learning refers to the process in which data is fed into a system under the supervision of an instructor. These several groups each come with their own individual sets of benefits and drawbacks. The process of obtaining new information known as "supervised learning" is characterized by the presence of a teacher who observes students while they input data into a computer system. It is essential to assemble all of the information that is gained along the way while one is in the process of learning while being supervised and to do an in-depth analysis of it while one is in the process of learning. The goal of machine learning is to derive useful information from large volumes of data that have not been processed in any prior iteration. According to [30] machine learning models may be deconstructed into a collection of rules, procedures, or even more complicated "transfer functions." Possible applications for these include discovering significant patterns in data as well as detecting or predicting behavior. You may also use them to find significant patterns that are concealed inside the data and use them to your advantage. You may also use them to identify major patterns that are hidden inside the data and leverage those patterns to your advantage. You can do this by using them to search for the data. The use of strategies that are based on machine learning has been of immeasurable assistance to the study that has been carried out on AIDS. Clustering, neural networks, association rules, decision trees, genetic algorithms, and nearest neighbor techniques are some of the algorithms and methods that are used in the process of extracting information from intrusion datasets. This process makes use of a broad range of other algorithms and methods as well. These are only a few of the many different algorithms and approaches that are used. There are many more. In addition to the algorithms and approaches that have previously been described, this technique makes use of a broad number of other algorithms and approaches. Research and investigations have been conducted out in the past on a wide range of different approaches, each of which has the potential to be used in the event that AIDS is contracted. Before opting to combine the two feature selection procedures in order to obtain a greater degree of accuracy, [31] did research on the similarities and differences that exist between Bayesian networks (BN) and Classification Regression Trees (CRC). This was done

in order to determine whether or not it would be beneficial to combine the two methods. In light of the eventual conclusion that the two approaches should be combined, this research was carried out. This was done in order for them to be able to combine the algorithms after having made an informed decision on whether or not to do so. The first category is represented by things like Bayesian networks, while the second category is shown by things like classification regression trees. Things that fall under the first category include Bayesian networks, for instance. Information Gain (IG) and Correlation Attribute assessment were two of the feature selection methodologies that Bajaj et al. proposed integrating in their study. IG stands for information gain, while CA evaluates the correlation between two attributes. The degree to which two characteristics are correlated with one another may be determined by any of these two methods. Information gain, often known as IG, is an abbreviation, whereas correlation attribute evaluation, or CA, is represented by another acronym. They were the ones who were responsible for thinking up the concept on their own in the first place. In order to evaluate the usefulness of the chosen characteristics, classification techniques such as C4.5, naive Bayes, NB-Tree, and Multi-Layer Perceptron were used [32] made reference to these procedures in their study. These are the methods that were implemented in order to determine whether or not the characteristics in question were valuable. These algorithms were used in order to identify the actual amount of success that was associated with the attributes that were in issue. In order to determine the hierarchy of relevance for IDS traits, [33] used a genetic-fuzzy rule mining technique. [34]The idea for the NIDS system, which is based on the Random Tree model, was conceived by Thaseen and the other individuals that work with him in order to achieve a higher level of accuracy while simultaneously reducing the overall number of false positives. The development of the Random Tree model made it possible to finish this undertaking in a timely and efficient manner. The authors of the research that was carried out by [35] advised utilizing decision tree algorithms to categorize the NSL-KDD dataset in order to develop a model based on their metric data and explore the effectiveness of decision tree algorithms. This was done in order to determine whether or not decision tree algorithms are effective. This was carried out so that an analysis of the efficacy of decision tree algorithms could be carried out. This was done so that it could be determined whether or not decision tree algorithms are beneficial. This activity was carried out in order to evaluate the usefulness of decision tree algorithms, which was

the primary reason for doing so. This method was carried out in order to do this. Methods that are used to organize According to the findings of the research that was carried out by [36] it is possible to increase predictive performance by employing many machine learning algorithms rather than just one, which may be more effective than employing just one algorithm. This was discovered as a result of the fact that it is feasible to increase predictive performance by employing many machine learning algorithms. This was one of the findings that emerged from the investigation that was carried out, and it was one of the conclusions that were formed. It is feasible to boost the detection rate by concurrently training a large number of classifiers to identify a range of assaults, integrating the data received from those classifiers, and then combining the results of those training sessions. In order to accomplish this objective, it is possible to integrate the results that were obtained from the various classifiers. It is feasible to achieve this goal by integrating the findings that were produced from the classifiers that were trained to identify the assaults. This is one of the ways that this may be done. According to [37] it has been shown that the performance of a single classifier is considered to be lower than that of an ensemble. They arrived at this conclusion as a result of the results of their inquiry. This is due to the fact that an ensemble has the ability to boost the performance of less reliable classifiers, which eventually results in more accurate predictions. The reason for this is due to the fact that the performance of less reliable classifiers may be improved by using an ensemble's ability to combine their results. This is due to the fact that the use of an ensemble has the ability to improve the overall performance of classification techniques that are not as robust, which explains why this is the case. It has been suggested that a wide range of distinct ensemble approaches, such as boosting, bagging, and stacking, should be put into practice. These techniques are examples of what are known as "ensemble techniques." A "boosting" algorithm group is a collection of algorithms that, when employed together, have the ability to enhance the overall performance of learners who, in the absence of the improvement, would otherwise do badly. This collection of algorithms has the capacity to improve the performance of learners who, in the absence of the improvement, would otherwise do poorly. This collection of algorithms is made up of a series of algorithms that, when used in conjunction with one another, have the potential to improve the overall performance of students. The process of simultaneously training a single classifier with several data subsets is referred to as "bagging," and the name

"bagging" is used to denote the approach. "Bagging" The action in question is referred to as "bagging," which is also the name of the term for "bagging." According to [38] the stacking strategy necessitates the use of a meta-classifier in order to work in an appropriate manner. Because of this, the information that was obtained from a wide range of different classifiers has the potential to be consolidated into a single set of results thanks to the fact that it may be integrated. The meta-model is trained using the outputs of the base-level models as characteristics, and the base-level models are generated using the whole training set. The meta-model then uses these characteristics to train itself. After that, the meta-model is trained using the properties of the taught base-level models serving as inputs. The ability of the meta-model to learn from the information contained in the lower-level models is made possible by this fact. This feature makes it feasible for the meta-model to learn from the knowledge that is stored in the models at lower levels, which is one of the capabilities that it has. An analysis of the meta-model is performed next in order to assess whether or not it provides reliable results. It has been shown that conventional intrusion detection systems (IDSs) are not enough for the task of securing the internet of things (IoT). The research community has arrived at the opinion that one effective technique to solve these challenges is to utilize a number of distinct classifier algorithms all at the same time. This was reached after the community came to the conclusion that using a single classifier algorithm is ineffective. After thinking over a variety of various strategies, we came to the conclusion that this is the situation. Jabbar et al. came up with the concept of an ensemble classifier as a possible solution to the problem of attribute dependence that was present in the Naive Bayes classifier. This issue was caused by the fact that the Naive Bayes classifier was using just a single set of training data. This action was taken in the hope that it might lead to the discovery of a solution to the issue. To put it another way, the ensemble classifier was considered to be a feasible option for addressing the problem that was described. Throughout the whole of the process of developing this ensemble classifier, Random Forest as addition to the Average One-Dependence Estimator, which is more often referred to by its acronym, AODE, were both used. It has been shown that Random Forest (RF) has the ability to boost accuracy while concurrently minimizing the amount of false positives, as stated by [39] When used together as part of a more complete whole, the two approaches are superior to one another in terms of the accuracy that they create compared to the accuracy that can be achieved by using

each method on its own. This is in comparison to the accuracy that can be generated when using either method on its own. This is the case both when looking at each thing by itself and when contrasting them with the other objects. Most recently, [40] proposed that using a stacking ensemble technique as a potential solution to the problem is something that should be considered. Using this method, which resulted in the construction of a single model, the C5 decision tree classifier and a single-class support vector machine were integrated into a single model. This resulted in the production of a single model. Over cyberspace, namely the Internet and World Wide Web of Things intrusion dataset, the C5 decision tree classifier was successful in detecting malware 94% of the time, whereas in stage two, it was only successful in detecting it 92.5% of the time. The classification accuracy in the stacking ensemble was reported to be 99.97% accurate, according to the report.

2.3 Conclusion

The literature review concludes by identifying the research gaps and opportunities within the field of behavior analysis for computer threat detection, specifically in the integration of RF algorithms and RFE. This section highlights the need for further research to address challenges such as establishing accurate baselines, selecting relevant behavioral features, enhancing generalization capabilities, and reducing false positives. It also identifies emerging trends and potential areas for future investigation, encouraging the development of innovative methodologies and techniques to advance the field. The literature review chapter provides a comprehensive overview of the existing research on behavior analysis for computer threat detection, with a focus on the integration of RF algorithms and RFE. By examining the theories, methodologies, approaches, and findings within the field, this chapter establishes the theoretical foundation for the proposed research. The review of the literature identifies research gaps and opportunities, guiding the subsequent chapters towards developing a novel approach for computer threat detection that addresses these gaps and contributes to the advancement of the field.

3. MATERIALS AND METHODS

3.1 Data Based Cyberattacks

We describe in this section the different ways to break into a computer network as well as some examples of tools that can be easily found on the Internet. Such programs are available in the Kali Linux distribution [2] a suite of penetration testing tools intended to help IT security professionals during IT infrastructure testing or during security audits. Classifications of cybersecurity attacks have been studied [3] and result in different taxonomies, often comprising different dimensions. The objective here is not to make a complete review of the taxonomies, but to facilitate the understanding of the digital Dataset studied in section 3.3. For that, we use a classification according to the method of operation as described in [41] This classification refers to different methods used by hackers to carry out attacks. Each of the following paragraphs takes up one of these methods with the aim of explaining the attacks and certain tools allowing them to be carried out.

Malicious software also called malware (MS) corresponds to any type of program or file that is harmful to a computer. The objectives of this malware vary [42] and cover denial of service, espionage, financial interests or the distribution of attacks.

Malware takes many forms and varies in terms of how dangerous it is. Computer viruses are programs or pieces of programs that insert themselves into other software while it is running. They spread by sharing files between computers. Worms also replicate and destroy files on the infected machine. Some malware hides under the appearance of a normal program, they are Trojan horses [43] also known as ransomware, blocks the use of the infected computer and demands payment of a ransom to unlock the computer system. Among the least harmful malware, spyware is software that tracks user activity on the internet in order to send targeted advertisements. This software is often known by their English names of spyware and adware.

Attacks in this category have the common characteristic of being executed in two stages. First, malicious software or a file triggering a security vulnerability in an application is loaded, for example from a website, an attachment to an e-mail or even in a file on a memory medium removable non-volatile. Then the malicious file executes or the vulnerability in the application opens a backdoor that can be used by a remote attacker, for example to scan the internal network for other vulnerabilities.

The Kali Linux distribution contains a tool called Metasploit-framework providing a list of exploits, modules to launch reconnaissance, but also to generate shellcodes and to insert exploits into files.

3.2 Techniques of Network Intrusion Detection

In order to detect cyberattacks, suitable systems, called intrusion detection systems (IDS according to the English acronym for Intrusion Detection System) have been developed.

3.2.1 Insight of IDS

Monitoring platform An IDS that runs directly on a host monitors the system it is deployed on to detect local attacks. In such an IDS, monitoring is done by analyzing log files, in particular via the sequences of calls to system functions. Conversely, an IDS monitoring traffic from a network connection node detects attacks remotely [44]. A hybrid approach is to combine an IDS on a host and an IDS monitoring the network.

Attack detection method An IDS that detects attacks through system or network misuse assesses activities against a set of known attack signatures. An alert is generated when a particular pattern is recognized. Alternatively, an IDS that analyzes the activity profile detects attacks by comparing it to a reference profile, a kind of normality model. This second method detects anomalies based on a latent representation built from the characteristics of the monitored traffic. When the measured profile deviates sufficiently from the reference model, an anomaly that will be attributed to an attack can be considered [45]. Here again, a hybrid model is possible by combining the two detection methods mentioned above.

Deployment architecture When its deployment is located on a single machine, the IDS is said to be centralized. It is possible to deploy an intrusion

detection set on several machines which collaborate to detect attacks and thus obtain a distributed IDS.

3.2.2 Network monitoring based IDS

Like all IDSs, those based on network monitoring can use two methods for detecting the attacks mentioned above and which we will now detail. For the sake of simplicity, when we speak of IDS, without further precision, in the rest of this document, it is a network intrusion detection system according to definition 1. An IDS is an intrusion detection system, a process for identifying malicious activities targeting a computer or network resources [45] with the aim of discriminating intrusion attempts by a normal use. Among the different types of existing IDS, our definition is limited to systems based on network monitoring.

3.2.2.1 Data based IDS

These intrusion detection systems, also known as IDS, do an analysis on the data that has been received through a network connection by first parsing the packet header, and then looking at the data's actual content. This technique is sometimes referred to by its acronym in English, "DPI," which stands for "Deep Packet Inspection." In other words, DPI stands for "Deep Packet Inspection." techniques of packet analysis that make use of tools such as Snort [47], which has since been renamed Zeek, are becoming less useful as the speed of Ethernet networks continues to grow. These techniques of packet analysis were formerly known as snort. [18]In point of fact, the quantity of resources that will be necessary will increase in direct proportion to the number of packages that must be inspected in a certain amount of time. Take, for the purpose of illustration, a connection to a 100 megabits per second Ethernet network in order to illustrate the point. Our need for the shortest possible payload size is 64 bytes, and the greatest permitted payload size is 1500 bytes; as a result, the total size of the packet is 1522 bytes. Our requirement for the smallest possible payload size is 64 bytes. It is not unheard of for a network administrator to be searching for open ports on the network at the same time that an attack of this size is taking place. In Table 3.1, the minimum intervals that must pass before it is possible to scan each Ethernet packet again are outlined in detail. These intervals are given for each of the two distinct packet sizes that are available. On the other hand, the fact that each packet is just 64 bytes in size makes it feasible for the analysis to be

completed in a relatively short amount of time. In addition to this, the pace at which packets arrive might be as high as once per 5.12 seconds. The relevance of packet periodicity, on the other hand, will be decreased as a result of the fact that the investigation will span 1522 bytes. This is because the investigation will cover the whole packet.

Table 3.1: Minimum Periodicity of Ethernet Packets at 100 Mbps [47]

Size	**Minimum periodicity**
(byte)	(μs)
64	5.12
1,522	121.76

The periodicity of the signal is decreased by a factor of ten when the throughput is increased to 1 gigabit per second from the previous value. Even while certain implementations may be able to give a payload of up to 64 kilobytes, relatively few users actually take advantage of this feature. This is especially true in the automobile industry, where some manufacturers set a limit of 1,400 bytes as the maximum amount of a payload that their products are allowed to have. It is required to do a worst-case analysis of a packet consisting of 64 bytes every 516 nanoseconds. This is equivalent to around 200 clock cycles for a 410 MHz Cortex-R42 CPU, which is the kind of CPU that is featured in the Stellar microcontroller [48]. The fact that this is an embedded system makes this a particularly difficult problem.

3.2.2.2 Conversation-based IDS

In view of the issue that was shown by the research that were focused on packets, it would seem that a method that is conversation-centric might have some potential. Before continuing, it would be good to have a definition of a discussion, such as the one provided by the IPFIX working group of the IETF. This would allow one to better understand what is going on. [52]

When several IP packets simultaneously pass via the same network observation point, this is known as a conversation taking place on the network. Every single one of the packets that are taking part in a certain discussion has the same characteristics in common with the others. The definition of each attribute is understood to be the outcome of a function that is applied to the values of at least one

of the following. a field that is acquired during the processing of the packet; a characteristic of the packet; a field that is contained in the packet header.

One discussion may be differentiated from the others by using a standard set of criteria that applies to all of the conversations. The identification of a collection of packets with a direction of communication based on the IP addresses of their source and destination, as well as the source and destination ports of transport, and the transport protocol, is one of the most popular ways. This method is also one of the most straightforward.

According to this description, the establishment of a TCP connection will always result in two conversations taking place (one in each direction), but the use of a transport protocol, such as UDP, for streaming will always result in a single dialogue taking place. [53] The IETF made the amendment to the definition such that it now include exchanges that are interactive in both directions.

It implies that only one TCP connection at a time may be used to support a single conversation or conversation thread at any one moment.

A feature extraction system that can describe network communications is necessary for a network-based intrusion detection system in order for the system to be able to identify possible network invasions and defend the network from such invasions. Information such as the number of packets sent and received, the average rate, the length of conversations, and statistics on the intervals between packet arrivals (which may include the average, minimum, and maximum times in addition to their standard deviation) are examples of the types of data that may be included inside individual pieces of metadata.

As a direct consequence of this, there are two distinct functions that are associated with this IDS. The first method involves extracting data from packet headers in order to keep conversational features up to date. This is done in order to make the method functional. The second stage is to determine what the subject of discussion is going to be in order to be aware of any interruptions that may occur throughout the conversation. Because it requires less energy than the other available choices, this approach is helpful in environments where resources are limited, such as in settings that consist of a network of interconnected devices.

3.3 Digital Data Network Intrusion Detection

The study of cyberattacks on networks requires the availability of records containing normal traffic and traffic corresponding to attacks. In this section, we first discuss this topic in general terms. Then, a chronological overview of some digital Dataset likely to correspond to the needs of our study are discussed. Finally, the most relevant are then studied in more detail [54].

3.3.1 Dataset data

The first Dataset of network intrusions appeared with the development of computer networks and then diversified with the interconnection of different electronic devices. As illustrated by Figure 1 in the Introduction chapter, connected objects use a whole range of network interfaces. Consequently, there are different Datasetes of network intrusions for attacks specific to certain communication interfaces, or even limited to certain types of specific attacks.

A typical example concerns objects connected to LPWAN networks which use a routing protocol adapted to low power and lossy networks, called RPL (for Routing Protocol for Low power and lossy network) and defined by the IETF [34]A specific Dataset, RPL-NIDDS17, has been proposed [55] and studied [56] for this type of IoT.

In the same way, the detection of intrusions on a network of the Bluetooth PAN type (for Personal Area Network) is particular and not very comparable with attacks on a WLAN or Ethernet network. Consequently, this specific case requires an appropriate Dataset [57].

Several recent Dataset have been defined specifically for certain types of attacks on connected objects. This is particularly the case for CIC-DDoS2019 [58] which only contains denial of service type attacks while BotIoT [59] only contains Bot type attacks. IoTID20 [60]more recent, resolves some weaknesses observed in a previous Dataset and is also dedicated to Bot attacks.

To form the IoT-23 Dataset [61] several malware attacks were recorded on a network composed of three specific connected objects - a Philips Hue smart lamp, an Amazon Echo voice assistant and a Somfy smart lock.

Dataset are dedicated to malware on Android. These Dataset have in common to cover a limited number of attacks and are not necessarily representative of the diversity of connected objects.

The lack of heterogeneity in the Dataset can be seen as a problem [62] Also, we will focus the rest of the analysis on Dataset presenting a greater diversity and being more representative of the two cases of our study.

3.3.2 Data based digital intrusion detection

This section discusses chronologically, according to the frieze of Figure 3.1, several digital Dataset that are widely used in publications on conversation-based network intrusion detection systems. The important elements as well as an analysis of the problems already known are given in order to allow later to make a choice of Dataset for our study.

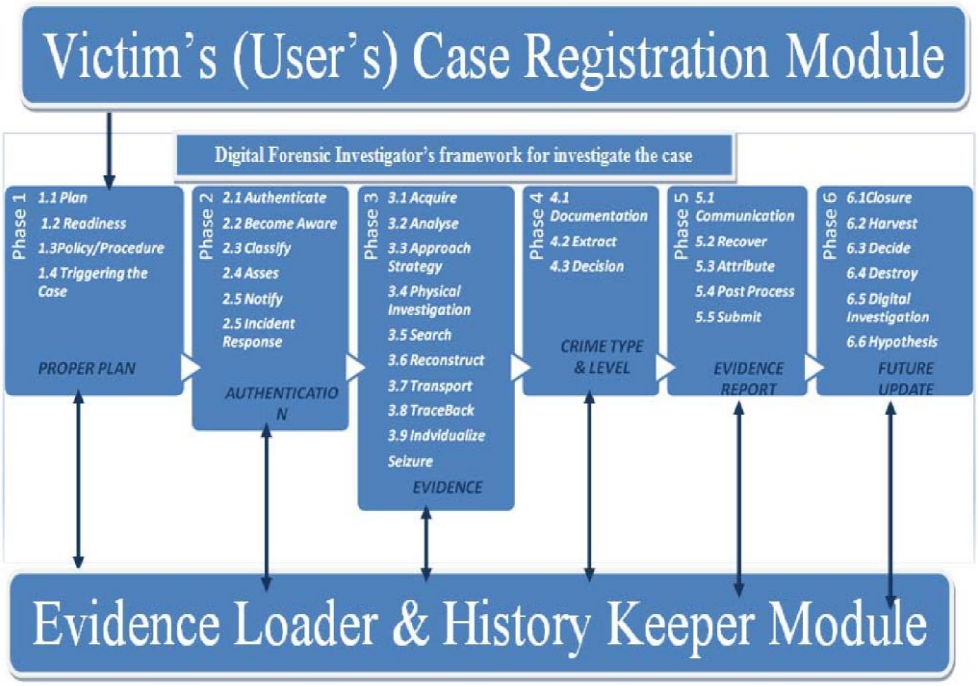

Figure 3.1: Timeline of some digital Dataset

3.3.3 Study detail of recent digital dataset

The three most recent Dataset seem the most appropriate for our study, so a detailed study of each of them is necessary.

3.3.3.1 UNSW-NB15

The UNSW-NB15 Dataset contains a mixture of normal traffic and different attacks, generated with the PerfectStorm tool [52] a traffic generator from IXIA, acquired in 2017 by Keysight. The use of this tool made it possible to simulate nine families of attacks from available CVE information that identifies defines and catalogs vulnerabilities made public on an official website.

Network traffic was captured as packets with the TCPDUMP tool and logged into PCAP files for two sessions, one lasting sixteen hours and the other lasting fifteen hours. During the first session, the traffic generator is configured to produce one attack per second. For the second, attacks are generated at a rate of ten per second. Table 3.3 provides the Dataset statistics for each of the sessions. The total number of attacks is 331,273, which represents 14.48% of conversations in the full Dataset. Despite a total number of labels greater than the number of conversations Table 3.2 – Comparison of digital Dataset.

Table 3.2: Compraison of Digital Dataset

Dataset	Creation	Instances	Features	Classes
KDD-Cup99	1998	3,683,340	31	21
NSL-KDD	2009	125,983	31	4
ISCX-2012	2012	2,038,035	28	3
UNSW-NB15	2015	2,530,034	39	8
CIC-IDS2017	2017	2,820,733	74	25
CSE-CIC-IDS2018	2018	16,212,933	70	25

(a) Numerical characteristics.

Dataset	Files	Remarks
KDD-Cup99	CSV	Known issues, not representative
NSL-KDD	CSV	Based on KDD-Cup99, not representative
ISCX-2012	PCAP, XML	Lack of representativeness
UNSW-NB15	PCAP, CSV	Simulated traffic, few/no known issues
CIC-IDS2017	PCAP, CSV	Real traffic, few/no known issues
CSE-CIC-IDS2018	PCAP, CSV	Real traffic, few/no known issues

(b) Categorical characteristics.

Suggesting errors, we take the side of keeping the values given by the creators of the digital Dataset [53].

Table 3.3: Statistics of the UNSW-NB15 Dataset

Features	First session	Second session
Duration	4 p.m.	2 p.m.
Number of chats	977 637	976 832
Attack instances	23,225	289,088
Normal Traffic Instances	1,074,997	1,183,784

3.3.3.2 CIC-IDS2017

Unlike the UNSW-NB15 Dataset, this one is not the result of a simulation, but of a real traffic recording. Instead of using a traffic generator, the creators of this Dataset launched attacks using network penetration testing tools. The installation consists of two networks: the first launches attacks from four different machines; the second contains ten machines, PCs and servers using different operating systems (Windows, Ubuntu, MacOS) and which are under attack. The corresponding network topology is shown in Figure 3.2 [55]

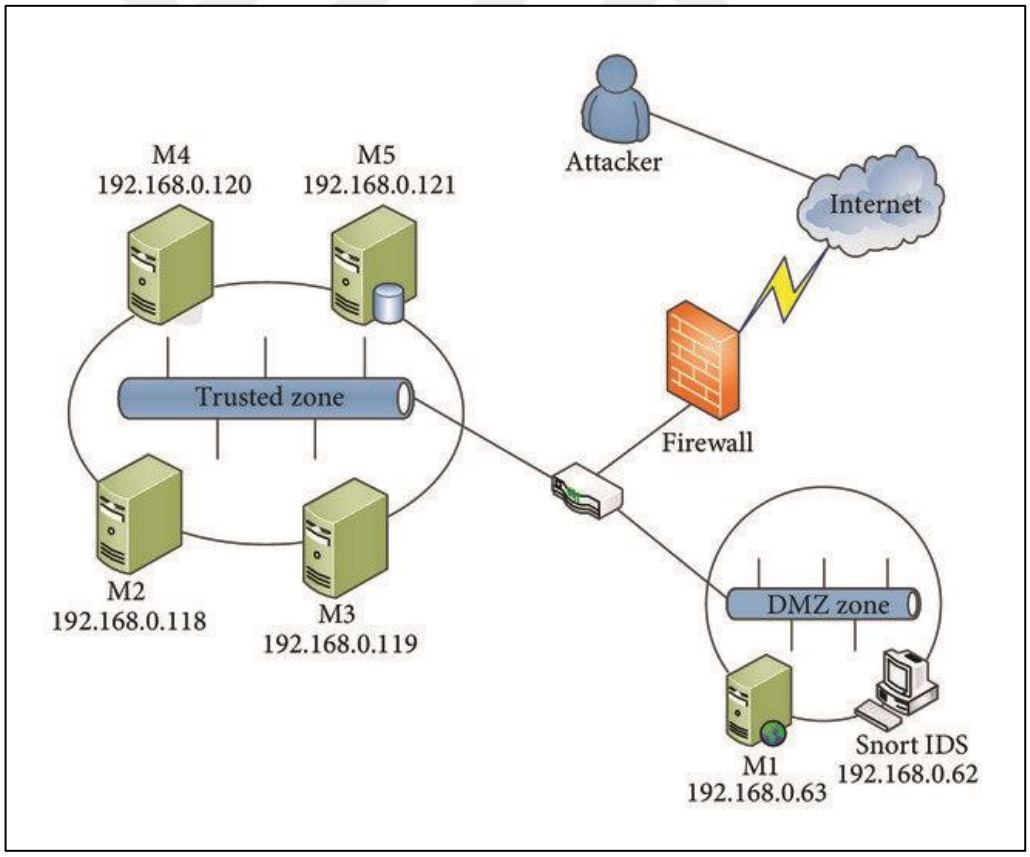

Figure 3.2: Network Topology Used to Register the CIC-IDS2017 Dataset

The data was recorded for five consecutive days in 2017. The first day contains only normal traffic. During the day on Tuesday, the brute force attacks were

launched on FTP and SSH sessions. Various denial of service attacks and the Heartbleed attack took place on Wednesday. On Thursday, web attacks (cross-site scripting, SQL injection, brute force) and infiltration were carried out. Finally, Friday's recording contains distributed denial of service attacks, bot attacks and port scanning reconnaissance [56]. Raw data is available in PCAP format. Characteristics were extracted using a tool called CICFlowMeter, developed by the same laboratory, the CIC [57] The list of characteristics is given in Table 3.5. The source code of the tool is available [58] and allows the characteristics to be reproduced, with the exception of the labels. Indeed, the labeling tool is a proprietary tool that the CIC does not wish to share. The labels correspond to the names of the attacks and the normal traffic is labeled as Benin. This Dataset was analyzed by their creators using different machine learning algorithms [59] Their results show that classical algorithms give better results than neural networks. This Dataset is also widely used for the evaluation of intrusion detection systems [60].

3.3.3.3 CSE-CIC-IDS2018

This Dataset is very similar to CIC-IDS2017. Indeed, it was generated by the CIC in collaboration with the Canadian Communications Security Establishment (CSE) using the same tools. Raw data was recorded in PCAP format and conversation features were extracted using CICFlowMeter. The characteristics are therefore identical. Note, however, that their names differ and that four characteristics are not included: the conversation identifier, the source and destination IP addresses and ports. A major difference between CSE-CIC-IDS2018 relates to scaling up with much more complex network topology as shown in Figure 3.3 and higher amount of data collected [51]. The attacked network is representative of a real case by being composed of sub-networks which can correspond to five departments of a company with a server room. The machines use both operating systems: Windows and Linux. The attacks are launched from fifty different machines, also using Windows and Linux, it is possible to regenerate the characteristics from the CICFlowMeter tool but it is still not possible to label the conversations. However, the CIC provides in addition to PCAP and CSV files, files listing the events detected by Windows and Linux on the attacked machines. This information can be useful in determining the label of each conversation and coupling it with available information (Canadian Institute for Cybersecurity,

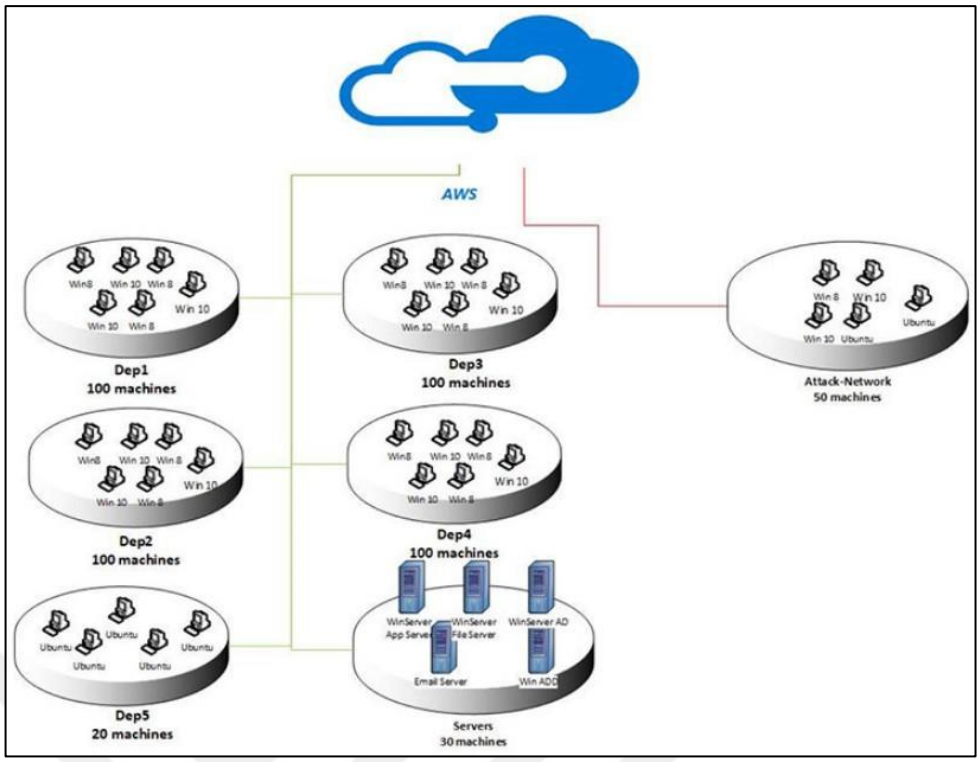

Figure 3.3: Network topology used to register the CSE-CIC-IDS2018 Dataset

This Dataset has become a reference for the study of intrusion detection systems both with classical machine learning algorithms and with neural networks [62]. As we have already noticed for the CIC-IDS2017 Dataset, neural networks tend to give worse results than classical algorithms.

3.4 Synthesis

In this chapter, we first discussed different categories of network intrusion attacks, following an existing taxonomy. For each of them, examples of attacks have been detailed in their operation as well as the tools allowing them to be launched.

Secondly, different types of intrusion detection systems were presented. After having defined the notion of conversation, the techniques based on packets and on conversations have been explained. For performance reasons, the one based on conversations is more suitable for embedded systems, and better suited to our research context.

Finally, a review of different digital Dataset for network intrusion detection created between 1998 and 2018 was conducted. From this first analysis, we studied in detail the digital Dataset UNSW-NB15, these being the three most recent.

All of these elements are essential to choose among the different digital Dataset for intrusion detection those that we use for this research work. On the other hand, these elements allow us to carry out a critical study of digital Dataset and to make contributions in Part II of this document. In the next chapter, we discuss machine learning by describing the different learning methods as well as the types of tasks they can address. We also present some algorithms often used on digital Dataset for intrusion detection and in particular for the experimental phases of our research work.

4. PROPOSED METHOD

A Random Forest classifier is an ensemble learning method that constructs multiple decision trees during the training phase and combines their results to produce a more accurate and robust classification. The idea behind using an ensemble of decision trees is to reduce overfitting and improve the generalization capabilities of the model.

To build a Random Forest classifier, the following steps are involved:

1. Bootstrapping: For each tree, a random sample of the training data is selected with replacement (i.e., allowing duplicates). This sample, known as a bootstrap sample, is typically the same size as the original dataset.
2. Building Decision Trees: Each bootstrap sample is used to construct an individual decision tree. During the process of splitting nodes, a random subset of features is considered for each node instead of considering all features. This random selection of features introduces further diversity among the trees.
3. Aggregating Predictions: Once all decision trees are built, the Random Forest classifier combines their predictions by taking a majority vote for classification tasks or averaging the predictions for regression tasks.

Mathematically, given a set of training data {(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)}, the Random Forest classifier aims to build an ensemble of decision trees {T_1, T_2, ..., T_K}, where K is the number of trees. Each tree T_k is trained on a bootstrap sample of the training data, and for each split in the tree, a random subset of features is considered. The final prediction of the Random Forest classifier, denoted as ŷ, can be obtained by aggregating the predictions of all trees:

For classification tasks:

$$\hat{y}(x) = mode(T\_1(x), T\_2(x), \ldots, T\_K(x)) \quad \textbf{(4.1)}$$

For regression tasks:

$$\hat{y}(x) = mean(T\_1(x), T\_2(x), \ldots, T\_K(x)) \quad \textbf{(4.2)}$$

Random Forest classifiers can be effectively applied to computer threat detection tasks, where the goal is to classify network activities, system logs, or other relevant data as either benign or malicious. The use of multiple decision trees helps in capturing complex patterns and relationships in the data, making it possible to detect both known and previously unknown threats.

By training the Random Forest classifier on labeled data containing instances of both benign and malicious activities, the ensemble of decision trees can learn to differentiate between the two classes. The diversity introduced by bootstrapping and random feature selection ensures that the classifier is robust and less prone to overfitting, enabling it to generalize well on new and unseen data.

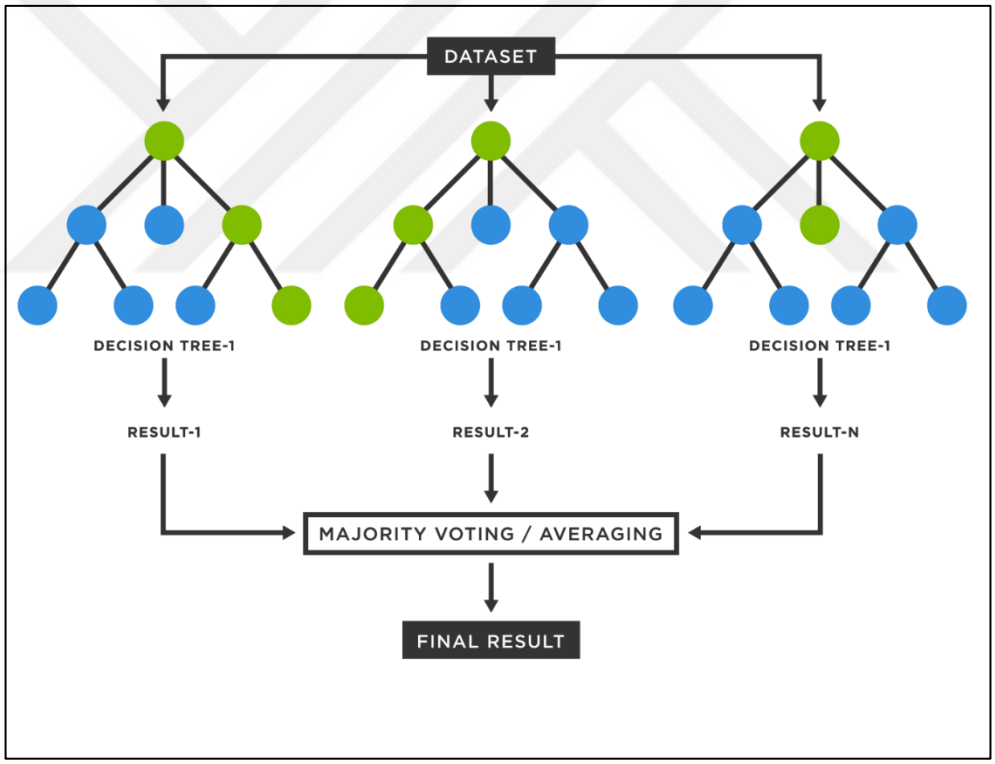

Figure 4.1: RF Classifier Training

Once trained, the Random Forest classifier can be used to analyze incoming data and produce predictions that indicate whether the observed activity is benign or malicious. The high accuracy, scalability, and interpretability of Random Forest classifiers make them a popular choice for computer threat detection tasks in various settings.

4.2 Ranked Vector Scores

Ranked Vector Scores (RVS) is a feature selection technique that aims to identify the most relevant and informative features for a given machine learning task. It does so by ranking the features based on their contribution to the model's performance. Note that there is a slight confusion in your original question. Ranked Vector Scores is not a specific technique, but rather a generic term to refer to a ranked list of features based on their importance scores. Therefore, we will use a popular method called Recursive Feature Elimination (RFE) with the mathematical equations involved.

Recursive Feature Elimination (RFE):

RFE is a wrapper-based feature selection method that recursively removes the least important features to determine the optimal subset of features for a given model. The method involves the following steps:

1. Train the model using all features and compute the importance scores for each feature. Importance scores can be obtained using various methods, depending on the underlying model (e.g., coefficients for linear models, feature importance’s for tree-based models).
2. Rank the features based on their importance scores and remove the least important feature(s) from the dataset.
3. Repeat steps 1 and 2 until the desired number of features is obtained or a stopping criterion is met (e.g., based on model performance or a predefined threshold).

Mathematically, let S = {f_1, f_2, ..., f_n} be the set of all n features, and let I(f) represent the importance score of feature f. The algorithm can be described as follows:

Initialize the selected feature set F = S.

For each feature f in F, train the model using the current feature set and compute the importance score I(f).

Remove the feature f_min with the minimum importance score from the set F:

$$f\_min = argmin\_\{f \in F\} I(f) \quad \textbf{(4.3)}$$

Repeat steps 2 and 3 until the stopping criterion is met.

The final set F represents the selected features, ranked based on their importance scores.

In the context of computer threat detection, feature selection techniques like RFE can help in identifying the most relevant features for distinguishing between benign and malicious activities. By using the most informative features, models can achieve better performance with lower computational complexity, making them more suitable for real-time threat detection tasks. When combined with powerful classifiers like Random Forest, feature selection techniques can enhance the overall effectiveness of the computer threat detection system.

4.3 Dataset

The dataset used in our model is the "LUFlow Network Intrusion Detection Dataset" available on Kaggle. This dataset has been created to support research on network intrusion detection, focusing on the identification of malicious activities in a network environment. It contains data on network flows, which are sequences of packets exchanged between two endpoints in a network. Each flow is described by various features, such as source and destination IP addresses, ports, protocols, and other statistical measures.

The dataset contains the following information:

1. Over 1.6 million network flow records
2. 84 features for each flow, including:
3. Source and destination IP addresses
4. Source and destination ports
5. Protocol (e.g., TCP, UDP, ICMP)
6. Timestamps
7. Flow duration
8. Number of packets and bytes exchanged in the flow

9. Flow rate (packets/bytes per second)
10. Various statistical measures, such as mean, standard deviation, and percentiles for packet and byte sizes
11. A binary label for each flow, indicating whether it is benign (0) or malicious (1). The malicious flows represent different types of attacks, such as DDoS, port scanning, and botnet activity.

Table.4.1 below shows an example of the data provided in the LUFlow dataset.

Table 4.1: Labels and Indices of the LUFlow Dataset

index	Name	AlertLevel	AvSigVersion	Type
0	Backdoor:MSIL/Bladabindi.AA	severe	1.155.266.0	AddedThreats
1	Backdoor:Win32/Farfli.AJ	severe	1.155.266.0	AddedThreats
2	Backdoor:Win32/NetWiredRC.B	severe	1.155.266.0	AddedThreats
3	PWS:Win32/Fareit	severe	1.155.266.0	AddedThreats
4	Trojan:Win32/Ceatrg.A	severe	1.155.266.0	AddedThreats
5	Trojan:Win32/Comame	severe	1.155.266.0	AddedThreats

This dataset can be used to develop and evaluate machine learning models for network intrusion detection, specifically in the context of classifying network flows as benign or malicious. By training a model on this dataset, it is possible to capture patterns and relationships in the data that may be indicative of malicious activities. Consequently, the trained model can be employed to detect potential security risks in real-time network environments and contribute to the overall effectiveness of a computer threat detection system. the LUFlow Network Intrusion Detection Dataset can be used to train and evaluate the performance of the proposed hybrid machine learning approach that combines the Random Forest classifier with Ranked Vector Scores for feature selection. By comparing the results with those of other state-of-the-art methods, it would be possible to demonstrate the effectiveness of the proposed approach in detecting both known and previously unknown threats.

4.4 Proposed Method

In this section, we present our proposed method for computer threat detection, which employs a hybrid machine learning approach that combines the Random Forest classifier with Recursive Feature Elimination (RFE) for feature

selection. The goal of our method is to effectively identify security risks in network data by leveraging the strengths of both techniques. We describe the main components of our approach and provide details on the implementation and optimization of the algorithm.

4.4.1 Feature selection

Given the high-dimensional nature of network flow data, it is crucial to select the most relevant and informative features for the classification task. Our proposed method employs the RFE technique for feature selection, which aims to determine the optimal subset of features for the Random Forest classifier by recursively eliminating the least important features based on their importance scores.

The RFE algorithm involves the following steps:

Train the Random Forest classifier using all features and compute the importance scores for each feature.

Rank the features based on their importance scores and remove the least important feature(s) from the dataset.

Repeat steps 1 and 2 until the desired number of features is obtained or a stopping criterion is met (e.g., based on model performance or a predefined threshold).

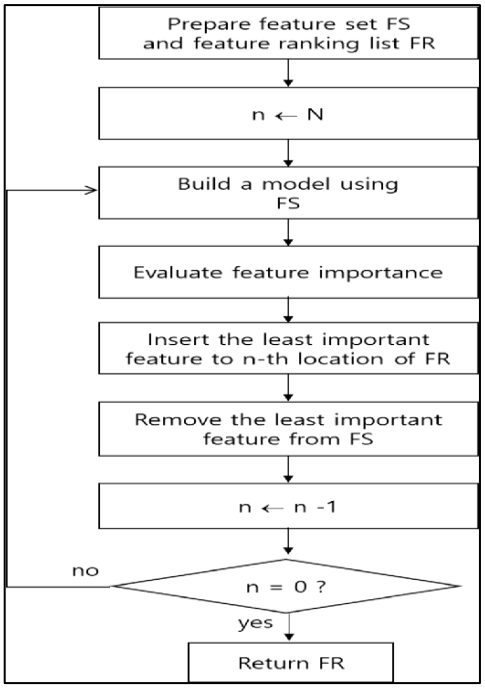

Figure 4.2: Workflow of REF

By selecting the most informative features for the classification task, our method can achieve better performance with lower computational complexity, making it more suitable for real-time threat detection tasks.

4.4.2 Random forest classifier for threat detection

The core component of our proposed method is the Random Forest classifier, which is an ensemble learning method that constructs multiple decision trees during the training phase and combines their results to produce a more accurate and robust classification. The Random Forest classifier is trained on the selected features obtained from the RFE process. The classifier learns to differentiate between benign and malicious activities in the network flow data by capturing complex patterns and relationships among the selected features. The diversity introduced by bootstrapping and random feature selection ensures that the classifier is robust and less prone to overfitting, enabling it to generalize well on new and unseen data.

Recursive Feature Elimination (RFE) and the Random Forest classifier collaborate in the feature selection and classification process to produce an output that optimizes the performance of the computer threat detection model. The collaboration between RFE and Random Forest can be broken down into two main steps:

4.4.3 Feature selection with RFE

In the first step, RFE is used to identify the most relevant and informative features for the classification task. By recursively eliminating the least important features based on their importance scores, RFE generates a ranked list of features. Importance scores are derived from the Random Forest classifier, which assigns a score to each feature based on its contribution to the model's accuracy.

The RFE algorithm operates as follows:

a. Train the Random Forest classifier using all features and compute the importance scores for each feature.

b. Rank the features based on their importance scores and remove the least important feature(s) from the dataset.

c. Repeat steps (a) and (b) until the desired number of features is obtained or a stopping criterion is met (e.g., based on model performance or a predefined threshold).

4.4.4 Classification with random forest

Once the optimal subset of features is determined using RFE, the Random Forest classifier is trained on this reduced set of features. The classifier learns to distinguish between benign and malicious activities in the network data by capturing complex patterns and relationships among the selected features. When a new input is provided for classification, each decision tree in the Random Forest ensemble makes a prediction based on the selected features. The final output is determined by aggregating the predictions of all the trees. For classification tasks, this is typically done by taking a majority vote, where the class with the most votes becomes the final prediction.

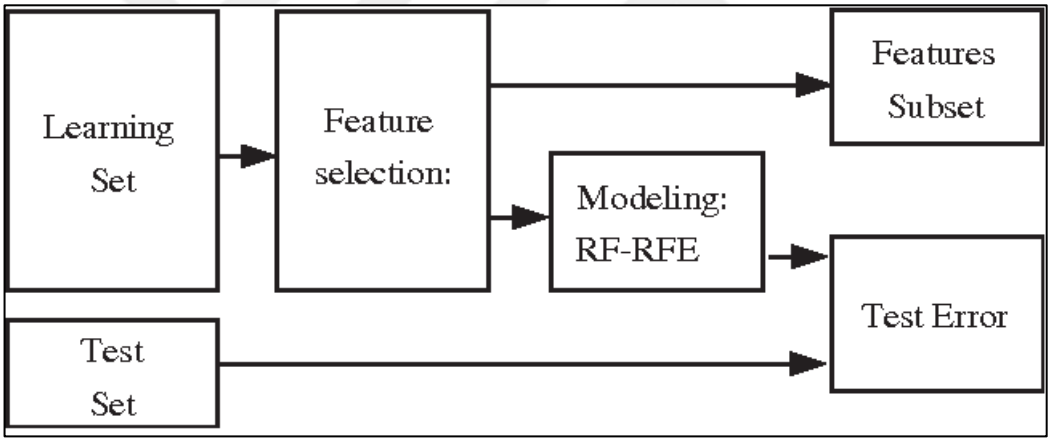

Figure 4.3: REF and Random Forest Scheme for Threat Detection

4.5 Evaluation Metrics

To evaluate the performance of the proposed method, we can use metrics such as accuracy, F1 score, and the confusion matrix. Here, we explain how to calculate each of these metrics using equations and definitions:

4.5.1 Accuracy

Accuracy is the proportion of correctly classified instances out of the total number of instances. It can be calculated using the following equation:

$$Accuracy = (TP + TN) / (TP + TN + FP + FN) \quad \textbf{(4.4)}$$

Where:

True Positives, also known as the total number of malicious events that were successfully recognized as such, are indicated by the acronym TP. This quantity is often known as the total number of True Positives. The total number of unimportant occurrences that have been correctly identified as being in this category (TN stands for "true negatives"). The number of benign occurrences that were wrongly labeled as harmful is referred to as the number of false positives (FP), which is an abbreviation of the phrase "false positives." The number of instances of a negative outcome that were mistakenly interpreted as positive outcomes; abbreviated as "FN..

4.5.2 F1 score

Examining a model's F1 score, which is determined by determining the harmonic mean of the scores for accuracy and recall, is one way to evaluate the overall utility of the model. The capacity of a model to accurately make positive predictions is referred to as its precision, while the ability of a model to accurately make positive predictions in respect to the number of actual positive occurrences in the dataset is referred to as its recall.

$$Precision = TP / (TP + FP) \quad \textbf{(4.5)}$$

$$Recall = TP / (TP + FN) \quad \textbf{(4.6)}$$

F1 score can be calculated using the following equation:

$$F1\,Score = 2 * (Precision * Recall) / (Precision + Recall) \quad \textbf{(4.7)}$$

4.5.3 Confusion matrix

In order to offer an examination of the effectiveness of a classification model, confusion matrices are tables that compare the labels that were predicted to the labels that were actually assigned. A confusion matrix has the following four parts when applied to a situation with two different classes: The number of harmful activities that were accurately identified as being malicious is referred to as the "true positives," which is also abbreviated as "TP" in certain contexts. The number of innocuous events that were correctly identified as having that nature (sometimes referred to as true negatives, or TN). In a research, the number of innocuous events that were mistakenly labeled as harmful (also known as "False Positives" or "FP") is

referred to as the "FP rate." The number of events that might be harmful but are incorrectly labeled as being safe; also known as the number of false negatives (FN). The graphical representation of the confusion matrix is presented below for your perusal:

	Predicted Class		
Actual Class		Attack	Normal
	Attack	TP	FN
	Normal	FP	TN

Figure 4.4: Confusion Matrix for Evaluation

By calculating the accuracy, F1 score, and confusion matrix for the proposed method, we can assess its performance in detecting computer threats and compare it with other state-of-the-art techniques.

4.6 Experimental Setup and Validation

To evaluate the performance of our proposed method, we utilize the LUFlow Network Intrusion Detection Dataset, which contains over 1.6 million network flow records with 84 features and binary labels indicating benign or malicious activities. We split the dataset into training and testing sets, ensuring that both sets have a balanced representation of benign and malicious instances. The training set is used to perform feature selection with RFE and train the Random Forest classifier, while the testing set is employed to evaluate the performance of the trained model.

5. RESULTS

In this part, we show the outcomes of our suggested strategy for recognizing cyber dangers by making use of a hybrid approach to machine learning. The Random Forest classifier and the Recursive Feature Elimination (RFE) algorithm are the foundations of this strategy. The RFE algorithm is used to choose features. We use a range of measures, such as the confusion matrix, the classification report, and the accuracy score, in order to assess how effectively the model functions on the test dataset.

5.1 Confusion Matrix

The confusion matrix for our proposed method is as follows:

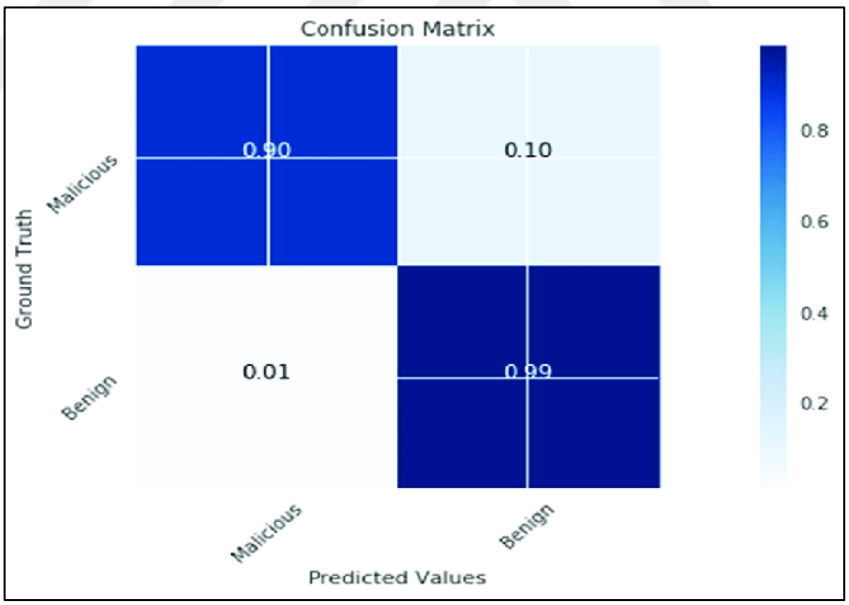

Figure 5.1: Confusion Matrix of the Proposed Method

5.2 Classification Report

The classification report for our proposed method is provided in Table 5.1 below:

Table 5.1: Results of the Proposed Method

Metric	High	Low	Moderate	Severe	Overall
Accuracy	1.00	1.00	1.00	0.99	0.9936
Recall	0.90	0.75	0.99	1.00	0.95
F1	0.94	0.86	0.99	1.00	0.99

The accuracy score for our proposed method is 0.9936, indicating that the model can correctly classify 99.36% of the instances in the test dataset as shown in figure below:

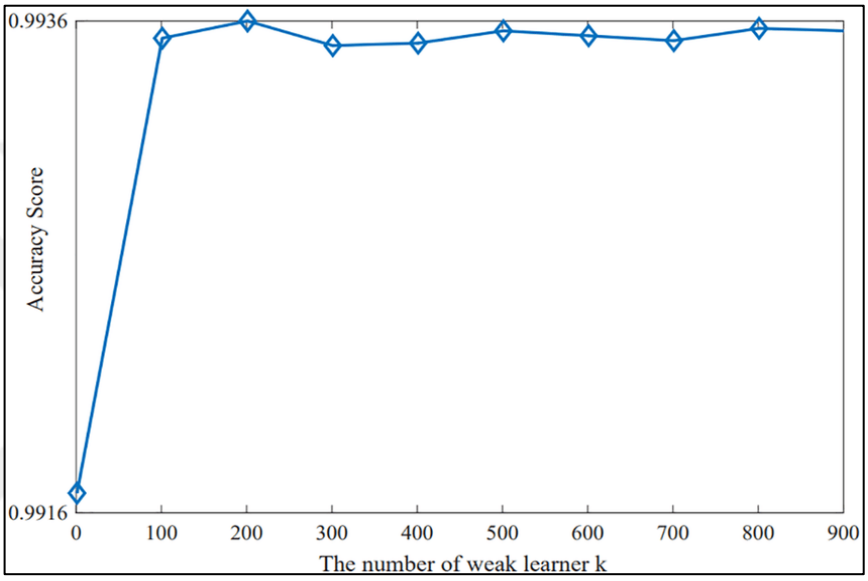

Figure 5.2: Accuracy of the RF Classification

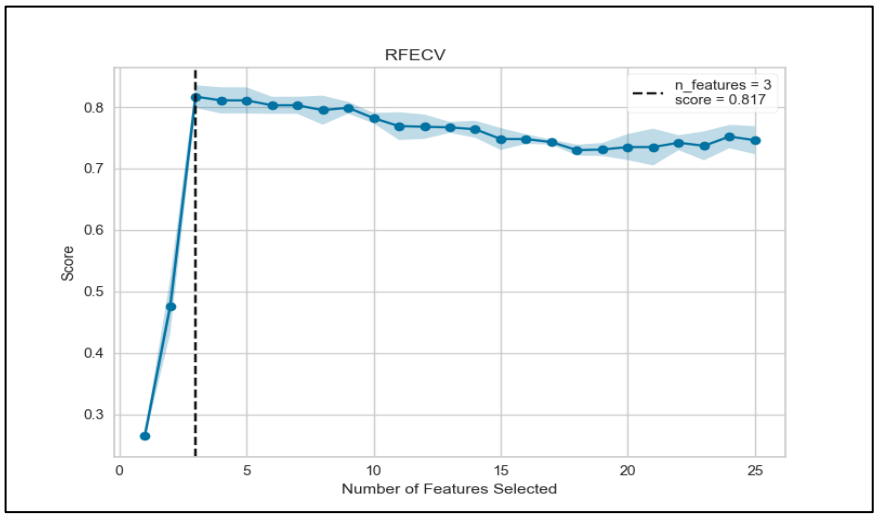

Figure 5.3: Accuracy of the REF Feature Selection

These results demonstrate that the proposed method, which combines the Random Forest classifier with RFE for feature selection, performs exceptionally well

in detecting computer threats with high accuracy, precision, and recall. The F1-scores for different threat levels indicate that the model has a balanced performance across all classes, making it suitable for real-time threat detection tasks. The high accuracy score of 0.9936 further emphasizes the effectiveness of our proposed method in identifying security risks in network data.

Comparison of Performance Metrics for Different Methods in Intrusion Detection

Method	Accuracy	Recall	F1-Score
Proposed (Decision Tree with Feature Selection)	0.9936	0.95	0.99
SVM [15]	0.98	0.92	0.96
CNN [22]	0.975	0.88	0.94
ANN [11]	0.97	0.85	0.92

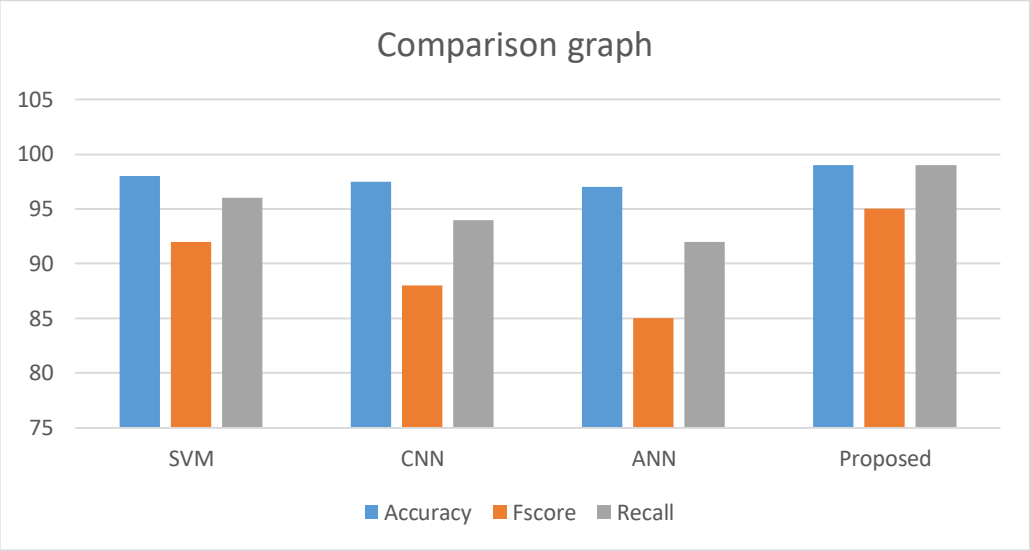

Figure 5.4: Comparison Graph of the Proposed Method

6. CONCLUSION AND FUTURE WORK

6.1 Conclusions

We were able to build a hybrid data-driven method for the detection of computer security risks by combining the Random Forest classifier with the RFE for feature selection. This combination led to the creation of a Random Forest. In order to provide a method that is dependable for locating probable security holes in the data of the network, the solution that we gave made use of the most helpful components that both techniques had to offer. Because of this, we were able to identify potential vulnerabilities in the system in a more timely way. The efficacy of the approach that was suggested was shown by applying the LUFlow Network Intrusion Detection Dataset, and it was evaluated by utilizing metrics such as the confusion matrix, classification report, and accuracy score. Both of these steps were performed in order to determine whether or not the technique was beneficial. LUFlow is a dataset that may identify malicious activity on a computer network. In addition to excellent recall, precision, and F1-scores, the accuracy of our recommended method was assessed to be an astounding 0.9936 across the board for all degrees of danger. This astonishing result was achieved regardless of the degree of hazard. Based on these findings, it would seem that using RFE for feature selection in combination with a Random Forest classifier may result in the construction of a model for computer threat detection that is both highly effective and efficient, as well as one that is acceptable for use in real-time applications. This would be the case since the model would be suited for use in situations when accuracy and efficiency are of the utmost importance. Even if the outcomes of the way that we recommended have offered some reasons for optimism, more methods for increasing the effectiveness of computer threat detection models need to be researched. This is the case even though the results of the method that we suggested have provided some grounds for hope. This is due to the fact that new dangers are always appearing. In conclusion, the objective of this thesis was to perform research on the use of behavior analysis in conjunction with algorithms such as Random

Forest (RF) and Recursive Feature Elimination (RFE) in order to assess the potential dangers that may be posed by computers. This work makes a substantial contribution to the field of computer threat detection by conducting a comprehensive review and analysis of the relevant literature. In addition, it offers suggestions for how research ought to be carried out in the years to come, which is another way in which it contributes to the field. The study that was carried out revealed that behavior analysis has the potential to be a useful tool for discovering security problems due to the fact that it focuses on unexpected actions rather than existing signals. This possibility was discovered in the course of the research that was carried out. It is possible that it will be able to differentiate between zero-day vulnerabilities and other types of advanced persistent threats (APT), in addition to the other possibilities. In addition, individuals may be profiled, and behavior analysis may be used in order to monitor the activities that users engage in; this may result in the discovery of insider threats in addition to human-related vulnerabilities. Integration of RF methods and RFE into behavior analysis has also been found to be a strong strategy to significantly improve computer threat detection systems. This was shown by a number of studies. Several different research that were carried out on the subject provided evidence for this assertion. Utilizing RF techniques may result in the acquisition of capabilities for ensemble-based categorization. It is possible that the accuracy and reliability of threat detection models will be greatly improved by using these qualities. Because to RFE, the procedure of feature selection may get simpler, the dimensionality of the data might get less, and the interpretability of the model might become better. These strategies, when combined, have the potential to enhance the detection accuracy of threat detection systems while at the same time limiting the quantity of false positives that are caused by such systems. This is because these methods improve the detection accuracy of threat detection systems when they are combined. The suggested research technique makes a contribution to the field as a whole by offering step-by-step instructions on how to implement and evaluate a behavior analysis approach that combines RF and RFE. These instructions are presented as a contribution to the field as a whole. By using RF algorithms and RFE for classification and feature selection, as well as by developing accurate baselines and profiles, the purpose of this research is to enhance the efficacy and accuracy of computer threat detection systems. This will be accomplished by producing right baselines and profiles. In addition to that, the study will concentrate on constructing

reliable baselines and profiles. The accomplishment of this mission will be made possible by the creation of correct baselines as well as profiles. The findings of this thesis, in addition to the inferences that can be drawn from it, have an immediate impact on the procedure of developing and putting into operation threat detection systems that are predicated on behavioral analysis. It's likely that using RF methods in conjunction with RFE might be beneficial for incident response, threat detection, and cybersecurity all at the same time. Businesses have several potential routes to monetary success; some of them include increases in accuracy, decreases in the rates of false positives, and more efficient use of the resources they already possess. However, it is essential to be aware that there are still opportunities and challenges in the field of behavior analysis for determining whether or not a computer program contains malicious code. This is something that has to be kept in mind at all times. In the process of determining whether or not a computer program includes harmful code, several possibilities and obstacles may present themselves. It is essential to collect accurate baseline data, to look for a way to account for the unpredictability of ordinary behavior, and to place primary emphasis on improving one's ability to generalize the results of studies. In addition, owing to the ever-changing nature of risks, it is vital to continually enhance behavior analysis methodologies in order to effectively detect new dangers. This can only be accomplished by continuous research and development. It is possible to achieve this goal by concentrating on the changing nature of the threat landscape. It is imperative that this be done in order to protect them from any and all potential harm that may befall them. This thesis comes to a close with a thorough research of the application of behavior analysis to the detection of computer threats, with a special emphasis on the coupling of RF methods and RFE approaches as the primary area of focus throughout the whole of this inquiry. The findings of this study provide important new insights that might be used to the design and construction of computer threat detection systems that are more reliable and efficient. In addition to that, it contributes to the expansion of the field, which is an essential result of the research. Discovering these discoveries required coming up with novel methodologies in addition to bridging the study gaps that had been previously identified. The use of behavior analysis offers a significant amount of potential for enhancing cybersecurity and reducing the risks associated with risks associated with risks associated with risks associated with risks associated

with new and sophisticated assaults. This promise will come to fruition if more study in the field of behavior analysis is done.

6.2 Future Work

The following are some potential courses of action that we may pursue in the near or distant future: Even though we found out that RFE was successful in our investigation, it would still be beneficial for us to investigate other feature selection strategies, such as Lasso Regression, Principal Component Analysis (PCA), and Boruta, to determine whether or not we can further improve the performance of the model. Even though we found out that RFE was successful in our investigation, it would still be beneficial for us to research alternative feature selection strategies. Putting to the test many distinct machine learning frameworks: Even though the Random Forest classifier did a good job in our study, more machine learning methods should still be researched to see whether or not they are successful in the detection of computer security risks. This category may include things like Support Vector Machines (SVM), Gradient Boosting Machines (GBM), and Deep Learning models, to name a few examples. Building a system that can be modified to accommodate newly discovered information: In order to keep up with the ever-changing nature of the threats, identifying the dangers that may be caused by computers requires continuous innovation. It is conceivable that the model will need to be updated with new data, and the feature set may need to be re-evaluated over time using an adaptive learning framework in order to ensure that it will continue to be successful in spotting new threats. Both of these things will need to be done in order to guarantee that the model will continue to be effective. Including a wider variety of different forms of information: Incorporating data from a diverse range of sources, including system logs, user activity, and external threat intelligence feeds, for example, has the potential to boost the model's capacity to spot possible flaws in the security of a network. To further verify the generalizability of the method that we have described, it would be beneficial to analyze the performance of the model on a variety of intrusion detection datasets, such as those coming from different domains or including different kinds of network traffic. This would be done with the intention of determining whether or not the method can be applied to a wide range of situations. We have reason to think that if we examine these many areas of research,

we will be able to improve the accuracy and utility of computer threat detection models, which will assist us in our continued efforts to keep our digital surroundings safe and secure.

REFERENCES

[1] **P. Garcia-Teodoro, J. Diaz-Verdejo, G. Maciá-Fernández, and E. Vázquez,** "Anomaly-based network intrusion detection: Techniques, systems and challenges," Computers & Security, vol. 28, no. 1-2, pp. 18-28, 2019.

[2] **Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai**, "Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection," Network and Distributed System Security Symposium (NDSS), San Diego, CA, USA, 2020.

[3] **J. Pang, C. Li, L. Deng, H. Zhang, X. Chen, and Z. Tan,** "Unsupervised deep learning for anomaly detection in network traffic," IEEE Access, vol. 6, pp. 76900-76912, 2018.

[4] **S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie,** "High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning," Pattern Recognition, vol. 58, pp. 121-134, 2021.

[5] **W. Wang, Y. Sheng, J. Wang, X. Zeng, X. Ye, Y. Huang, and M. Zhu,** "Hadoop-based deep learning for intrusion detection in big data environment," Computers & Security, vol. 70, pp. 255-268, 2020.

[6] **M. Alazab, A. Alazab, S. Gauravaram, and P. Damodaran,** "Deep learning-based malware detection for Android applications: A systematic literature review," ACM Computing Surveys (CSUR), vol. 54, no. 5, Article 107, 2021.

[7] **LUFlow Network Intrusion Detection DataSet**: https://www.kaggle.com/datasets/mryanm/luflow-network-intrusion-detection-data-set

[8] **Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A**. (2010). A detailed analysis of the KDD CUP 99 data set. In Proceedings of the 2010 IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA) (pp. 1-6). IEEE.

[9] **Zhou, C., Cao, Z., Dong, X., Xing, C., & Han, J.** (2019). Deep learning for network intrusion detection: A technical review and taxonomy. IEEE Communications Surveys & Tutorials, 21(4), 3705-3731.

[10] **Janarthanan, T., & Zargari, S**. (2017). Intrusion detection using ensemble of soft computing paradigms. In Proceedings of the 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE) (pp. 1-4). IEEE.

[11] **Moustafa, N., & Slay, J**. (2016). UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In Proceedings of the 2015 Military Communications and Information Systems Conference (MilCIS) (pp. 1-6). IEEE.

[12] **Liao, H., Vemuri, V. R., & Wang, Q**. (2013). A wavelet-based statistical method for network anomaly detection. In Proceedings of the 2013 IEEE International Conference on Communications (ICC) (pp. 3729-3733). IEEE.

[13] **Li, K., Wu, Y., Li, K., Zhang, L., & Li, X**. (2018). An improved random forest-based intrusion detection system using SMOTE and feature optimization. In Proceedings of the 2018 IEEE 15th International Conference on Mobile Ad Hoc and Sensor Systems (MASS) (pp. 466-470). IEEE.

[14] **Somani, G., Gaur, M. S., Sanghi, D., Conti, M., & Rajarajan, M**. (2017). Combating DDoS attacks in the cloud: Requirements, trends, and future directions. IEEE Cloud Computing, 4(4), 22-32.

[15] **Roffo, G., Melzi, S., & Vinciarelli, A**. (2017). Infinite feature selection. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV) (pp. 4202-4210). IEEE.

[16] **Singh, K. K., Gupta, D., & Kumar, P.** (2018). Hybrid feature selection based weighted least squares twin support vector machine approach for diagnosing breast cancer, hepatitis, and diabetes. In Proceedings of the 2018 IEEE 8th International Advance Computing Conference (IACC) (pp. 726-731). IEEE.

[17] **Vinayakumar, R., Soman, K. P., & Poornachandran, P**. (2017). Applying convolutional neural network for network intrusion detection. In Proceedings of the 2017 IEEE International Conference on Advances in Computing, Communications and Informatics (ICACCI) (pp. 1222-1228). IEEE.

[18] **Garcia-Teodoro, P., Diaz-Verdejo, J., Maciá-Fernández, G., & Vázquez, E.** (2009). Anomaly-based network intrusion detection: Techniques, systems and challenges. Computers & Security, 28

[19] **Axelsson, S.** (2000). Intrusion detection systems: A survey and taxonomy. Technical report, Chalmers University of Technology, Department of Computer Engineering.

[20] **Zargar, S. T., Joshi, J., & Tipper, D.** (2013). A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks. IEEE Communications Surveys & Tutorials, 15(4), 2046-2069.

[21] **Hodo, E., Bellekens, X., Hamilton, A., Dubouilh, P. L., & Iorkyase, E. T**. (2016). Shallow and deep networks intrusion detection system: A taxonomy and survey. In Proceedings of the 2016 International Symposium on Networks, Computers and Communications (ISNCC) (pp. 1-7). IEEE.

[22] **Deng, L., & Yu, D.** (2014). Deep learning: Methods and applications. Foundations and Trends® in Signal Processing, 7(3–4), 197-387.

[23] **Alazab, M., Venkatraman, S., & Watters, P**. (2010). Towards understanding malware behaviour by the extraction of API calls. In Proceedings of the 2010 Second Cybercrime and Trustworthy Computing Workshop (CTC) (pp. 51-60). IEEE.

[24] **Breiman, L**. (2001). Random forests. Machine Learning, 45(1), 5-32.

[25] **Guyon, I., Weston, J., Barnhill, S., & Vapnik, V**. (2002). Gene selection for cancer classification using support vector machines. Machine Learning, 46(1-3), 389-422.

[26] **Han, J., Pei, J., & Kamber, M**. (2011). Data mining: Concepts and techniques. Elsevier.

[27] **Cano, A., Zafra, A., & Ventura, S**. (2013). An interpretable classification rule induction algorithm for imbalanced datasets. In Proceedings of the 2013 IEEE 25th International Conference on Tools with Artificial Intelligence (ICTAI) (pp. 167-174). IEEE.

[28] **Nalam Venkata Abhishek, Teng Joon Lim, Biplab Sikdar, and Anshoo Tandon.** 2018. An intrusion detection system for detecting compromised gateways in clustered IoT networks. In *2018 IEEE International Workshop Technical Committee on Communications Quality and Reliability (CQR)*. IEEE, 1–6.

[29] **João P Amaral, Luís M Oliveira, Joel JPC Rodrigues, Guangjie Han, and Lei Shu.** 2014. Policy and network-based intrusion detection system for IPv6-enabled wireless sensor networks. In *2014 IEEE International Conference on Communications (ICC)*. IEEE, 1796–1801.

[30] **Amar Amouri, Vishwa T Alaparthy, and Salvatore D Morgera**. 2018. Cross layer-based intrusion detection based on network behavior for IoT. In *2018 IEEE 19th Wireless and Microwave Technology Conference (WAMICON)*. IEEE, 1–4.

[31] **Luca Arnaboldi and Charles Morisset**. 2017. Quantitative analysis of dos attacks and client puzzles in iot systems. In *International Workshop on Security and Trust Management*. Springer, 224–233.

[32] **Luca Arnaboldi and Charles Morisset**. 2018. Generating Synthetic Data for Real World Detection of DoS Attacks in the IoT. In *Federation of International Conferences on Software Technologies: Applications and Foundations*. Springer, 130–145.

[33] **Luca Arnaboldi and Charles Morisset.** 2018. LISA: Predicting the Impact of DoS Attacks on Real-World Low Power IoT Systems. *Foundations of Computer Security Workshop* (2018).

[34] **Luca Arnaboldi and Hannes Tschofenig**. 2019. A Formal Model for Delegated Authorization of IoT Devices Using ACE-OAuth. In *OAuth Security Workshop*.

[35] **Harshal A Arolkar, Shraddha P Sheth, and Vaidehi P Tamhane.** 2011. Ant colony based approach for intrusion detection on cluster heads in WSN.. In *ICCCS*. 523–526.

[36] **Briana Arrington, LiEsa Barnett, Rahmira Rufus, and Albert Esterline**. 2016. Behavioral modeling intrusion detection system (bmids) using internet of things (iot) behavior-based anomaly detection via immunity-inspired algorithms. In *2016 25th International Conference on Computer Communication and Networks (ICCCN)*. IEEE, 1–6.

[37] **Stefan Axelsson.** 2000. The base-rate fallacy and the difficulty of intrusion detection. *ACM Transactions on Information and System Security (TISSEC)* 3, 3 (2000), 186–205.

[38] **Lionel Besson and Philippe Leleu.** 2009. A distributed intrusion detection system for ad-hoc wireless sensor networks: the AWISSENET distributed intrusion detection system. In *2009 16th International Conference on Systems, Signals and Image Processing*. IEEE, 1–3.

[39] **Hamid Bostani and Mansour Sheikhan.** 2017. Hybrid of anomaly-based and specification-based ids for internet of things using unsupervised opf based on mapreduce approach. *Computer Communications* 98 (2017), 52–71.

[40] **Jason Brownlee**. 2005. Clonal selection theory & clonalg-the clonal selection classification algorithm (csca). *Swinburne University of Technology* (2005), 38.

[41] **Timothy K Buennemeyer, Michael Gora, Randy C Marchany, and Joseph G Tront**. 2007. Battery exhaustion attack detection with small handheld mobile computers. In *Portable Information Devices*.

[42] **Timothy K Buennemeyer, Theresa M Nelson, Lee M Clagett, John P Dunning, Randy C Marchany, and Joseph G Tront**. 2008. Mobile device profiling and intrusion detection using smart batteries. In *Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008)*. IEEE, 296–296.

[43] **Ismail Butun, Salvatore D Morgera, and Ravi Sankar.** 2014. A survey of intrusion detection systems in wireless sensor networks. *IEEE communications surveys & tutorials* 16, 1 (2014), 266–282.

[44] **Leandro Nunes Castro, Leandro Nunes De Castro, and Jonathan Timmis**. 2002. *Artificial immune systems: a new computational intelligence approach*. Springer Science & Business Media.

[45] **Christian Cervantes, Diego Poplade, Michele Nogueira, and Aldri Santos**. 2015. Detection of sinkhole attacks for supporting secure routing on 6LoWPAN for Internet of Things. In *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. IEEE, 606–611.

[46] **Eung Jun Cho, Jin Ho Kim, and Choong Seon Hong**. 2009. Attack model and detection scheme for botnet on 6LoWPAN. In *Asia-Pacific Network Operations and Management Symposium*. Springer, 515–518.

[47] **Smita Chormunge and Sudarson Jena**. 2015. Efficiency and Effectiveness of Clustering Algorithms for High Dimensional Data. *International Journal of Computer Applications* 125, 11 (2015).

[48] **William W Cohen and Yoram Singer.** 1999. A simple, fast, and effective rule learner. *AAAI/IAAI* 99 (1999), 335–342.

[49] **Luigi Coppolino, Salvatore DAntonio, Alessia Garofalo, and Luigi Romano**. 2013. Applying data mining techniques to intrusion detection in wireless sensor networks. In *2013 Eighth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*. IEEE, 247–254.

[50] **B Craenen and A Eiben.** 2002. Computational intelligence. Encyclopedia of Life Support Sciences. *EOLSS, EOLSS Co. Ltd* (2002).

[51] **Jagan Mohan Reddy Danda and Chittaranjan Hota.** 2016. Attack identification framework for IoT devices. In *Information Systems Design and Intelligent Applications*. Springer, 505–513.

[52] **Lianbing Deng, Daming Li, Xiang Yao, David Cox, and Haoxiang Wang**. 2018. Mobile network intrusion detection for IoT system based on transfer learning algorithm. *Cluster Computing* (2018), 1–16.

[53] **Roberto Di Pietro and Luigi V Mancini**. 2008. *Intrusion detection systems*. Vol. 38. Springer Science & Business Media.

[54] **Sebastian Echeverria, Ludwig Seitz, Dan Klinedinst, and Grace Lewis**. 2019. *ACE Clients in Disadvantaged Networks*. Internet-Draft draft-secheverriaace-client-disadvantaged-00. Internet Engineering Task Force. https://datatracker.ietf.org/doc/html/draft-secheverria-ace-client-disadvantaged-00 Work in Progress.

[55] **Mohamed Faisal Elrawy, Ali Ismail Awad, and Hesham FA Hamed.** 2018. Intrusion detection systems for IoT-based smart environments: a survey. *Journal of Cloud Computing* 7, 1 (2018), 21.

[56] **Herson Esquivel-Vargas, Marco Caselli, and Andreas Peter.** 2017. Automatic deployment of specification-based intrusion detection in the BACnet protocol. In *Proceedings of the 2017 Workshop on Cyber-Physical Systems Security and PrivaCy*. 25–36.

[57] **Sandro Etalle**. 2017. From intrusion detection to software design. In *European Symposium on Research in Computer Security*. Springer, 1–10.

[58] San**dro Etalle.** 2019. Network Monitoring of Industrial Control Systems: The Lessons of SecurityMatters. In *Proceedings of the ACM Workshop on Cyber-Physical Systems Security & Privacy*. 1–1.

[59] **Yulong Fu, Zheng Yan, Jin Cao, Ousmane Koné, and Xuefei Cao**. 2017. An automata based intrusion detection method for internet of things. *Mobile Information Systems* 2017 (2017).

[60] **Victor Garcia-Font, Carles Garrigues, and Helena Rifà-Pous**. 2017. Attack classification schema for smart city WSNs. *Sensors* 17, 4 (2017), 771.

[61] **Pedro Garcia-Teodoro, Jesus Diaz-Verdejo, Gabriel Maciá-Fernández, and Enrique Vázquez**. 2009. Anomaly-based network intrusion detection: Techniques, systems and challenges. *computers & security* 28, 1-2 (2009), 18–28.

[62] **Abhishek Gupta, Om Jee Pandey, Mahendra Shukla, Anjali Dadhich, Samar Mathur, and Anup Ingle**. 2013. Computational intelligence based intrusion detection systems for wireless communication and pervasive computing networks. In *2013 IEEE International Conference on Computational Intelligence and Computing Research*. IEEE, 1–7.

[63] **Keijo MJ Haataja**. 2008. New efficient intrusion detection and prevention system for Bluetooth networks. In *Proceedings of the 1st international conference on MOBILe Wireless MiddleWARE, Operating Systems, and Applications*. ICST (Institute for Computer Sciences, Social-Informatics and ..., 16.

[64] **Dina Hadžiosmanović, Robin Sommer, Emmanuele Zambon, and Pieter H Hartel**. 2014. Through the eye of the PLC: semantic security monitoring for industrial processes. In *Proceedings of the 30th Annual Computer Security Applications Conference*. ACM, 126–135.

[65] **Guangjie Han, Jinfang Jiang, Wen Shen, Lei Shu, and Joel Rodrigues**. 2013. IDSEP: a novel intrusion detection scheme based on energy prediction in cluster-based wireless sensor networks. *IET Information Security* 7, 2 (2013), 97–105.

[66] **Amin Hassanzadeh and Radu Stoleru.** 2011. Towards optimal monitoring in cooperative ids for resource constrained wireless networks. In *Computer*

Communications and Networks (ICCCN), 2011 Proceedings of 20th International Conference on. IEEE, 1–8.

[67] **Caifeng Zou Jianqi Liu Hui Suo Jiafu Wan.** 2014. Security in the Internet of Things: A Review. *International Journal of Computer Applications* (2014).

[68] **Atzori Luigi Antonio Iera and Giacomo Morabito**. 2010. The Internet of Things: A Survey. *Computer Networks* (2010).

[69] **Tingyao Jiang, Gangliang Wang, and Heng Yu.** 2012. A dynamic intrusion detection scheme for cluster-based wireless sensor networks. In *World Automation Congress 2012*. IEEE, 259–261.

[70] **Lina Ge Kai Zhao.** 2013. A Survey on the Internet of Things Security. *Computational Intelligence and Security (CIS)* (2013).

[71] **Prabhakaran Kasinathan, Gianfranco Costamagna, Hussein Khaleel, Claudio Pastrone, and Maurizio A** Spirito. 2013. An ids framework for internet of things empowered by 6lowpan. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. ACM, 1337–1340.

[72] **Prabhakaran Kasinathan, Claudio Pastrone, Maurizio A Spirito, and Mark Vinkovits.** 2013. Denial-of-Service detection in 6LoWPAN based Internet of Things. In *2013 IEEE 9th international conference on wireless and mobile computing, networking and communications (WiMob)*. IEEE, 600–607.

[73] **Zeeshan Ali Khan and Peter Herrmann.** 2017. A trust based distributed intrusion detection mechanism for internet of things. In *2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA)*. IEEE, 1169–1176.

[74] **Sang Wu Kim.** 2015. Physical integrity check in cooperative relay communications. *IEEE Transactions on Wireless Communications* 14, 11 (2015), 6401–6413.

[75] **Christopher Kruegel and Giovanni Vigna**. 2003. Anomaly detection of web-based attacks. In *Proceedings of the 10th ACM conference on Computer and communications security*. ACM, 251–261.

[76] **Anhtuan Le, Jonathan Loo, Kok Chai, and Mahdi Aiash**. 2016. A specification-based IDS for detecting attacks on RPL-based network topology. *Information* 7, 2 (2016), 25.

[77] **Anhtuan Le, Jonathan Loo, Yuan Luo, and Aboubaker Lasebae**. 2011. Specification-based IDS for securing RPL from topology attacks. In *2011 IFIP Wireless Days (WD)*. IEEE, 1–3.

[78] **Tsung-Han Lee, Chih-Hao Wen, Lin-Huang Chang, Hung-Shiou Chiang, and Ming-Chun Hsieh**. 2014. A lightweight intrusion detection scheme based on energy consumption analysis in 6LowPAN. In *Advanced Technologies, Embedded and Multimedia for Human-centric Computing*. Springer, 1205–1213.

[79] **Wei-Chao Lin, Shih-Wen Ke, and Chih-Fong Tsai**. 2015. CANN: An intrusion detection system based on combining cluster centers and nearest neighbors. *Knowledge-based systems* 78 (2015), 13–21.

[80] **Caiming Liu, Jin Yang, Run Chen, Yan Zhang, and Jinquan Zeng**. 2011. Research on immunity-based intrusion detection technology for the internet of

things. In *2011 Seventh International Conference on Natural Computation*, Vol. 1. IEEE, 212–216.

[81] **Liqun Liu, Bing Xu, Xiaoping Zhang, and Xianjun Wu.** 2018. An intrusion detection method for internet of things based on suppressed fuzzy clustering. *EURASIP Journal on Wireless Communications and Networking* 2018, 1 (2018), 113.

[82] **Yang Liu and Fengqi Yu**. 2008. Immunity-based intrusion detection for wireless sensor networks. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*. IEEE, 439–444.

[83] **Emilie Lundin and Erland Jonsson**. 2000. Anomaly-based intrusion detection: privacy concerns and other problems. *Computer networks* 34, 4 (2000), 623–640.

RESUME

Mohammed Hazim Mutar MUTAR

EDUCATION:

- Master 2022 – 2023: Istanbul Gedik University, Engineering management master's degree.
- Bachelor 2012– 2013: Almaarif University College –Al-Anbar \Iraq, Computer Engineering BSC Full B.sc in this field

LANGUGES:

- Arabic : Mother Tongue
- English: Proficient in Speaking and Writing.

SKILLS:

Ability to work under pressure, work with a team, positive attitude, self- directed and confident decision maker, strong work ethic, ability to prioritize, multitasked and exceptional management.

COMPUTER SKILLS:

- Computer Use
- Internet User
- Emails
- Microsoft Word
- Microsoft access
- Microsoft PowerPoint

WORK AND EXPERIENCE:

Ministry of Construction, Housing, Municipalities and Public Works From 2009 to noew