Machine Learning and Knowledge-Based Integrated Intrusion Detection Schemes

Yu Shen

Thesis submitted to the University of Ottawa
in fulfillment of the requirements for the
Master of Applied Science in
Electrical and Computer Engineering

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

As electronic computer technology advances, files and data are kept in computers and exchanged through networks. The computer is a physically closed system for users, making it harder for others to steal data via direct touch. Computer networks, on the other hand, can be used by hackers to gain access to user accounts and steal sensitive data. The academics are concentrating their efforts on preventing network attacks and assuring data security. The Intrusion Detection System (IDS) relies on network traffic and host logs to detect and protect against network threats. They all, however, necessitate a lot of data analysis and quick reaction tactics, which puts a lot of pressure on network managers. The advancement of AI allows computers to take over difficult and time-consuming data processing activities, resulting in more intelligent network attack protection techniques and timely alerts of suspected network attacks.

The SCVIC-APT-2021 dataset which is specific to the APT attacks is generated to serve as a benchmark for APT detection. A Virtual Private Network (VPN) connects two network domains to form the basic network environment for creating the dataset. Kali Linux is used as a hacker to launch multiple rounds of APT attacks and compromise two network domains from the external network. The generated dataset contains six APT stages, each of which includes different attack techniques. Following that, a knowledge-based machine learning model is proposed to detect APT attacks on the developed SCVIC-APT-2021 dataset. The macro average F1-score increases by 11.01% and reach up to 81.92% when compared to the supervised baseline model. NSL-KDD and UNSW-NB15 are then utilized as benchmarks to verify the performance of the proposed model. The weighted average F1-score on both datasets can reach 76.42% and 79.20%, respectively.

Since some network attacks leave host-based information such as system logs on the network devices, the detection scheme that integrates network-based features and host-based features are used to boost the network attack detection capabilities of IDS. The raw data of CSE-CIC-IDS2018 [115] is utilized to create the SCIVC-CIDS-2021 dataset which includes both network-based features and host-based features. To ensure precise classification results, the SCVIC-CIDS-2021 is labelled with the attacking techniques. Due to the high dimensionalities of the features in the produced dataset, Autoencoder (AE) and Gated Recurrent Unit (GRU) are employed to reduce the dimensionality of network-based and host-based features, respectively. Finally, classification of the data points is performed using knowledge-based PKI and PKI Difference (PKID) models. Among these, the PKID model performs better with a macro average F1-score of 96.60%, which is 7.62% higher than the results only utilizing network-based features.

Acknowledgements

First and foremost, I would like to thank Dr. Burak Kantarci and Dr. Hussein Mouftah, my supervisors. They provide altruistic assistance and rigorous instruction when I am pursing the master degree. I am strongly influenced by their diligent work ethic. They responds to me as soon as possible if there is an issue, whether it is late at night or on a holiday. I learn a lot from our discussion, and they can always point me in the right direction if I run into a snag. Without their help, I would not have gone so far on the my academic career.

Next, I would like to say thank you to Dr. Murat Simsek, Jinxin Liu, and Zhiyan Chen. They provide me with materials and resources that help me avoid several pitfalls in my machine learning studies.Therefore, thank them again for their encouragement and support. In addition, I would also like to thank all the other members of the laboratory who worked together with me, Nima Taherifard, Ahmed Omara, Nahid Parvaresh, Yuwei Wang, Jichu Jiang, Bin Xiao, Ceren Comert, Johan Fernandes, Samhita Kuili, Yakup Akkaya, Mohsen Dastjerdi.

Of course, I want to express my gratitude to Dr. Petar Djukic and Dr. Mehran Bagheri. We are able to make progress based on the original achievements thanks to their assistance on collaboration initiatives. At the same time, working closely with industry gives me a sense of Canada's bustling information industry and academia. As a result, I value this opportunity much and consider myself lucky to be able to exchange research ideas with predecessors.

Finally, I would like to express my gratitude to my parents and my wife, Lu Li, for their unwavering support for my study. Even though I am far away from them, I can sense their high expectations for me. We went through the torment of Covid-19 together while pursuing my master's degree, which brought our hearts closer. I will utilise facts to demonstrate that I will not let them down and will accomplish excellent success in my career and life in the future.

Table of Contents

List of Figures

List of Tables

Abbreviations

5G 5th Generation of Communication 1, 3

AE Autoencoder 58–62, 64, 69, 71

AI Artificial Intelligence 1

ANN Artificial Neural Network 23

ANOVA Analysis of Variance 42, 43, 46

APT Advanced Persistent Threat 6–9, 14, 15, 17, 19, 22, 25–28, 30, 31, 35, 37–39, 41, 42, 50, 68–70

ATCTDS Automatic Temporal Correlation Traffic Detection System 19

BERT Bidirectional Encoder Representations from Transformers 57, 58

BiLSTM Bidirectional Long-Short Term Memory 19

C2 Command and Control 36, 37

CNN Convolutional Neural Network 24

CNN-LSTM Convolutional Neural Network-Long Short Term Memory 24

DAE Deep Autoencoder 23

DBSCAN Density-based Spatial Clustering of Applications with Noise 25

DC Domain Controller 32, 34, 35, 37, 39

DDoS Distributed Denial of Service 13

DE Data Exfiltration 7, 28, 31, 35–37, 39

DNN Deep Neural Network 23

DNS Domain Name System 7, 30, 36, 37

DoS Denial of Service 7, 41

DT Decision Tree 21, 22, 49, 64, 65

FCN Fully Connected Network 23, 24

FTP File Transfer Protocol 31, 41

GCN Graph Convolutional Network 19

GMM Gaussian Mixture Model 5, 49

GRU Gated Recurrent Unit 58, 59, 61, 62, 64, 65, 69, 71

HIDS Host-based Intrusion Detection System 2, 3, 7, 8, 55, 68

HMI Human Machine Interface 12

IDS Intrusion Detection System 2, 3, 6, 7, 9, 11, 17, 19, 23, 26–28, 30, 55, 68, 70

IoT Internet of Things 15, 19, 22, 23

K-Means K-Means Unsupervised Algorithm 5, 21, 25, 49, 65–67

KNN K-Nearest Neighbor 22

LAN Local Area Network 13, 28

LM Lateral Movement 7, 17, 19, 20, 28, 30–33, 35, 37–39

LR Linear Regression 23

LSTM Long Short Term Memory 24, 61

MI Mutual Information 43

ML Machine Learning 3, 5, 8, 9, 11, 25, 27, 41

MLP Multi-layer Perceptron 19, 23

NB Naive Bayes 22

NIDS Network-based Intrusion Detection System 2, 3, 7, 8, 55, 68

NLP Natural Language Processing 57

OS Operating System 32, 33

PKI Prior Knowledge Input 1, 9, 27, 42, 45, 46, 49, 50, 56, 58, 59, 62–65, 69

PKID Prior Knowledge Input Difference 9, 56, 63–65, 67, 69

PLC Programmable Logic Controller 12

R2L Remote to Local 17, 22, 41

RDP Remote Desktop Protocol 7, 13, 19, 20, 30, 32

RF Random Forest 21, 22, 64, 65

RNN Recurrent Neural Network 24

Seq2Seq Sequence to Sequence 24

SVM Support Vector Machine 21, 23

U2R User to Root 15, 22, 41

UML Unified Modeling Language 20

VAE Variational Autoenco 23, 24

VAE-FCN Variational Autoencoder-Fully Connected Network 24

VPN Virtual Private Network 30, 37

VSFTPD Very Secure FTP Daemon 31

WinRM Windows Remote Management 33

WMI Windows Management Instrumentation 32, 33

WSS Within-cluster Sum of Square 65, 66

Chapter 1

Introduction

According to the Kepios's report [4], there are around 4.6 billion Internet users globally in 2021 and the number of social media users among them exceeds 4.2 billion. The actual numbers may be greater owing to the influence of COVID-19. These data demonstrate the Internet's continued rise in the post-epidemic age, as well as people's growing reliance on it in their everyday lives. However, the number of network attacks encountered by Internet users worldwide climbed by 29% in 2021 compared to the previous year [1]. As a result, knowing how to maintain network security and protect one's critical interests becomes mandatory training for all Internet users. Combating and detecting network attacks becomes hot topics among governments and businesses.

Artificial Intelligence (AI) has attracted increasing attention in recent years as computing power has increased, and it has been effectively implemented in a variety of fields [37]. Applying AI to protect network security has become one of the trendiest subjects for researchers to investigate [71]. This thesis seeks to identify network attacks using the proposed Prior Knowledge Input (PKI) model based on machine learning. Network-based and host-based features are integrated to boost the model's detection performance against network attacks.

1.1 Background

With the arrival of the 5th Generation of Communication (5G) era, the network's popularity soar to new heights. People are becoming more reliant on the conveniences that the Internet provides, such as video chat and online shopping. Simultaneously, in order to

simplify communication, the computers transmit and store a large amount of critical data. The stressful period of mailing letters and waiting for answers is now gone, as new information can reach any part of the globe in an instant via the Internet. However, because of the network's simplicity and speed, many users are likely to disclose sensitive information and data to it without taking any measures. As a result, the frequency of data breaches rises year after year. According to the RiskBased Security, data breaches exposed 7.9 billion records in the first nine months of 2019 [2]. This is more than double the amount of records disclosed in the same time period in 2018. International Data Corporation states that the global spending on network security solutions would reach $133.7 billion by 2022 [3]. Governments around the world provide guidance to businesses, organizations and individuals in responding to increasing network threats to help them prevent data leakage and network attacks.

1.1.1 Intrusion Detection System

An Intrusion Detection System (IDS) is a defence mechanism that monitors unusual activities and network attacks that occur during normal computer usage and alerts network administrators when dangerous events happen [110]. IDS is categorized into Network-based Intrusion Detection System (NIDS) and Host-based Intrusion Detection System (HIDS) based on the difference in function and monitoring content.

NIDS detects suspicious behaviour and network threats by inspecting the header and content of every network traffic packets in the domain [121]. To evaluate network traffic, NIDS is implemented on critical nodes in the network. For instance, NIDS can be placed on the subnet's interface to interact with the firewall to guard against network threats.

The system configuration and application actions of computers in the domain are detected and analyzed by HIDS. Unlike NIDS, HIDS can be installed on any network device, whether it is a PC or a server [63]. When HIDS detects unusual changes, such as overwriting, file deletion, or port opening, it alerts the administrator. As a result, when confronted with threats inside devices, HIDS is an effective defence mechanism.

Both IDS have its own set of advantages. The differences of NIDS and HIDS is demonstrated in Table 1.1. NIDS can efficiently intercept network threats from outside network and identify real-time network traffic. However, when an internal network assault occurs, NIDS's effectiveness is limited. HIDS is able to detect attacks from the internal network by comparing historical configuration snapshots of the system. This means that HIDS needs historical information as a reference to determine whether recently generated events logs are malicious. The attack speed of HIDS is slower than NIDS. The two IDS are not in a

Table 1.1: The difference between NIDS and HIDS

	NIDS	HIDS
Location	Crucial Point	Any Device
Content	Network Packet	Host Event
Attack Source	External Attacks	Interal Attacks
Time	Real-time	Historical-time

competitive relationship but complement and cooperate with each other. The relationship between them also inspired the proposed network-based and host-based features integrated detection method. The combined IDS structure, NIDS structure and HIDS structure are illustrated in Fig. 1.1. When a network attack originates from the external network, the NIDS in the network domain interface establishes the initial barrier against the malicious packets. If an attack penetrates the internal network, the HIDS installed on user devices assess whether or not damaging events occur and generate event logs for further examination. On the contrary, if the attackers send out packets from the internal network, the HIDS monitor network ports and request to block illegal communication. Then, the NIDS utilize packets header and payload to prevent data exfiltration.

With the rapid development of security technology, the market puts forward higher requirements for IDS's detection performance and defence scheme.

First, the capacity of IDS to distinguish between normal and malicious traffic must be increased. In a typical network, normal traffic accounts for the vast majority of traffic, whereas malicious traffic accounts for only a small portion. Users' experience will be harmed if IDS incorrectly classifies normal traffic as malicious traffic. Second, the increasing network bandwidth in the 5G era substantially increases the information transfer rate, putting more demands on IDS timeliness [111]. One of the targets sought by researchers is to accomplish real-time monitoring with multiple times the quantity of data. Third, some network attacks has a long incubation period. To put it another way, when hackers get administrator privileges on an internal computer, they are not eager to take detrimental acts right away, but rather wait for the perfect chance to strike when they need it most. This necessitates IDS taking action against the threat in the early stage of the attack [108].

1.1.2 Knowledge-based Machine Learning Method

Machine Learning (ML) is the study of how computers mimic or execute human learning processes in order to acquire new information or abilities and rearrange existing knowledge

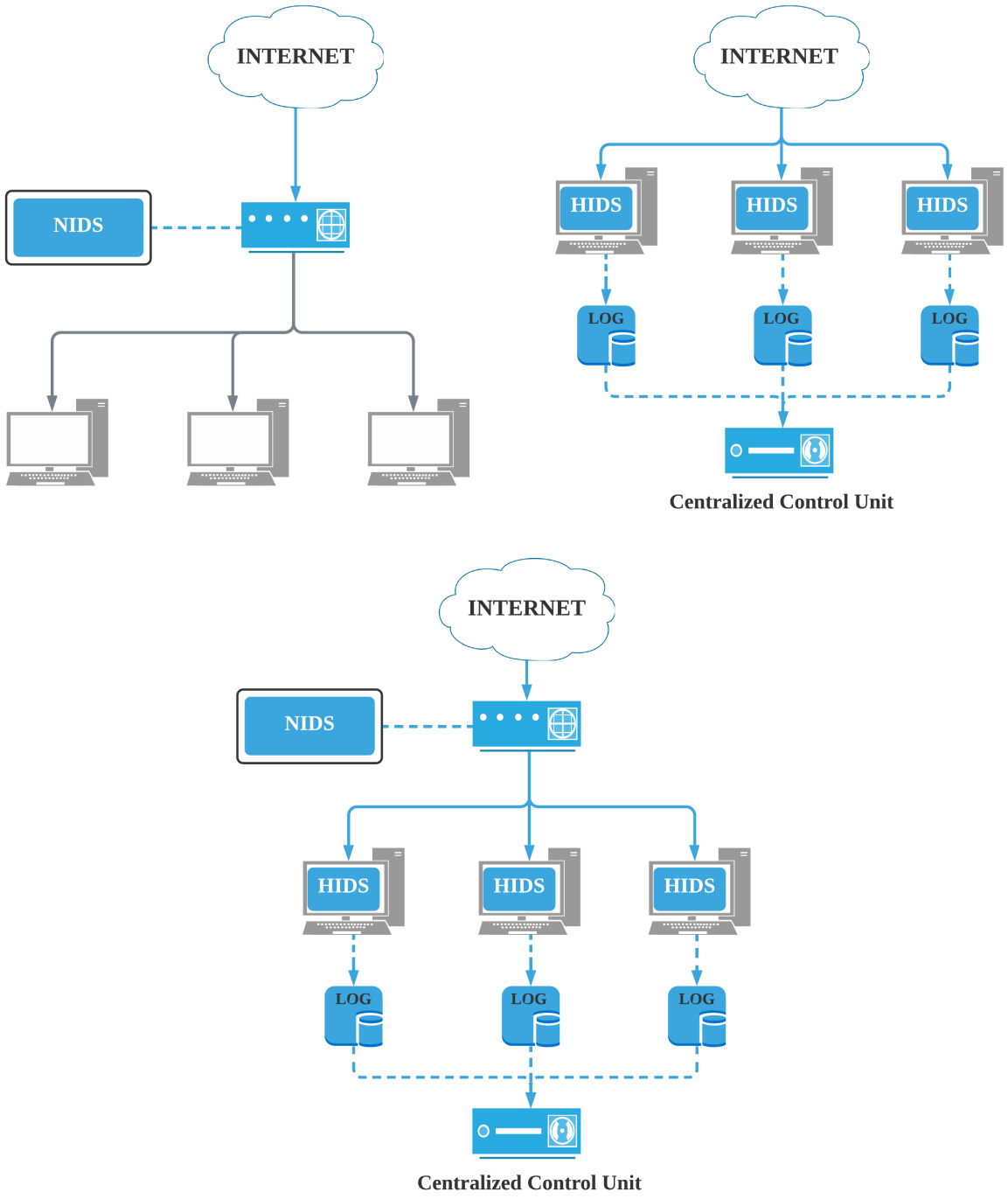


Figure 1.1: NIDS, HIDS and integrated IDS structures

structures in order to continually improve their own performance. It is the foundation of AI and the means through which computers acquire intelligence. ML advances significantly with the development in computing power, starting with a learning machine constructed by Alan Turing in the 1950s [87]. ML becomes an integral element of people's daily lives since the dawn of the digital era.

The purpose of ML is to create a precise mapping connection in order to address a specific problem. Despite the numerous cross-applications of ML in diverse fields, they may all be classified as practical applications of regression and classification, the two fundamental domains of ML [53]. The primary distinction between these two categories is whether the data values are continuous or discrete. It is a classification problem if the data is discrete. Otherwise, it is a regression problem. This thesis investigates the identification of network attacks as a typical classification problem. Network flows and host events are utilized to extract numerical features, which are then classified using ML algorithms to see if the flows or events represent network attacks. Network attack techniques are used as the data point's labels if it requires a more precise classification.

Datasets with a limited quantity of data typically require basic data cleansing before being fed to ML algorithms. When presented with a huge dataset, ML algorithms may find it challenging to build an accurate relationship mapping function directly from the original data. As a result, before the data is sent to the ML algorithm, feature engineering is used to select and analyze features in order to improve the algorithm's decision-making ability. Furthermore, [149] proposes the knowledge-based model to decrease training complexity.

The knowledge-based model's main purpose is to extract knowledge from the original data and incorporate it into the training and testing process in order to decrease training complexity. After obtaining the knowledge, the ML algorithm may achieve the same or even higher model performance with less data [123]. As a result, the knowledge-based method expands on ML's potential and improves on the original results. Since it may potentially suit any function without paying attention to the underlying mathematical logic, the neural network is regarded one of the best methods for obtaining knowledge from a dataset [127]. The input is fed to the knowledge layer to obtain knowledge as well as the fully connected layer during the forward propagation of the neural network. The gained knowledge then is used as one of the references for neural network decision-making. Fig.1.2 provides a schematic diagram of the knowledge-based neural network structure.

Unsupervised algorithms, in addition to neural networks, can be employed as a pre-classifier to offer knowledge for the dataset [125]. Unsupervised clustering methods such as K-Means Unsupervised Algorithm (K-Means), Gaussian Mixture Model (GMM), and others are used in this study to offer pre-classification results as the prior knowledge for

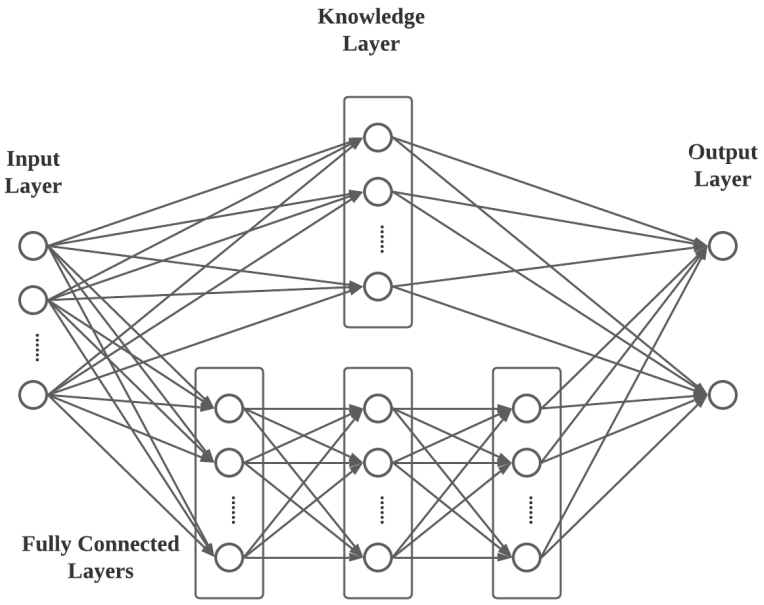


Figure 1.2: Knowledge-based neural network structure

attack detection and improve IDS attack detection performance.

1.2 Motivations

According to previous discussions, the global Internet has grown in the post-epidemic age, and the number of Internet and social media users is on the rise. The Internet evolves into one of the key drivers of social and economic progress. However, the deteriorating network security condition becomes a significant impediment to the Internet's development. Advanced Persistent Threat (APT) attacks, in particular, which rise in recent years, posed significant concealed threats to businesses, governments, and even national network security. As a result, the focus of this thesis is on preserving network security, data encryption, and user privacy.

After reviewing a large amount of literature on network security and IDS, it is discovered that most researchers use out-of-date datasets for network flow analysis and testing, such as NSL-KDD and UNSW-NB15 [36] [59]. Despite the fact that these datasets are commonly used as benchmarks for detecting network attacks, the network traffic and attack

types contained within them do not properly match the architecture and traffic patterns of today's networks [84]. At the same time, the majority of today's popular datasets are still concentrate on detecting early network attack techniques like Denial of Service (DoS) and Flooding. The form of these network attacks is relatively basic, and they can be detected with high accuracy using basic algorithms. In terms of model performance, there is not much opportunity for improvement. Because of its complicated and varied attack methodologies and long-term latency, the APT attack places increased demands on the IDS system's detection capabilities. Some APT attack detections focus on just one attack technique, such as Remote Desktop Protocol (RDP)-based Lateral Movement (LM) [18] and Domain Name System (DNS)-based Data Exfiltration (DE) [8]. Although these detection methods are efficient and accurate, it is impossible to explain an entire APT attack chain using a single attack approach, which is insufficient for detecting complex and variable APT attacks. Some researchers are also working on figuring out the entire APT attack chain [160]. The APT attack path of hackers in the internal network is predicted using graph analysis and the Markov chain [75]. Although these algorithms could predict the whole APT attack chain, the model detection effectiveness is poor when there is a variation between the anticipated attack chain and the actual attack path. Based on the preceding analysis, one of the most pressing concerns in the network security field is how to thoroughly and effectively identify all attack techniques and stages in the APT attack chain.

Although NIDS and HIDS are extensively utilized in intranet defence as complementary domain defence tactics, their processes do not conflict with one other, according to Section 1.1.1's research evaluation. In other words, NIDS monitors network flows entering and exiting the network interface, whereas HIDS focuses on the system's event logs and configuration snapshots. When an intranet device gets network attack traffic, the system's internal settings are likely to be changed, leading in the loss of root privilege and sensitive data. As a result, there is a link between the traffic of the network attack and the logs kept in the system. Previous researchers overlooks this crucial detail. As a result, one of the main aims of this thesis is to integrate network-based flows and host-based event logs to improve the model's performance in identifying network attacks.

There is no relevant study indicating how the knowledge-based model works in the subject of network security classification, despite the fact that it performs well in other classification and regression disciplines [89]. Furthermore, the scability and resilience of the knowledge-based model can be examined due to the complexity of network data, particularly the extended latency and various attack strategies of the APT attack. The performance of knowledge-based models in network security is also a focus of this research in order to cover the gap of knowledge-based models in this field.

1.3 Objectives

The goal of the research is to develop a method that is both efficient and accurate for detecting attacks in vast volumes of data and complicated network environments. First, the proposed knowledge-based model is capable of detecting APT attacks at various phases as well as clearly identifying the attack techniques used by attackers. When performing data labelling, it is critical to determine the relevant attack techniques based on the IP address and attack timeline as much as feasible. Furthermore, when confronted with the dataset imbalance problem, the model must overcome bias and variance in order to produce better results.

Second, when faced with a dataset that mixes network-based and host-based features, the knowledge-based model's performance should outperform that of employing only one of the two features. This necessitates the model discovering the link between network-based and host-based properties, rather than considering them as two separate pieces, as NIDS and HIDS.

1.4 Contributions

A dataset for APT attacks is created after research into the cycles and strategies of APT attacks. The collaboration between NIDS and HIDS inspired the creation of a hybrid dataset comprising network-based and host-based features. In benchmark datasets and laboratory-generated datasets, the knowledge-based ML model is proposed to distinguish network attack data points from normal data points.

The details are demonstrated as follows.

- **SCVIC-APT-2021 Dataset Generation**: The first contribution of this study is the development of an APT-specific network intrusion dataset. Multiple rounds of APT attacks are implemented within the simulated internal network of the testbed, which comprises two network domains. The SCVIC-APT-2021 dataset is assembled by extracting network flows from various devices. Since most APT datasets have privacy concerns, the generated SCVIC-APT-2021 could serve as a baseline for ML-based APT detection techniques.
- **Network Flows and Host Events Integrated Intrusion Detection Scheme**: The thesis' second contribution is the proposal of an unique network intrusion detection technique that integrates network flows and host events to improve network

attack detection performance. In most IDS, network traffic and host events are considered differently. However, network flows may create host events on compromised devices as a result of the attack, which might lead to information loss if the IDS ignores the relationship between network flows and host events. The proposed method combines the two components into a single data point to improve intrusion detection performance.

- **Prior Knowledge Input Model for Detection Improvement**: To improve the data points categorization outcomes, the PKI model is incorporated, which supplies the ML-based intrusion detection algorithms with previous knowledge. To decrease training complexity and processing time, unsupervised approaches in PKI transmit pre-classification results to the final decision-making classifier. The PKI model is considered to achieve the highest decision-making classifier performance.

1.5 Thesis Structure

This thesis is divided into five chapters. The Chapter 2 discusses the current state of network attacks and conducts a literature review. In Section 2.1, we discuss the taxonomy of common network attacks and the APT attacks which are one of the subjects of this thesis. We demonstrate signature-based and anomaly-based network attack detection schemes in Section 2.2. Section 2.3 classifies and discusses several machine learning-based detection approaches. Finally, the gap analysis between previous researches and this study are illustrated in Section 2.4.

Chapter 3 discusses a strategy for detecting APT attacks based on network traffics. We describe the datset utilized of the work in Section 3.1. The dataset, called SCVIC-APT-2021, for APT attacks is produced. NSL-KDD and UNSW-NB15 are used as benchmarks to evaluate the proposed model's performance. Section 3.2 presents a knowledge-based PKI model for detecting APT network attacks. The experiment's results and conclusions are explained in Sections 3.3 and 3.4, respectively.

Chapter 4 integrates network-based and host-based features to identify network attacks. In Section 4.1, the CICIDS 2018 dataset's raw data is utilized to create the SCVIC-CIDS-2021 that satisfies the experiment goals. We proposed PKI and Prior Knowledge Input Difference (PKID) models in Section 4.2 for detecting network attacks in the generated hybrid datasets. Sections 4.3 and 4.4 detail the proposed model's performance and experimental results.

Finally, in Chapter 5, we summarize the methodologies and results from this thesis and suggest future directions in related fields.

Chapter 2

Related Work

With the advancement of electronic computers and the Internet, a growing number of vital data files are digitized, automatically processed, and stored locally or delivered over the Internet. Through networking and digital technologies, governments and many emerging firms have dramatically enhanced office productivity and economic advantages. Malicious hackers, on the other hand, may attempt to steal governmental or business secrets for nefarious objectives [42]. The internal network's security is put to the ultimate rigorous test at this moment. This is also why the field of information and network security has gotten a lot of attention. The firewall is an important protection mechanism in the classic network attack detection mode. It accomplishes its goal of network security by preventing illegal traffic from entering and exiting specific network ports. Firewalls, on the other hand, cannot totally protect against hackers' attacks when faced with novel vulnerabilities or when attackers utilize authorized ports. IDS, as a new protection mechanism, monitors network traffic or host events to compensate for firewall weaknesses [102]. As a result, in order to set the direction for the thesis, this chapter reviews and evaluates the evolution of IDS and detection methods in recent years.

At the beginning of this chapter, the network attacks are classified into six categories in order to offer an overall picture. Next, classification criteria such as signature-based detection schemes and anomaly-based detection schemes are employed to identify contemporary network attack defensive tactics. Finally, ML-based method is discussed as a new form of the intrusion detection system in recent years.

2.1 Network Attack Taxonomy

This section describes an attack taxonomy in the literature based on operational impact and attacks in different network layers. The literature in this area is typically driven by the availability of datasets. Various datasets are used in many of the papers examined in this thesis, consequently leading to different standards to classify attack types. To address the lack of a common dataset framework requires a unified attack taxonomy to summarize all threats and validate the efficiency of using machine learning algorithms.

2.1.1 Common Network Attacks

Networks are an essential asset in a business [104]. Adversaries take advantage of cybersecurity weaknesses in networks to 1) obtain confidential information, 2) disrupt operations, and 3) cause damage potentially to the reputation and interests of an enterprise under attack.

Among the network vulnerabilities [7], the simplest one is that an attacker can directly send commands to data acquisition equipment. Since most Programmable Logic Controller (PLC) and data acquisition servers lack basic authentication, they generally accept any information sent to them. All of the adversaries aim to establish a connection with the device via breaking in through the firewall and then send appropriate commands to take control.

Another effective attack method for the adversaries is to export the operator's Human Machine Interface (HMI) directly to an external device. Off-the-shelf tools can complete this operation under Windows and UNIX environments. Finally, the adversary can use a Man-in-the-Middle attack in which the attacker plays the role of a middleman. If the attacker knows to communication protocol, they can change data packets between the HMI and the server to both deceive the operator and take control of the network[39].

Below is a list of some of the attacks in the network security literature:

- **Zero-day Attacks**: Zero-day exploits refer to the time it takes from the detection of an attack to fix the design flaws that developers have not recognized the time of release and until the vulnerability is overcome [22]. Such attacks may occur on field devices and servers. For example, using overflowing buffers, an adversary can inject malicious executable code into running programs in order to take command of various industrial processes.

- **Non-prioritization of Tasks**: Non-prioritization of tasks is a severe flaw in many types of real-time Local Area Network (LAN) [17]. As an example, memory sharing among the tasks with equal priority results in serious security issues. Accessibility to create an Object Entry Point in the kernel domain is a feature of VxWorks, a proprietary real-time operating system, resulting in loopholes.

- **Database Injection**: Database injection exploits the vulnerabilities in an internal network [21], which widely occur in SQL-based databases. The hackers send fake SQL commands from a web server to a database server, for example, via a supervisory control and data acquisition system, subsequently obtaining unwarranted access to both data and digitally controlled the network devices.

- **Vulnerabilities in Legacy Systems**: In association with a legacy system, networks usually lacks sufficient user and system authorization, and data authorization verification [32], leading to attackers gaining unauthorized access to the system. When faced with a new generation of infrastructure, older devices and the associated communication protocols cannot be encrypted. Attackers can discover usernames and passwords by sniffing software that allows them to monitor users' internet traffic in real-time, capturing all the data flowing to and from the users' computer.

- **Default Setting**: Through enumeration methods, an attacker can easily crack unsafe devices with default passwords and settings [6], and then control the network devices and implement a corresponding network attack.

- **Remote Access Policy**: When a user is connected to an unaudited dial-up line or a remote login server with remote control system, like RDP, an attacker can easily access the interal network inside the enterprise through the back door [7].

- **DDoS Attacks**: Invalid sources and limited access control allow attackers to drain servers by performing Distributed Denial of Service (DDoS) attacks on vulnerable unpatched systems [109]. DDoS attacks can be precisely divided into two forms: bandwidth consumption type and resource consumption type. They all require a significant amount of network and equipment resources through many legal or forged requests to achieve the purpose of paralyzing the network and the system.

- **Web Application Attacks**: Network users are increasingly connected to networks and accessible anywhere via a web-interface. As a result, unprotected systems are vulnerable to cross-site scripting and SQL injection attacks [106]. Web applications are increasingly used in cyber attacks and cyber crimes due to the popularity of cloud services.

2.1.2 Advanced Persist Threat

Individuals were the major targets of hackers in the early days of the Internet. Network attackers steal internet account passwords, mess with system data, and disrupt people's life. The breadth of the early network attacks was limited, and the damage was not huge. However, as technology advances and the internet becomes more widely used, hackers are able to devise complete strategies and employ complicated technology chains to carry out large-scale, high-harm attacks. The APT attacks, which has progressively gained traction in recent years because to its stealth and long-term threat, is one of them. APT targets, unlike network attacks on individuals, are frequently huge organizations like governments, scientific research institutions, and businesses. For instance, Stuxnet successfully attacked the industrial control system placed on the physically isolated intranet in 2011. After years of planning and latency, it delayed Iran's nuclear development. Throughout 2018, the APT28 organization utilized the attack technology of Office templates to inject remote macro documents into foreign affairs organizations in North American, European countries, and government entities in the former Soviet Union to conduct targeted attacks.

As the name of the APT attack appears, it contains three distinctive characteristics.

- **Advanced**: The APT attack is advanced since the attackers do not follow a determined attack strategy. To begin, they do research and analysis on the target network, develop the most effective attack strategy for the network, then attack using any network vulnerabilities and computer backdoors they control over. In addition to attacks, social engineering and psychological approaches, such as social media investigating and phishing emails, are used to unwittingly lead targeted network users into traps and become a link in the attack chain.

- **Persistent**: The APT attack typically lasts months or even years from the start of the attack until the attackers find the desired information. Hackers have plenty of time to locate and examine the target network, as well as change their attack strategy to the current scenario in order to mask their tracks more effectively. At the same time, the attackers' payload and traffic is tiny in comparison to a vast volume of normal traffic, making traditional defence techniques like network administrators and firewalls difficult to detect.

- **Threat**: The APT attack is tough for a single person to perform because it is exceedingly intricate and needs the initiator to learn a huge variety of attack methodologies and computer theories. The APT attack is always directed at the financed attacker

organization. Their goal is clear, and they typically have the support of the organization or the country. Following the theft of the target information, the hackers execute follow-up processing, such as deleting traces, making it harder to conduct investigations, and the corporation or country suffer significant damages.

From the above characteristics, it is clear that the APT attack integrates the currently known attack methods, and according to the customized attack strategy, it infringes the target network for a long time, which is highly secretive and harmful. As a result, one of the goals of this thesis's study is to create a new network traffic detection dataset for the APT attack and feed it into the proposed PKI model to identify the APT attack flows in network traffic.

2.1.3 Attacks Taxonomy

Operational impact [122], a cyber-attack taxonomy methodology, is utilized to organize attack types (Fig. 2.1). Meanwhile, attack with red denotes is a type of APT attacks. Attacks under user compromise, install malware, web compromise and root compromise all belongs to the APT group. These attacks are commonly distributed in free network datasets, such as NSL-KDD, KDD99, UNSW-NB15, MAWLab CICIDS2018 and AWID. Although we show each attack type as independent, an attacker would typically combine multiple of these attacks overtime to accomplish their final goal. The types of operational impact are:

- **Denial of Service** – aim to paralyze a target system and its related devices by flooding or crashing them and making them unable to provide services for their intended purpose by flooding or crashing them [27]. Typical DoS attack techniques include TCP SYN flood, UDP flood, and ICMP flood, while in the Internet of Things (IoT) context, DoS can extend to blackhole, sinkhole, gray hole, hello flood, and Clone ID attacks. Most DoS techniques flood vast amounts of traffic or data into the system; however, the features of a DoS attack are relatively easily extracted and identified compared with other types of attacks. According to the number of packets in a certain period, machine learning can determine a threshold automatically to raise attack alerts. As a result of its significant features, not only supervised machine learning algorithms but also unsupervised ones like clustering can recognize such attacks.
- **Root Compromise** – is about gaining unauthorized privileges of an administrator on a particular host [67]. User to Root (U2R) is a class of attack that compromises

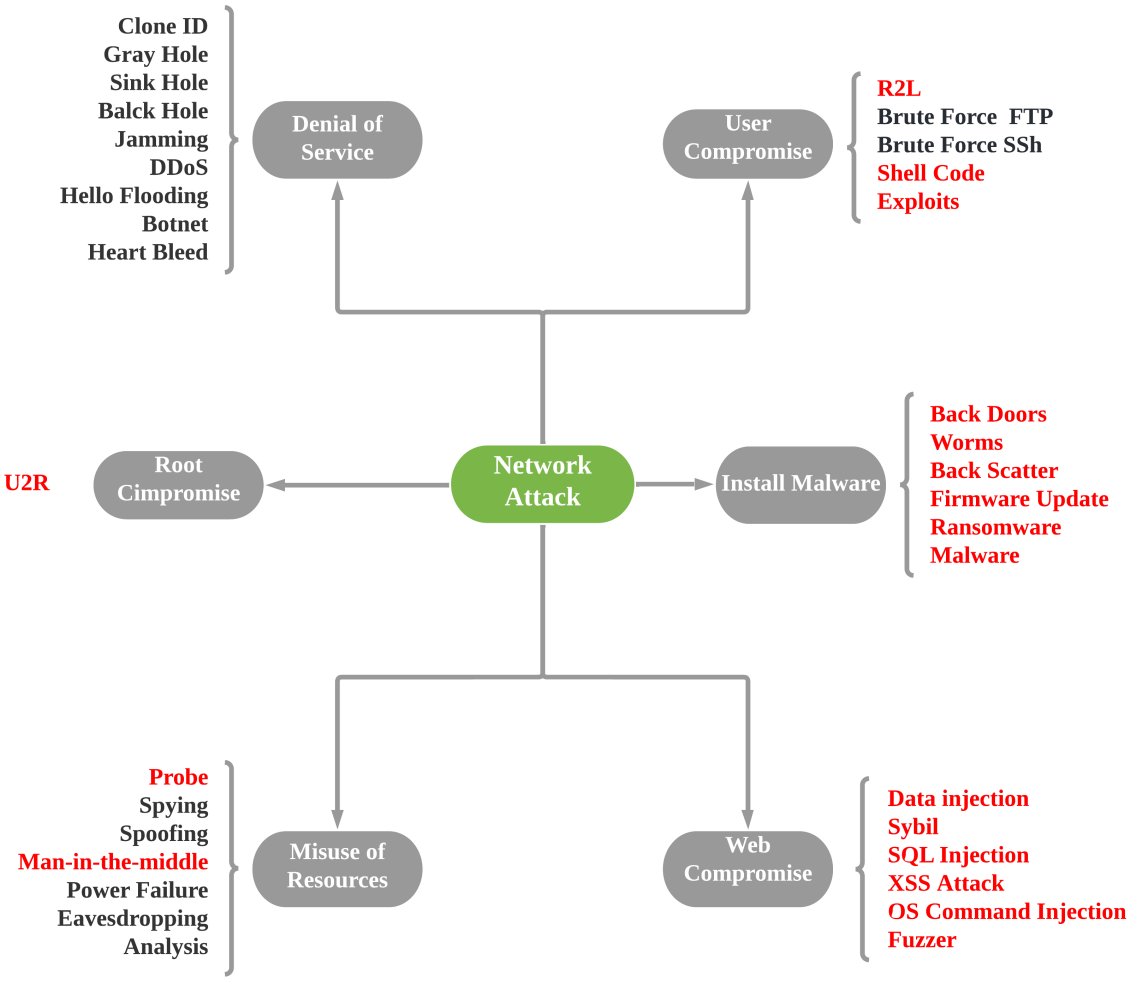


Figure 2.1: Attack Taxonomy

the root, which is usually hacked in order to illegally obtain the root's privileges when legally accessing a local machine. A Remote to Local (R2L) attack is documented in open datasets such as NSL-KDD and KDD99.

- **Misuse of Resources** – refers to any network or system resources used in a wrong way or purpose, such as IP sweeping, port scanning, and spoofing [68]. Since this kind of threat can be the initial step of an attack, it exists in almost all types of network intrusion benchmarks, such as NSL-KDD, UNSW-NB15, and CICIDS2017.

- **User Compromise** – perpetrators apply tools or inject a piece of code to gain regular user privileges and further run malicious code or steal user data from victimized devices [98]. This kind of operation is always followed by local to root attacks to escalate privileges, discover more valuable data, or more fully control victimized devices. The techniques categorized as 'user compromise' also can be used for the LM stage in APT attacks. However, various datasets have different names for user compromise, such as Remote to User, as part of an Exploit attack.

- **Install Malware** – malware is software that allows malicious users the ability to compromise a system, resulting in potential exposure of sensitive information or obtaining remote control of the host [43]. Experience shows that cybercriminals install spyware [131], Trojans [133], worms [76] and execute arbitrary code [47] in users' computers and devices. There are a number of attack strategies, including back doors, back scatter, and MAWILAB, among others.

- **Web Compromise** – occurs when attackers gain access through a poorly structured website or a Web application. Specifically, attacks that can occur through web compromises are usually through cross-site scripting or SQL injection [106]. In Fig. 2.1, SQL Injection, Sybil, Data Injection, and other web attacks belong to a class of web compromise attacks.

2.2 Intrusion Detection Schemes

This section provides a review of IDS for networks, emphasizing machine learning approaches. Intrusion detection systems can be categorized into three types based on the detection mechanism, namely anomaly-based, signature-based, and hybrid [157]. Table 2.1 summaries the different approaches to implement network intrusion systems [113], (e.g., statistical, data mining, and machine learning), among which machine learning based methods [134], are used in hybrid detection systems [54].

Table 2.1: IDS Techniques

IDS Technique		REF.	Benefits	Limitations
Signature-based Techniques		[151] [157] [50] [119] [95] [64] [88]	Easy to implement; Low false alarm rate;High true positive rates for known attacks	Hard to detect zero-day attacks; Hard to detect the variation of known attacks; Frequently updates for new attacks
Anomaly-based Techniques	Statistical	[100]	High detection accuracy	Vulnerable to injecting malicious traces
	ML-based	[137] [45][150]	Easy to detect explicit and implicit pattern	Expensive computation resources
	Control flow graph	[147]	Comprehensively analyze malicious behavior	Time-consuming; IDS itself can be attacked
	Finite State Automate	[24]	Able to detect a series actions	False Alarms
Hybrid Techniques	Signature +Anomaly	[28]	Low false positive and false negative	Relatively slow speed
	ML Hybrid	[128]	Decrease false alarm	Large model and expensive computation

2.2.1 Signature-based Methods

A signature detection system differentiates malicious from standard traffic patterns or application data [151]. If the network behavior matches a stored pattern or signature of nefarious code, it is marked as abnormal. Thus, signature detection can identify attacks but only if a signature (e.g., label) for the attack is available. Hence, the method performs poorly in cases representing a new kind of attack [157]. The study in [50] presents a signature-based network IDS that uses sensors to match network packets against a pre-configured set of intrusion signatures. Moreover, the authors give several design schemes for a multi-threaded network intrusion detection system.

Lu et al. [80] presented an Automatic Temporal Correlation Traffic Detection System (ATCTDS) after examining the characteristics of malware supplied by APT attacks. Flow filtering in the first phase and fuzzy recognition in the second phase can detect and segregate benign flows to the fullest degree possible. ATCTDS's layered construction considerably minimises detection time. When an APT attacker starts transmitting payloads, the transmission interval of malicious packets differs dramatically from that of normal packets, according to a comparison. They defined 11 types of temporal correlation features linked to the APT malware transmission process, as well as two types of combination features including temporal correlation features and network features. The above features are supplied to a gradient enhanced decision trees anomaly detection system, which performed attack identification.

Liu et al. [78] set out to investigate the performance of a network attack detection approach based on a machine learning algorithm in Contiki-NG, an IoT environment. They employed a range of supervised and unsupervised algorithms to classify network traffic after referring to the attack categories in the NSL-KDD data set. The results show that supervised performance is superior to unsupervised performance, with the False Positive Rate of XGBoost reaching up to 97.0%.

To identify network attacks, Do et al. [38] proposed a composite deep learning model. After feature extraction, network data is transmitted to Bidirectional Long-Short Term Memory (BiLSTM) for processing. The output of BiLSTM is then processed using Graph Convolutional Network (GCN) which provides the final classification result. The model exhibits the greatest performance among all assessment criteria when compared to Multi-layer Perceptron (MLP) and GCN alone. However, the data processing time has increased as a result of the presence of BiLSTM.

Bai et al. [19] proposed leveraging Windows event logs to identify RDP LM attacks. The cornerstone of their LM attack detection is that event logs are produced on the Windows system once hackers join the intranet and use RDP to compromise other machines.

The combined dataset is utilised to detect harmful LM events using seven supervised machine learning methods. Finally, Logit Boost provides the best detection result for the LM attack based on RDP, with 99.99% accuracy and 99.7% F1-Score, respectively.

2.2.2 Anomaly-based Methods

An anomaly detection system compares a captured pattern in a data stream to pre-stored baseline patterns of known threats. If the patterns match, action is taken accordingly the operational needs of the time. The system can eliminate unknown attacks, so-called "zero-day" attacks, by simulating the network's normal operations to detect deviations [100]. However, the downside of anomaly detection is the potential for inevitable false alarms. Based on applied techniques, anomaly-based intrusion detection methods can be categorized according to their use of learning, statistics, and miscellaneous techniques [137]. In a statistics-based approach, network traffic activities capture and create a profile representing the random behavior of network traffic. This profile takes various essential features into account (e.g., traffic rate, number of packets, connections, multiple IP addresses). The anomaly detection procedure for network traffic utilizes two datasets; One contains the current observed profile, and the other is a previously trained statistical profile. As network events occur, the existing profile is determined, and a quantitative value is calculated by comparing the two actions. Thus, intrusion detection systems can figure out anomalies based on the calculated scores. On the one hand, the statistics-based methods lead to increased detection accuracy. However, the downside of statistical methods is that malicious users could hack into the system by training the system and its parameters by injecting malicious code.

The category of miscellaneous detection techniques consists of all the approaches not included in the statistical and machine learning groups and includes using such approaches as control-flow graphs [147], finite-automata [24], and description languages [150]. For instance, the study in [58] provides a framework to determine intrusion based on the general-purpose Unified Modeling Language (UML). The authors design a UML profile called UMLintr that allows software developers to specify intrusions using UML notations extended to suit the context of various intrusion scenarios. The framework uses the expressiveness of UML and eradicates the requirement of using attack languages. It is unnecessary to study the UML language for attack descriptions, thus reducing developers' efforts in specifying intrusion scenarios. The authors in [150] introduce a description language (e.g. n-grams) for intrusion detection. They show both anomaly detection and classification based on n-grams. Besides, they develop criteria for utilizing data using different schemes and applying the defined criteria to web intrusion detection. Although

miscellaneous detection techniques are flexible and scalable, they are not effective for high quality and high volumes of data due to time-consuming data management caused by a low-power and limited memory environment [45].

In recent years machine learning algorithms have emerged as a tool to assist in anomaly-based intrusion detection systems. Machine learning techniques are used to build explicit or implicit models that utilize pattern analysis to classify data. A machine learning-based network intrusion system can change its execution strategy as it obtains new information [45]. There are a number of machine learning algorithms used in anomaly-based detection systems including Support Vector Machine (SVM), Bayesian, neural network, clustering and outlier detection [137] [45]. The main disadvantage is the need for relatively expensive computation resources to perform the necessary data processing.

2.2.3 Hybrid Methods

As for a hybrid detection system, it combines the signature-based and anomaly-based approaches to overcome their drawbacks and benefit from their advantages [128]. Hence, hybrid detection systems can synthesize the benefits of two methods such as the detection of "zero-day" attacks and the detection of signatures of known attacks. The study in [128] introduces hybrid techniques namely using Gaussian Mixture clustering with Random Forest (RF) Classifiers as well as using K-Means clustering with RF Classifiers in order to detect intrusions. The authors reported that this proposed hybrid approach decreased the false alarm rate to 0.04%. The authors in [28] demonstrate a hybrid intrusion detection system, called INTI, which was designed to detect sinkhole attacks in the routing services within an internal network. INTI links anomaly-based concepts and signature-based methods to monitor and audit packets between nodes and extracts reputation and trust features for node evaluation.

2.3 Machine Learning in Attack Detection

SVM is well-known for its ability to effectively handle high-dimensional datasets, even if the number of dimensions is higher than the number of samples, while still maintaining its effectiveness. Since only a subset of samples, named as support vectors, is used by SVM in its decision function, SVM is memory efficient [55] [162] [11] [20] [152] [9] [10] [23] [93].

Decision Tree (DT) algorithms depend on some pre-conditions and select a uni-variate split for the root of the tree and recursively repeats the process [114]. Pruning reduces

the size of trees and is performed after a complete tree has been built. The study in [114] shows that its DT achieved 99.99% accuracy to identify Bot-IoT attacks. Furthermore, the study in [62] demonstrates 95.64% overall accuracy using NSL-KDD dataset via DT.

RF is an ensemble machine learning algorithm that consists of a number of decision trees [13] [81]. Each decision tree gives a prediction result and an overall result via voting on by all decision trees. For instance, it is assumed that three decision trees are applied in RF for binary classification, class A and class B. Two of the trees give prediction results at a point belonging to class A, while one of three trees estimates the sample belongs in class B. Then RF considers the data as part of class A since the majority of members voted for class A. Multiple class classification follows in the same manner as binary classification. The RF algorithm is deployed to detect network attacks in Ethernet IoT and self-designed networks. Meanwhile, we review several popular open-source network attack datasets (e.g., NSL-KDD, AWID, CICIDS2017, UNSW-NB15) [55].

A K-Nearest Neighbor (KNN) algorithm is a data classification method that estimates the likelihood that a data point will become a member of one group or another, depending on which group the nearest data point belongs. It means a KNN classifier is a distance function, in effect an application of the Pythagorean theorem, which measures the difference or similarity between two instances [152]. The KNN machine learning approach is practical in several network categories (e.g., IoT, industry WLAN IoT, and self-designed networks) where each network could be affected by different attacks. From the literature, KNN has been deployed to detect several network attacks such as APT, remote tripping command injection, relay setting change, data injection, ransomware, and DoS. However, the KNN-based classification approach almost can not detect R2L and U2R attacks since the number of these attack traffic are extremely low. Therefore, considering the promising performance of KNN for use in IoT networks, various papers have proposed approaches based on this algorithm with the intent to mitigate network attacks [152].

Naive Bayes (NB) algorithm utilizes the Bayes' Theorem, where classification assumes that the presence of a particular feature in a class is not related to the presence of any other feature. It is deployed in [107] [20]. The studies in [163] describe classification methods based on the use of for network intrusion detection under various network types. However, detection performance is not satisfied enough. For instance, in [20], the overall accuracy is 74.19% for UNSW-NB15 dataset.

XGBoost is an ensemble machine learning algorithm based on decision trees that uses a gradient enhancement framework [52]. The studies [144] report on an XGBoost-based classification method to mitigate network attacks with a detection performance of 97%.

Another Ensemble learning method is strategically generating multiple models and

combining them to solve specific computational intelligence problems. Ensemble learning algorithms utilize a number of different learning algorithms to obtain better performance than simply using a single learning algorithm. For example, the ensemble approach can integrate various algorithms such as Adaboost, decision trees, the Naive Bayes classifier, and Artificial Neural Network (ANN) learning methods to identify network attacks, including in IoT networks [116]. The authors in [49] have proposed a new model that takes SVM as the primary learners and trains multiple SVM models in parallel. This approach achieved more than 99% accuracy in both KDD99 and NSL-KDD datasets. The work in [69] demonstrates a comprehensive review of IDS integrating ensemble models.

The Linear Regression (LR) algorithm attempts to model the relationship between two variables by fitting a linear equation to observed data. Thus, One variable is considered an explanatory variable whereas the other is a dependent variable [86]. LR demonstrates 100% accuracy to identify network attacks, which is applicable to various types of attacks such as DoS, probing, and malicious control [107] [55].

A MLP is a type of feed forward artificial neural network consisting of at least three layers of nodes in terms of an input layer, a hidden layer, and an output layer. Nonlinear activation functions in the hidden layer(s) and the output layer for each node or neuron are used as a supervised learning technique. Moreover, MLP can classify non-linearly separable data. To this end, several publications present MLP-based methods to identify network attacks with a promising overall accuracy [146] [65] [105] [144].

A Fully Connected Network (FCN) contains fully connected layers where all inputs from one layer are connected to each activation unit in the previous layer. Fully connected neural network algorithms have proven effective at network intrusion detection with promising performance under different network types. A Deep Neural Network (DNN) model with one input layer, two hidden layers, and one output layer. The three layers are fully connected, which means neurons in each layer have connections to all active neurons in the previous layer. Assuming that there are m neurons in layer $l-1$.

A Deep Autoencoder (DAE) consists of two symmetric multi-layer feed forward networks (e.g., encoder and decoder). DAE is used to reproduce the input at the output layer via the encoder and decoder. The number of neurons in the output layer is the same as the number of neurons in the input layer [159] [158]. Researchers apply DAE in various fields such as dimension reduction, anomaly detection, and noise filtering [66] [120].

Variational Autoenco (VAE) is a type of neural network that consists of 1) an automatic encoder with regularized coding distribution in a lower-dimensional space and 2) a decoder for reconstructing the compressed data [82]. The encoder maps the original input x into z under a given activation function. Meanwhile, the decoder maps z back to $x^{'}$. The

objective is to minimize the constructed data loss between original data x and the new data $x^{'}$.

VAE can be widely used to detect various network attacks. 97.5% detection efficiency can be achieved in the UNSW-NB15 data set [82]. The authors in [82] show VAE is utilized and implemented to detect network attacks. Variational Autoencoder-Fully Connected Network (VAE-FCN) connect a VAE to an FCN, while VAE-Pure is a model based on the inclusion of labels, which are viewed as an independent feature. VAE-FCN does not use labeling information and operates in an unsupervised manner [82], and can detect attacks in many different network categories. However, the performance is not particularly good; for instance, in [82], overall recall is 75.3% under the Kyoto-Honeypot dataset.

A Recurrent Neural Network (RNN) is a type of neural network that takes sequence data as input and performs recursion in the direction of sequence evolution [146]. It is usually used to process sequence information such as strings, lists, tuples, byte arrays, and range objects. Like other neural networks, it has an input layer, an output layer, and a hidden layer. However, the value of the RNN's hidden layer depends not only on the current input; but also on the hidden layer result of the last input [156]. The authors in [155] [146] introduced RNN integrated framework to detect attacks.

Intending to avoid long-term dependency issues, Long Short Term Memory (LSTM) is a special type of RNN [57]. It is used in classifying, processing, and making predictions based on time series data, noting that there can be lags of unknown duration between important events in a time series. LSTM remembers information for long periods with their typical behavior. The study in [40] [40]demonstrates a LSTM-based approach for mitigating network intrusions.

A Sequence to Sequence (Seq2Seq) algorithm consists of two RNNs corresponding to the encoder and decoder. Both encoder and decoder can have one or more units, either an LSTM or Gate Recurrent Unit decoder [82]. The authors in [82] demonstrate that the Seq2Seq algorithm has outstanding performance in terms of 100% accuracy for a number of network attack types. Typically, Seq2Seq is a generative model, whereas, in [82], it is used as a supervised method using labels to identify network intrusions.

A Convolutional Neural Network-Long Short Term Memory (CNN-LSTM) combines Convolutional Neural Network (CNN) layers and LSTM to classify or predict data. CNN layers consist of convolution layers and max-pooling layers used for extracting features of input data. The multi-layers of the LSTM are stacked behind the CNN layers and used to predict a data sequence. With CNN's auxiliary, this model can not only determine the relationship among long-term sequences; but also minimize the total error on a set of training sequences [148]. CNN-LSTMis widely adopted to solve activity recognition as well

Table 2.2: The limitations of previous work

Ref.	Network	Host	Limitations
[34]	✓	×	Does not include the host-based features in the detection models. Only network flows features are considered.
[135]	✓	×	
[46]	×	✓	Does not include the network-based features in the detection models. The host events are processed to obtain the final detection results.
[96]	×	✓	

as image and video descriptions [91] [145].

The K-Means clustering algorithm is an unsupervised ML Methods, which originates from a vector quantization method in signal processing [11]. The purpose of the K-Means clustering is to divide n points into k clusters so that every point belongs to the cluster corresponding to the nearest mean. K-Means are usually applied to the fields of vector quantization, cluster analysis, and feature learning. K-Means integrated methods were presented in [33] [11] for attack detection.

Density-based Spatial Clustering of Applications with Noise (DBSCAN) is an unsupervised learning method that is well-known for clustering samples based on their density and identifying outliers located in low-density areas [112]. The studies in [33] [29] illustrated DBSCAN algorithm to detect network intrusion.

SE

2.4 Gap Analysis

In this section, the gap between previous related researches and this thesis are analyzed. Table 2.2 illustrates the limitations of previous work.

Due to the fact that APT targets enterprises' internal networks that contain the significant economic or scientific value, and these attack flows are restricted to internal use, the majority of researchers are unable to access APT's first-hand material. Additionally, if researchers desire to do an in-depth study on APT, they must invest significant time in

developing and maintaining local area networks that closely resemble contemporary network topologies. These limitations have stymied APT research and exacerbated the lack of publicly accessible APT datasets. After extensively reviewing all steps of the APT, we build the testbed and collect all APT attack network traffic packets, which are aggregated to generate the SCVIC-APT-2021 dataset.

The majority of mainstream IDS regard network flows and host events as distinct components for attack detection; for example, [34] [135] focus on network flows while [46] [96] concentrate on host events. When a network attack happens, the victim receives the attacker's network packets. Victim is hacked by a succession of damaging actions, such as information leakage and data loss, that are carried out in accordance with the attacker's instructions and payloads. When the attacker takes actions, such as installing payloads, logging in and out, it may leave traces on the victim device. If network flows and host events are analyzed independently, this linkage information between network flows and host events is lost which leads to reducing the attack detection efficiency. As a result, the approach of integrating network flows and host events is presented to address this problem. Simultaneously, the victim's network flows and host events are provided to the machine learning detection technique to increase the detection efficiency.

Chapter 3

Network-based Advanced Persistent Threat Detection Scheme

APT attack detection schemes have been widely researched and implemented in IDS. If IDS detects the APT attack, the administrator will be notified and instructed to take action as quickly as possible to prevent the attack from further spreading [129]. Non-ML detection methods and ML-based detection methods are the two primary research paths. Traditional algorithms and modelling tools like Markov chains [25], finite state machines [161], and graph analysis [85] are utilized in non-ML detection approaches to detect the attack chain of APT attacks and anticipate all attack pathways. Although these approaches can identify and block APT attacks to some extent, the attacker can simply evade detection if the APT attack does not follow the predicted route [140]. To compensate for the drawbacks of non-ML detection approaches, dynamic ML-based detection methods are presented. To categorise data points, ML-based IDS leverages network or host features. The ML-based techniques are more versatile since they can assess a single network flow or host event [132]. At the same time, ML's scalability allows it to identify new sorts of network threats that non-ML approaches miss.

Network traffic is simpler to obtain than host events, and network-based features are more abundant than host-based features. Therefore, this chapter relies on the network traffic to identify APT attacks. The majority of popular datasets in the network security fields comprise network traffic; some are APT attacks, while the majority are focused on traditional network attack types [117]. The SCVIC-APT-2021 Dataset based on six distinct stages of APT attacks is generated to help identify APT attacks more reliably. In addition, the proposed PKI model is applied to the benchmark datasets and the SCVIC-APT-2021 Dataset in order to identify and prevent APT attacks efficiently and accurately.

3.1 Dataset

Although the features of each network intrusion dataset varies, they nonetheless share certain common characteristics. PCAP format, which includes original network packets, and CSV/ARFF format, which contains features derived from network flows, are available in almost all public network incursion datasets [101].

In this study, three datasets are utilized to detect the APT attack. NSL-KDD and UNSWNB-15 are two public datasets that are extensively used as benchmarks to assess models' performance. SCVIC-APT-2021 is a dataset created by a self-built LAN according to the real-world APT attack. Five typical APT attack stages are included in this dataset.

3.1.1 SCVIC-APT-2021

APT Attack Main Stages

It is vital to understand the major stages and attack methods of the APT before developing the APT dataset. This is the foundation for restoring the complete APT attack chain, as well as the key to the dataset that represents the APT attack network traffic.

The six primary stages of an APT attack are shown in Fig. 3.1. The yellow color blocks (e.g., Reconnaissance, Initial Compromise, Post Stage) indicate the initial and final stages of the APT attack, which mainly include information collection and trace spiritual work. The red color blocks (e.g., LM, Pivoting, DE) represent the attack stages of APT, in which hackers gradually conquer the network domain and leak data. Hackers may change the sequence of the three attack stages according to the network environment.

Reconnaissance: In the Reconnaissance stage, the adversaries aim to collect enough information to prepare for the APT attack and develop appropriate attack strategies for the target network. The information collected social engineering information and target network's infrastructure information. Adversaries collect social information, hobbies, and social networking of users of the target network [161]. This information is used to assist in completing the *Initial Compromise* phase. Target network's infrastructure information includes the models of network switches, routers, and the communication protocols used in the network [99]. Attackers use covert attack methods based on this information to ensure their security and construct a reasonable attack path to bypass the network administrators and IDS.

Initial Compromise: In this stage, the attacker uses the information collected in the *Reconnaissance* stage to attack the network user or the target network's vulnerabilities

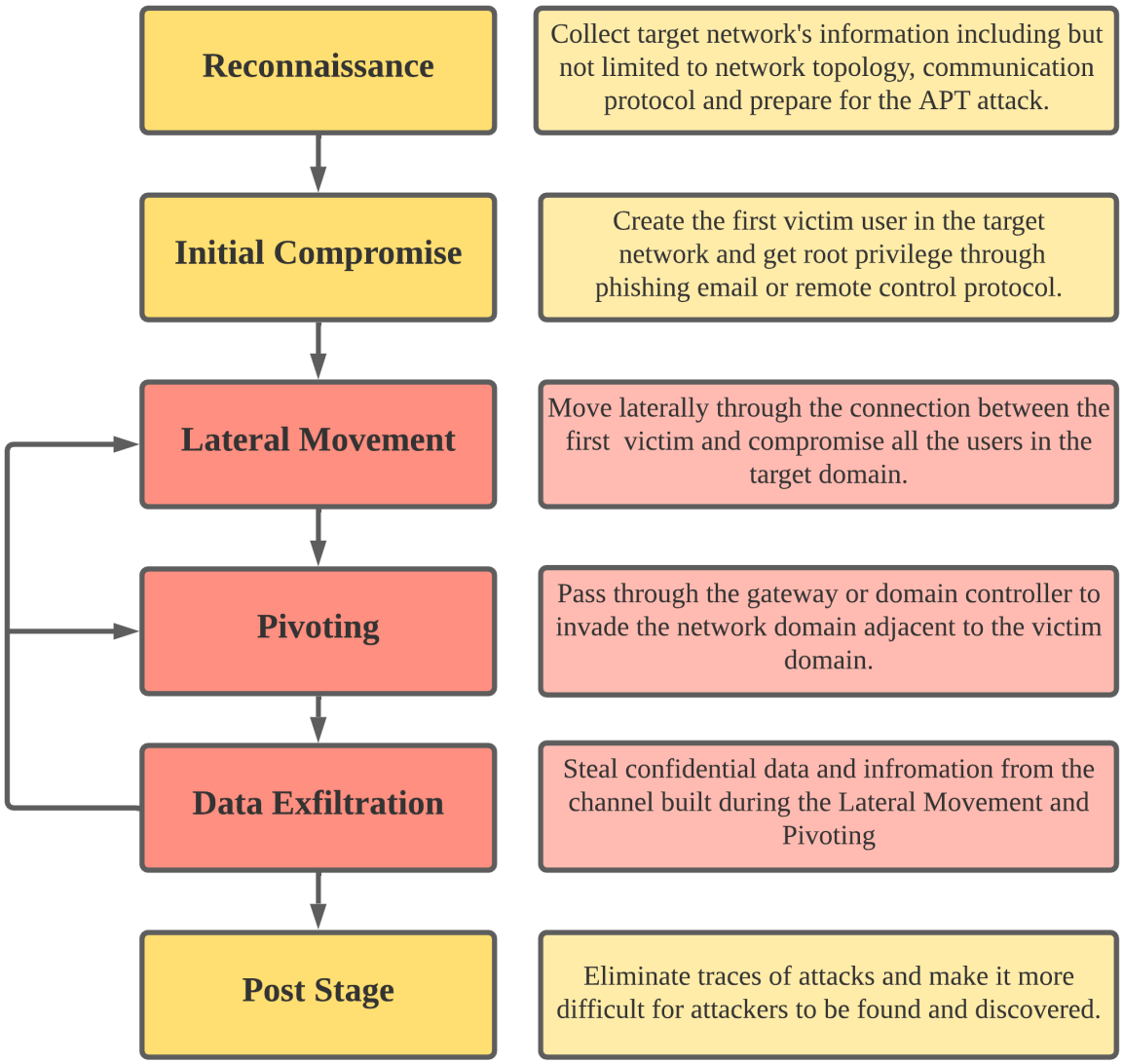


Figure 3.1: APT Attack's 6 stages

to obtain the administrator privileges of the compromised computer. Then, they use this node to carry out large-scale attacks on the target network. *Initial Compromise* mainly categorizes two forms. The first form is mainly for computer users [30]. After collecting enough social engineering information, adversaries attack network users through

different techniques such as phishing emails or links to trick users into clicking in order for the attackers to gain administrator privileges. The second type aims at computer vulnerabilities. By understanding the computer's operating system, patch version, and installed software, attackers use known vulnerabilities, and advanced hackers even use zero-day vulnerabilities to obtain root privileges [94].

Lateral Movement: In this stage, the APT attackers use the compromised node to obtain other computers' credentials in the same network domain. Attackers use various methods to move laterally according to the actual situation in the network domain [139]. Commonly used techniques include pass-the-hash, pass-the-ticket, and RDP [12]. It is worth mentioning that LM and reconnaissance within the network domain can occur simultaneously. Unlike the original reconnaissance, *Internal Reconnaissance* mainly refers to the attacker's activities inside the target network. After obtaining the administrator credentials, the adversaries obtain essential information such as the computer's running process, IP, port numbers. With the deepening of reconnaissance, attackers can also modify the attack strategy at any time, making APT attacks more difficult to detect.

Pivoting: For companies and organizations, a multi-location office has become a typical application scenario. Therefore, the internal network is often composed of two or more remote subnets responsible for different tasks. Each subnet has a domain controller and several user computers. The domain controllers are connected by a communication channel such as a Virtual Private Network (VPN). When an APT attacker encounters the need to move laterally across different subnets, pivoting attacks become the most powerful tool [14].

Date Exfiltration: This stage mainly uses the session established by the previous attack for information and file return. To avoid being noticed by the IDS system and anti-virus software, attackers usually use the tools that have been written to cut the files and then transfer them. Advanced hackers will take the different cut parts to different DNS servers and then collect the sliced file fragments to reduce the possibility of discovery [90].

Post Stage: When the hackers obtain all the required information, the APT attack phase is over. The primary purpose of the post stage is to clean up the traces of APT attacks and prevent follow-up investigations [103]. During an APT attack, the attacker installs various malware on the compromised computer. Multiple logins and reconnaissance generate log files and browsing records. Therefore, cleaning up these traces can well hide the APT attack and avoid tracking investigation.

Table 3.1: All the attack techniques used for SCVIC-APT-2021 generation

Attack Stage	Lateral Movement	Pivoting	Data Exfiltration
Attack Techniques	Pass the Hash/Ticket	AutoRoute	DNS Tunnelling
	Remote Desktop Protocol	Socks4a	C2 Tunnelling
	WMI	Proxy Chain	Encode and Encrypt

APT Attack Techniques

LM, Pivoting, and DE, the red blocks in Fig. 3.1, are the three major attack stages of APT. These three stages have no set order and can be carried out depending on the attack's requirements [154]. Furthermore, each attack stage can be completed using a number of attack techniques. As a result, detecting a single attack technique has a limited promotion on detecting the whole attack chain. Each attack stage is executed by three to four attack techniques during the generation the SCVIC-APT-2021, in order to better suit the APT attack in the actual network environment. These attack techniques, as demonstrated in Table 3.1, will be demonstrated one by one in this part.

Metasploit Framework: The Metasploit Framework is the world's most advanced penetration testing tool. It comes with a number of tools that can be used for security testing, making the process of penetration testing much easier [26]. All contemporary operating systems are compatible with Metasploit Framework, although Kali Linux is popular since it incorporates Metasploit Framework natively and does not require user configuration files. As a result, Kali Linux is employed as the attacker's operating system and several vulnerabilities and payloads included therein are applied to exploit the two existing network domains while creating the SCVIC-APT-2021 dataset.

Initial Compromise: Before launching the three major stages of the APT attack, the attacker must first accomplish the Initial Compromise and utilise the compromised machine as a springboard for later attacks. In general, social engineering, phishing emails, and other technologies are frequently employed at this time, making it difficult to protect network users. A vast number of internal networks employ File Transfer Protocol (FTP) as an application layer protocol between the client and the server [118]. The Very Secure FTP Daemon (VSFTPD) is an FTP server for Unix computers. It is also the default FTP server for well-known Linux operating systems, such as Ubuntu and CentOS. One of the most important characteristics of vsftpd is security. The "smileyface" backdoor in vsftpd-2.3.4, on the other hand, allows users to login with ":)" and receive a command shell on port 6200. To finish the ensuing APT attack, it is assumed that the Ubuntu system runs

vsftpd-2.3.4, and then the attacker utilizes the Metasploit Framework's tools to complete the Initial Compromise.

Lateral Movement: When an attacker takes control of a computer on the intranet in a penetration test, the compromised host is used as a springboard to acquire access to additional machines in the domain by different ways such as collecting credentials to further broaden the reach of the victims. The attacker may eventually obtain access to the Domain Controller (DC) and perhaps control all machines in the intranet environment using such methods. During the generation of the SCVIC-APT-2021, the LM attack is implemented using the four techniques listed below.

- **Pass the Hash**: This approach attacks the users by locating the password hash value (typically NTLM Hash) [60]. The password hash value is used to complete user authentication and is equivalent to the plaintext password. When logging in to computers, most users utilize domain accounts, while a large number of machines share the same local administrator account and password while installation. As a result, if the computer's local administrator account and password are leaked, the attacker may log in to additional computers on the intranet host using the hash transfer mechanism. The attacker does not have to spend time breaking the password hash value using the hash transfer attack (thereby obtaining the password plaintext). This strategy allows the attacker to finish the data theft and move laterally within the domain.

- **Pass the Ticket**: The Kerberos ticket is extracted through operating system credential dumping when the attacker performs a Pass the Ticket attack [130]. Kerberos tickets are often associated with users and are used to get access to some specific resources that the user has access to. This implies that if an attacker acquires the domain administrator's Kerberos ticket on a certain device, it will have access to all operating privileges and private information on any machine in that cracked domain.

- **Remote Desktop Protocol**: RDP is a multi-channel remote desktop protocol that allows a local computer to connect to and control a remote computer over the internet. Clients are available for most versions of Windows, Linux, and Mac OS. The RDP client listens for data on the TCP 3389 port and performs operations as directed by the remote controller. After obtaining a valid user credential, the attacker can log in as a legitimate user and take control of the remote machine [18]. Fig. 3.2 demonstrates the RDP-based LM Attack.

- **Windows Management Instrumentation**: The core foundation for controlling data and running objects in Windows Operating System (OS) is Windows Man-

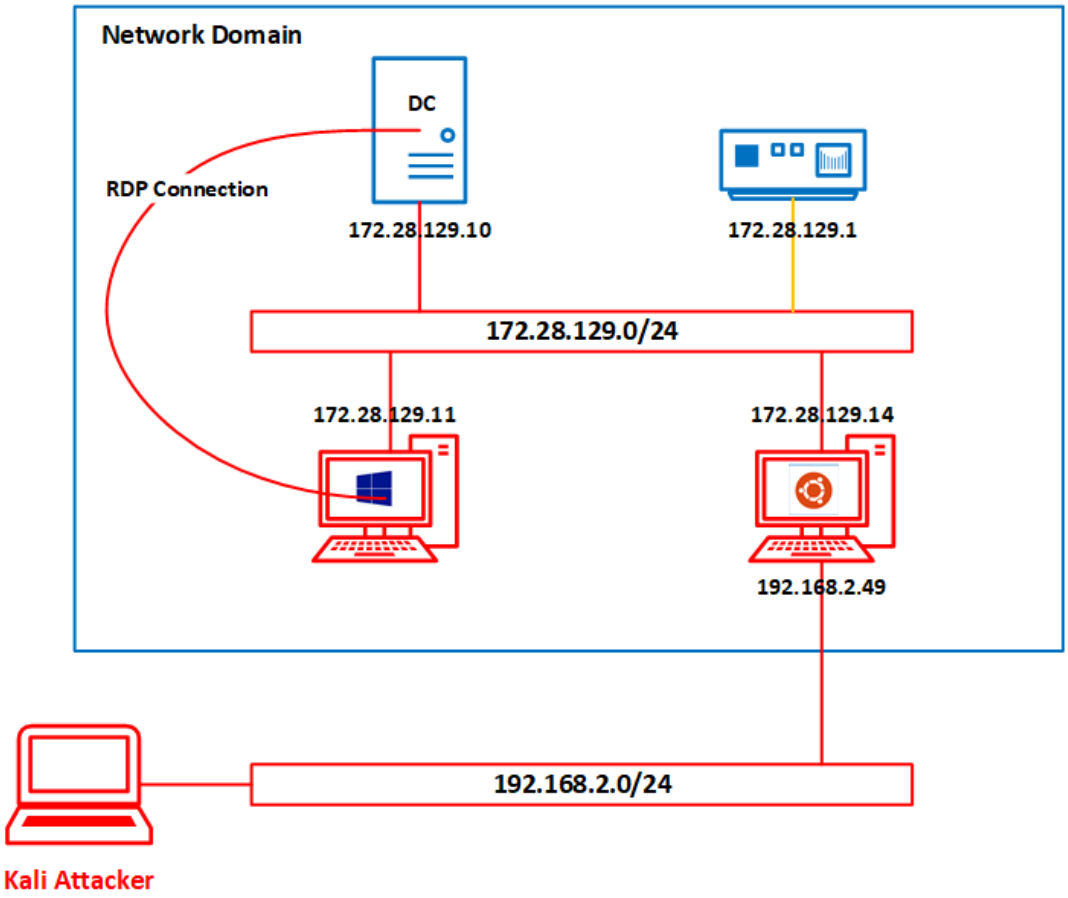


Figure 3.2: RDP-based Lateral Movement Attack

agement Instrumentation (WMI). WMI is supported by all Windows OS starting with Windows 98. Based on Windows Remote Management (WinRM), the WMI service allows for localised customization or remote service control. WMI scripts or programmes can be written by administrators and developers to remotely control machines and perform associated activities. At the same time, attackers employ WMI to interface with remote devices, collect data, and execute LM payloads [142]. WMI-based attacks, on the other hand, can only be deployed on PCs running the Windows OS.

Pivoting: Pivoting is a method of routing traffic from a attacker's computer to networks that the attacker can not directly access [15]. The internal network is frequently

separated into numerous network domains according to different roles and permissions when a business creates the network architecture. Some are in charge of management, while others are in charge of data storage. A firewall or a DC connects each network domain. At first, the attacker can only access the network domain through an external network interface, and it cannot access the core business or secret information directly. As a result, in order to uncover crucial files, hackers must employ Pivoting technology to break down the barriers across network domains. An example of Pivoting is depicted in Fig. 3.3

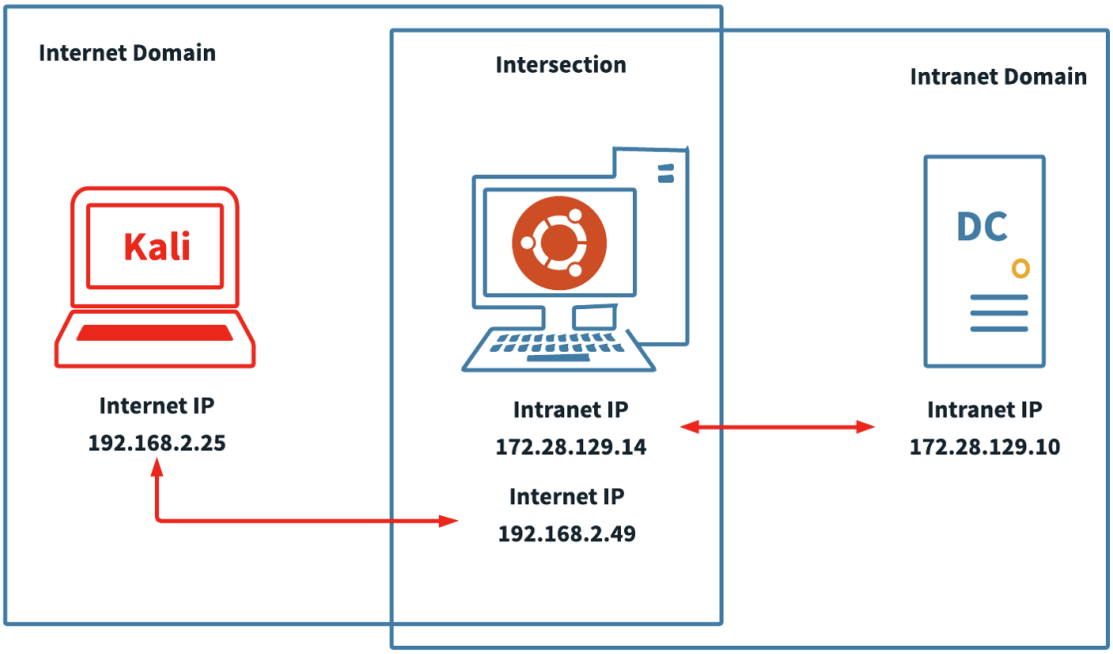


Figure 3.3: An example for Pivoting attack

Suppose there are two network domains:

- Internet Domain (External Network): a network that hackers can directly access: 192.168.2.0/24
- Intranet Domain (Internal Network): a network that hackers cannot directly access: 172.28.129.0/24

Pivoting is the method by which hackers route traffic from a cracked network domain to a target network domain via specified machines or transit points. In the Fig. 3.3, Hackers try to get into the intersection of internal and external networks, which is a Linux PC. The computer is connected to two networks and can access both at the same time. Attackers use this machine to break into internal networks and take control of other machine, the internal DC.

The attack techniques used for generating SCVIC-APT-2021 are illustrated below.

- **Autoroute**: Following cracking a specific computer, the exploited device is used as a springboard for future attacks [61]. Autoroute is a Metasploit Framework-based automated routing script. It establishes a reverse shell between the attacker and the victim system, allowing the attacker to invade the next network domain via the victim computer.

- **SOCKS4A**: SOCKS is a network transmission protocol that is primarily used as an intermediate between internal network users and external network servers for communication [97]. Users in the firewall connect to the SCOCKS proxy server in order to access external network resources. The SOCKS proxy server transmits the request to the destination host address if the user is permitted to contact the external resources; else, the user's request will be intercepted. SOCKS4A is a modified version of the SOCKS4 protocol that allows users to connect to hosts that are unable to resolve domain names. The Metasploit Framework's socks4a module lets attackers to set up a listening tunnel that other programmes can use to attack victims [61]. As a result, this approach serves as a prerequisite for future proxychain attacks.

- **Proxychain**: If an attacker wants to pivot to target a network or computer that cannot be reached directly, it can use a SOCKS proxy server on a victim machine with access to the resource [48]. As a result, the attacker achieves its aim of progressively penetrating the target network domain by working its way down the proxy chain. More than three layers of poxychain are sometimes required in a vast and complicated internal network environment, and the premise is identical to that of two levels. However, as the poxychain grows, the attack may fail due to increased bandwidth loss and decreased stability [48]. As a result, the attackers use flexible attack approaches to breach the internal network and extract critical information during the APT attack.

Data Exfiltration: The technique by which an attacker steals and outputs internal data is known as DE [141]. Through LM and Pivoting, hackers gained root privileges and

valuable data on most machines in the target network prior to data transmission. These data are packaged and sent out of the intranet via channels or cloud services. A high volume of network traffic or DNS requests are created throughout the DE stage. Encode and Encrypt, Command and Control (C2) Tunnelling, and DNS Tunnelling are used for Date Exfiltration stage while generating SCVIC-APT-2021.

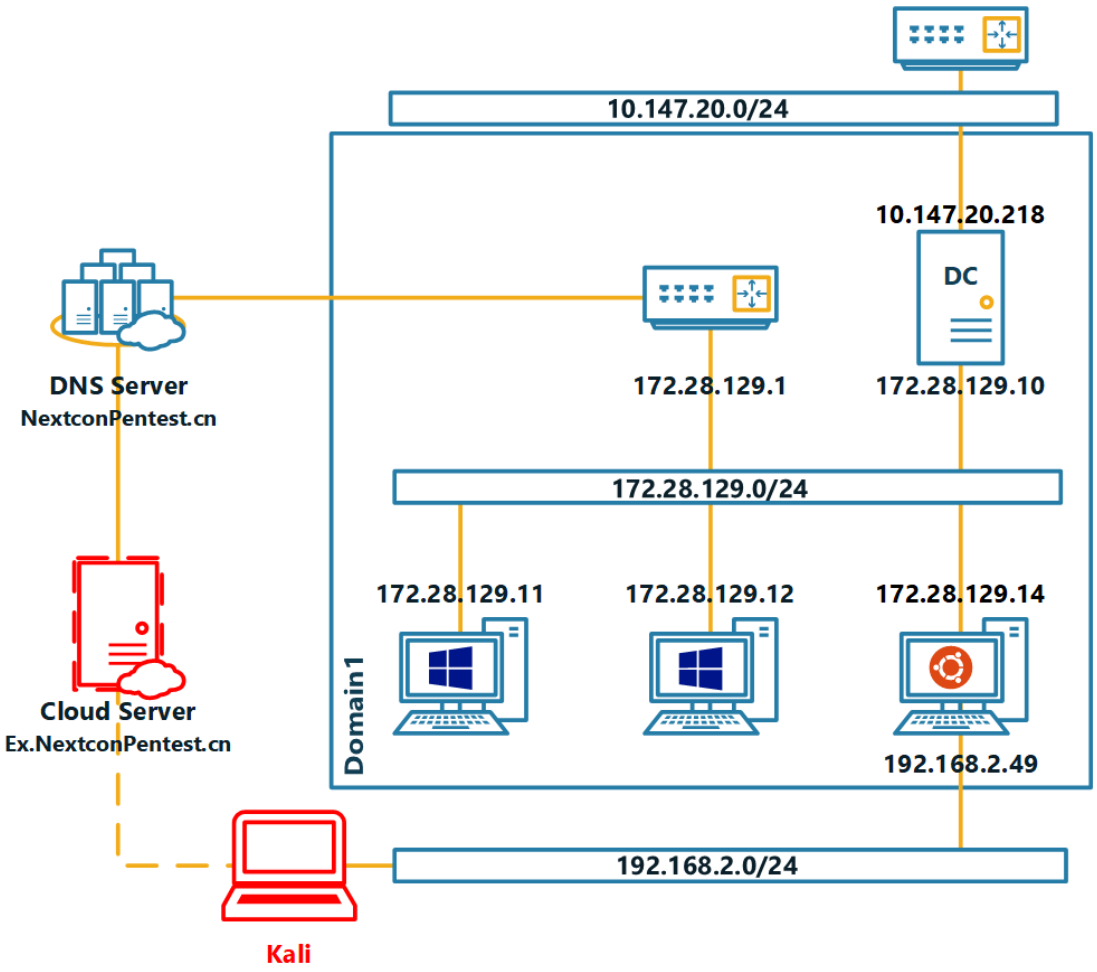


Figure 3.4: DNS Tunnelling attack in Data Exfiltration Stage

- **Encode and Encrypt**: The attacker must compress and package the target data using an encryption method before delivering it as regular traffic. This lowers the risk of data breaches detection while also speeding up data transfer. After obtaining

the stolen data locally, the attacker must decompress it using the same process in order to restore the original data.

- **C2 Tunnelling**: Attackers use the C2 channel created in the preceding LM and Pivoting attacks to convey data while dealing with with a modest quantity of data, such as text files and numerical data [138]. The C2 channel is quicker and more reliable than DNS Tunnelling for data transmission, but the quantity of data that can be transmitted is restricted. It is necessary to split the files when come across a little bigger file, such as a 2-3 MB PDF, before transferring it.

- **DNS Tunnelling**: When confronted with huge files, the attacker creates a cloud server and transfers data to the outside network via the victim's DNS server [138]. In general, DNS tunnelling necessitates the presence of a DNS server on the infected domain. In order to do server-side tunnelling and target data download operations, the attacker must also control a cloud server. A DNS server is present in the network domain, as indicated in Fig. 3.4. The data to be leaked is submitted to the DNS server as a domain name query request in a particular format after being encoded by the attacker. The DNS server then passes the request to the attacker's cloud server, which is posing as the authoritative DNS server. After that, the attacker can download and decrypt data from the requests.

SCVIC-APT-2021 Generation

The SCVIC-APT-2021 dataset [79], which refers to the APT attack's six phases, contains a variety of APT attacks developed by the laboratory's internal computers and virtual machines in the OCI 5G ENCQOR Project Machine Learning-based Firewall-less Security Automation for the Network Edge. LM, Pivoting, and DE are three APT techniques that use a chain of operations to allow attackers to move between multiple network domains and steal data from internal computers users. In addition to the previous three attacks, Initial Compromise and Reconnaissance attacks are defined as data packets created during the attacker's initial attack and scouting phase. As a result, the SCVIC-APT-2021 dataset contains five APT network attacks as well as one type of normal traffic.

The network environment on which the SCVIC-APT-2021 relies on is depicted in Fig. 3.5. To manage the network, two network domains are formed in Virtual Box, and each network domain has a DC with administrator credentials. All managed machines have DC as the root user. Then, to facilitate inter-domain communication, Zero Tier, a specialized network VPN connection tool, is utilized to link two DC. Then, to imitate typical users in the network domain, multiple virtual computers are added to the two network domains.

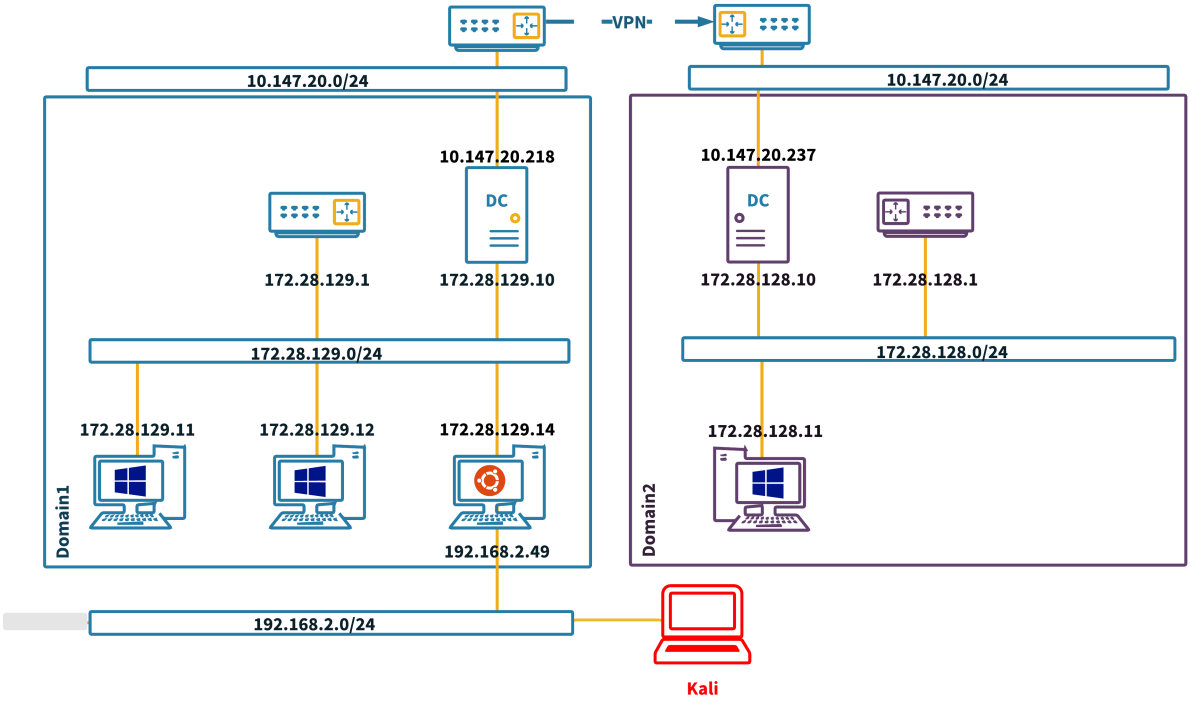


Figure 3.5: Network Domain Structure for Dataset Generation

Windows Server 2008 or Ubuntu 14.04 LTS are the operating systems used by these regular users. At the same time, the Ubuntu machine in Domain 1 is permitted to connect to the external network, putting it at risk of being the first victim. Finally, the Kali Linux PC is used to impersonate a attacker on a public network. Wireshark, a bit-by-bit packet capture and analysis software, is installed on all machines to continuously monitor and collect network traffic packets for later use.

The attacker's Kali Linux must first attack the external network interface in Domain 1, which is the Ubuntu 14.04 LTS, before launching the APT attack. This procedure generates data traffic that is labelled as Initial Compromise. During the APT attack, the attacker breaks into the two domains and steel all computer administrator credentials. Then, in each normal user machine, the attacker looks for high-value information files and returns them. After gaining access to the external network interface, the attacker can use the Ubuntu machine as a springboard in Domain 1 to carry out a LM attack and gain root privileges to additional systems. If the attackers succeed in cracking the Domain Controller, they will have access to sensitive data such as login credentials for all other

machines. Consequently, the network domain as a whole will be self-defeated. When the attacker gains root privilege on a computer, it may undertake Reconnaissance and DE attacks at the same time as the LM. As a result, there is no logical sequence to these three attacks. The attacker can use reconnaissance techniques to uncover the presence of Domain 2 once the DC in Domain 1 is compromised. The adversary then uses a Pivoting attack to break down network domain boundaries and go from Domain 1 to Domain 2. Similarly, attackers can carry out LM, Reconnaissance, and DE attacks in Domain 2 until it get administrator credentials for all machines or identify the needed information, as they could in Domain 1.

Domain 1 and Domain 2 are entirely penetrated by four rounds of independent APT attacks in order to ensure the universality of APT attacks. In four rounds of attacks of training set, the attack techniques and sequence are not the same. Different attack techniques demonstrated in Table 3.1 are combined to implement one round of APT attack according to the flexibility of APT attack. In some rounds, hackers initially infiltrated all computers in Domain 1 before moving on to machines in Domain 2. Hackers employ jump attacks and move laterally between Domain 1 and Domain 2 repeatedly in some rounds. Appendix A contains the precise attack trace and ground truth. Malicious data points are created in the training set using the malicious packets generated by the four rounds of attacks. Then, another round of the APT attack is employed to add malicious flows into the testing set.

Normal Traffic class consists of two components. The initial portion is produced when APT attacks are implemented. The second portion is comprised of the normal traffic of the 4SICS dataset [5]. The 4ICS dataset is designed to represent the normal traffic within an industry control system. HTTP, HTTPS, SSH, TCP, and UDP are among the communication protocols utilized by normal network traffic in the 4ICS dataset. Our lab also develops APT attack traffics with these protocols. Moreover, Industry control systems are among the primary targets of APT attacks. The industry control system is responsible for managing machine operations and serves as the brain of contemporary factories. The 4ICS dataset is therefore suitable for supplying normal traffic to the SCVIC-APT-2021 dataset.

When the APT attack is over, the PCAP files which are captured by Wireshark installed on all computers are delivered to CICFlowMeter-V3 for network feature extraction. CICFLowMeter-V3 is an Ethernet flow generator and analyzer that is frequently used in the field of cybersecurity to create datasets such as the Android Adware-General Malware dataset (CICAAGM2017) [72], the IPS/IDS dataset (CICIDS2017) [115], the Android Malware dataset (CICAndMal2017) [73], and so on. Each network flow receives 83 network flow features after being processed by CICFlowMeter-V3. The dataset is then labelled

Table 3.2: The dataset distribution of SCVIC-APT-2021

Dataset	Number of Records					
	Data Exfiltration	Initial Compromise	Lateral Movement	Normal Traffic	Pivoting	Recon
Training Set	527	73	729	254835	2122	833
Testing Set	74	77	142	55583	360	251

with the attack trace, schedule, and network address that are recorded during the breach. Finally, all hardware and time-related features are removed, leaving network traffic features and labels. The dataset distribution is shown in Table 3.2.

3.1.2 NSL-KDD

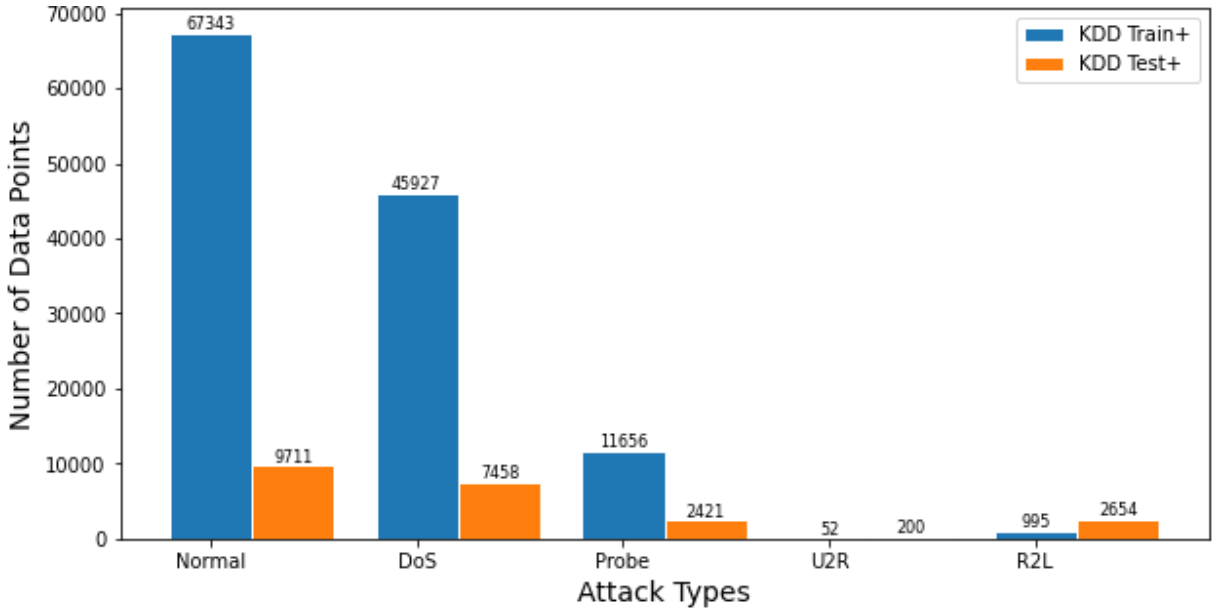


Figure 3.6: The Dataset distribution of NSL-KDD

DARPA was simulated by MIT Lincoln Labs in 1998, based on which, KDD99 was formed as a network intrusion dataset by extracting features from raw TCP dump data [84]. NSL-KDD removed redundant records and further improved the original KDD99; thereby,

Table 3.3: NSL-KDD Distribution

Dataset	Number of Records					
	Total	Normal	DoS	Probe	U2R	R2L
KDD Train+	125973	67343 (53%)	45927 (37%)	11656 (9.11%)	52 (0.04%)	995 (0.85%)
KDD Test+	22544	9711 (43%)	7458 (33%)	2421 (11%)	200 (0.9%)	2654 (12.1%)

both datasets have the same attack types and features, while the significant difference is the number of records under each class. Both datasets' attacks fall into four main categories: DoS, probing, R2L, and U2R, among which R2L and U2R are the most challenging for ML algorithms to detect. It is worth to note that the features in both datasets are not pure network-based features. The features consist of three groups: primary features of each network connection, content features of domain knowledge, and flow features within a two-second window. The primary features of TCP connections and the traffic features are network-based, whereas the content features can be considered as host-based features, e.g., system calls (number of operations on access files), shell logs, and FTP logs. The dataset distribution of NSL-KDD is demonstrated in Fig. 3.6 and Table 3.3.

3.1.3 UNSW-NB15

The University of New South Wales Canberra's Cyber Range Lab gathered the UNSW-NB15 dataset in 2015 [70]. Under modern network structures, the IXIA PerfectStorm tool is used to mimic and produce benign and malicious traffic. Since the dataset's production time is not far, this dataset includes a range of recent network attacks that are not present in the earlier dataset. Network traffic detection algorithms have been challenged by these new sorts of network assaults. Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms are among the nine forms of cyber attacks included in this dataset. Backdoors, Exploits, Reconnaissance, Shellcode, and Worms are examples of the APT attack included in this dataset. Each network flow in the dataset has 49 features. All features can be grouped as network features, basic features, content features, time features, general-purpose features, connection features, and labelled features.

UNSW-NB15 training set CSV and testing set CSV are created from a division of the original dataset as a training and testing set respectively. The training set has 175,341 records and the testing set contains 82,332 records from attack and normal categories.

The partitional training and testing set is used in the detection of the APT attack. The training set and testing set distribution can be found in Fig. 3.7.

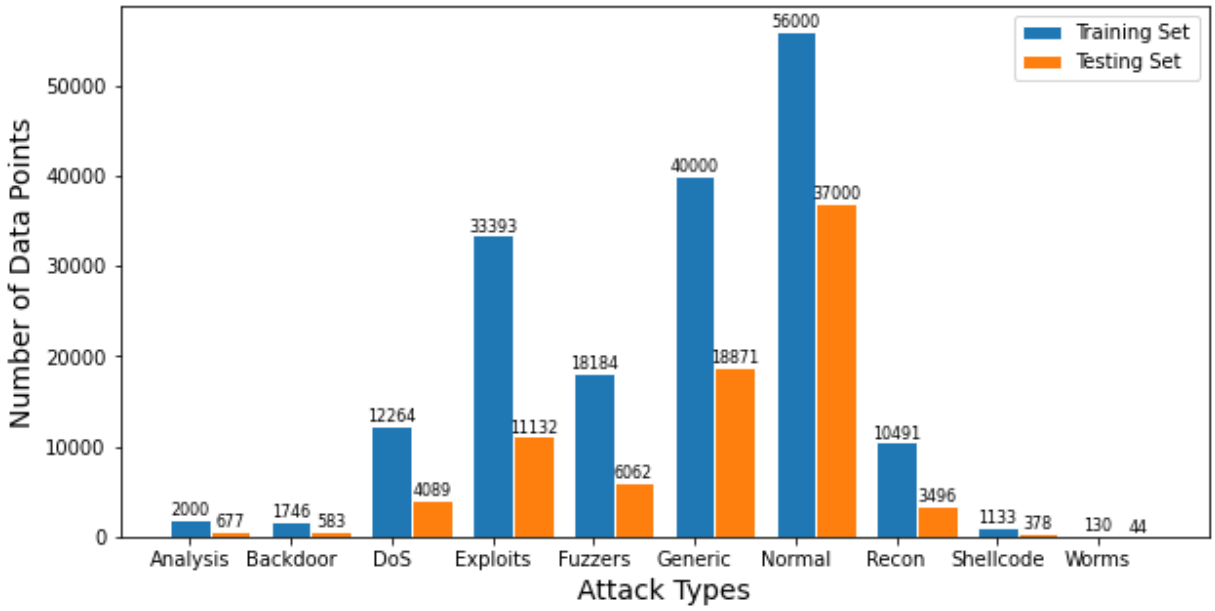


Figure 3.7: The dataset distribution of UNSW-NB15

3.2 Methodologies

The generation process of the SCVIC-APT-2021 dataset is introduced in the previous section. NSL-KDD, UNSW-NB15 dataset and SCVIC-APT-2021 are sent to the PKI and Progressive PKI models proposed in this section for network attack detection.

3.2.1 Feature Selection

To minimise detection time, reduce the risk of overfitting, and increase detection efficiency, feature selection methods are employed before attack detection to select features in the dataset that are more favourable to attack identification. Three filter approaches,such as Chi-square, Mutual Information and Analysis of Variance (ANOVA), are used to select features from the dataset.

The chi-square test's fundamental idea is to assess the accuracy of a theory by examining the difference between the real and theoretical values [77]. When doing so, it's common to assume the two variables are actually independent, and then look at the difference between the real and theoretical values. If the variation is small enough, the mistake is assumed to be a natural sample error produced by poor measuring procedures or an unintentional error. If this occurs, the two are indeed independent, and the hypothesis is accepted at this time; if the deviation is so large that it is unlikely that the error is caused by chance or inaccurate measurement, the two are thought to be relevant, and the hypothesis is rejected and the alternative hypothesis accepted.

The chi-square test examines the degree of dependency between stochastic variables. Employing this method may eliminate the features that are most likely to be independent from the class label and so irrelevant for classification. Chi-square feature selection mehtod calculate the chi-squared statistics for each non-negative feature and the class label. This score can be used to select the "k" best features from variables which have strong tights with the class label. The feature values must only comprise non-negative features like booleans or frequencies.

The Mutual Information (MI) between two random variables is a non-negative number that indicates how closely the variables are related [41]. It is 0 if and only if the two random variables are independent, while a larger value indicates greater dependency. The following assumption underpins the application of MI theory for feature extraction: items that appear frequently in one class but less frequently in other classes have stronger mutual information for that class. MI is a common metric while comparing features and classes. One feature fall within one group if they have the largest mutual information.

ANOVA is a statistical technique that is used to compare the means of two or more groups that are statistically distinct [44]. It employs F-test to determine whether there is a statistically significant difference between the groups. If there is no significant difference between groups and all variances are identical, the F-ratio for ANOVA will be close to 1.

3.2.2 Prior Knowledge Input Model

In the fields of machine learning and deep learning, knowledge-based models are critical. It offers a wide variety of applications in autonomous vehicle signal processing and data categorization [127]. The knowledge-based model is composed of two stages. The first stage is knowledge acquisition. There are no limitations on the method of acquiring knowledge. Neural networks, unsupervised learning, and supervised learning are allowed approaches for acquiring knowledge about a dataset. The second stage contains the standard machine

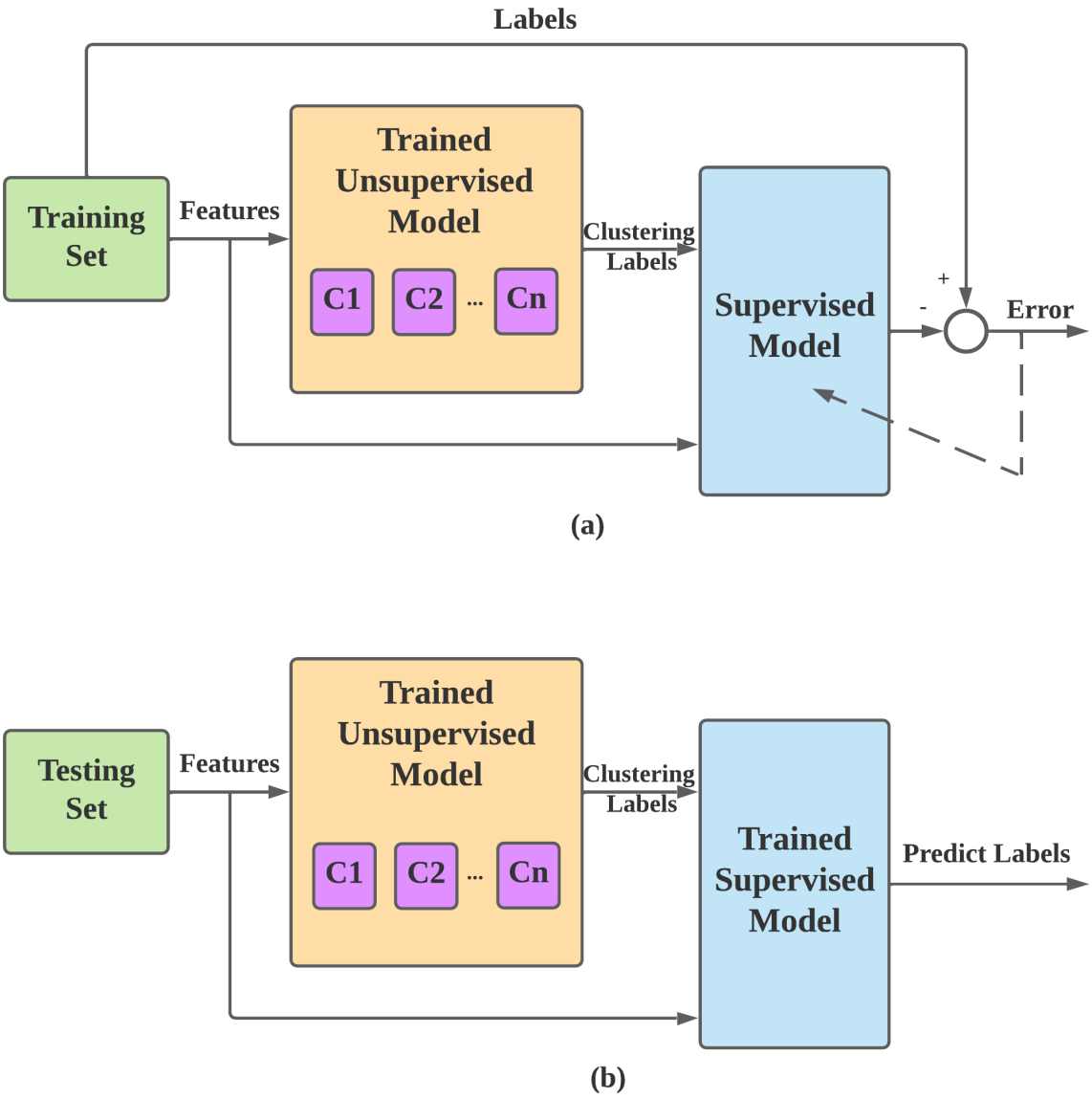


Figure 3.8: Training and testing phases of the PKI model

learning or deep learning models. It involves sending the previously acquired knowledge and the dataset's original features to the final model for regression or classification [136]. The

addition of knowledge to the PKI model can help the original model perform even better. This is because the primary purpose of knowledge is to lessen the complexity and duration of training. This simplifies the task of machine learning algorithms in determining the precision mapping between features and labels [124]. In this network traffic classification challenge, a significant amount of normal traffic and a small amount of network attack traffic are simultaneously delivered to the machine learning model for training. Due to the model's severe imbalance, it performs poorly at identifying network attack traffic. As a result, unsupervised pre-classification prior knowledge is contributed to the training process to increase the detection performance of network attack traffic.

The model's training and testing phases are depicted in Fig. 3.8. To begin, it is trained as an unsupervised technique for acquiring prior knowledge. The most critical parameter for unsupervised clustering algorithms is the number of clusters. This parameter specifies the number of classes into which unsupervised data should be grouped. To pick the number of clusters parameter, the validation set is segregated from the testing set. Following then, the PKI model's formal training begin. The training set's features are supplied into the trained unsupervised model in order to get prior knowledge, in other words, pre-classified labels. The pre-classified labels and features are then combined and fed into the training phase of the supervised model. The supervised model's error gotten from training phase is responsible for updating its parameters.

The testing phase is essentially identical to the training phase. The sole distinction is that the test phase is not required to determine the error and update the supervised model's parameters. During the testing phase, all data points' features are passed to the unsupervised model in order to gather prior knowledge. Following that, the supervised model is supplied with prior knowledge and features for ultimate decision-making.

3.3 Results

This section displays training details and experiment results for the SCVIC-APT-2021, NSL-KDD, and UNSW-NB15 datasets, with the goal of exposing readers to the optimal models and parameters selection process.

The Macro Average F1-score is used as the criterion for evaluation. As formulated in (3.1), it is defined as the average of F1 scores of all classes where F1 score denotes the harmonic mean of precision and recall parameters, i.e., $F1\ score = (2 \times Precision \times Recall)/(Precision + Recall)$. The macro average of F1-scores of all classes can be used to solve the problem of evaluation criteria caused by data imbalance.

Table 3.4: The Baseline One for SCVIC-APT-2021, NSL-KDD and UNSW-NB15

Dataset	Decision Tree	Random Forest	XGBoost
SCVIC-APT-2021 Macro Avg F1-score	0.6684	0.7529	0.7091
NSL-KDD Weighted Avg F1-score	0.7104	0.7020	0.7301
UNSW-NB15 Weighted Avg F1-score	0.7629	0.7785	0.7811

$$Macro\ Avg\ F1\ score = \frac{\sum_i^m F1\ score^{(i)}}{m} \tag{3.1}$$

Prior to feature selection, all features from the original dataset are sent directly to the three selected supervised algorithms, Decision tree, Random Forest and XGBoost, to generate Baseline One, which serves as a reference for the performance of proposed model. The Baseline One is demonstrated in Table 3.4. Due to the SCVIC-APT-2021's imbalance, the macro average F1-score is employed as the evaluation criterion. The number of data points in each category of the NSL-KDD and UNSW-NB15 datasets is roughly fair. As a consequence, the weighted average F1-score is utilized to evaluate the classification performance.

Following the acquisition of Baseline One, filter-based feature selection approaches such as Chi2, ANOVA, and Mutual Information are used to help simplify the dataset and eliminate redundant features. For each feature selection method, we go through all possible number of features and select the best one with highest F1-score on validation set. SCVIC-APT-2021, for example, contains 76 network features. Chi2, ANOVA, and Mutual Information are used to choose from 1 to 76 features from the dataset separately. After that, we evaluate the selected features' performance in the validation set to determine the optimal number of features with highest F1-score. Due to the unsatisfactory performance of decision trees on the SCVIC-APT-2021 dataset, further studies use solely Random Forest and XGBoost. The optimal number of features for the SCVIC-APT-2021, NSL-KDD and UNSW-NB15 are demonstrated in Table 3.5, 3.6 and 3.7. The bold rows in the tables represent the ideal combinations of supervised technique, feature selection method, and number of features. Their selected features are transmitted to the PKI model in order to boost the performance of network attack detection. These optimal combinations are used as Baseline Two for the following PKI model.

Table 3.5: The optimal number of features for SCVIC-APT-2021 dataset (Baseline Two)

Supervised Methods	Feature Selection	Optimal Number of Features	Macro Avg F1-score
Random Forest	**Chi2**	**51**	**0.8034**
	ANOVA	50	0.7738
	Mutual Info	12	0.7883
XGBoost	**Chi2**	**49**	**0.8092**
	ANOVA	19	0.7616
	Mutual Info	30	0.7310

Table 3.6: The optimal number of features for NSL-KDD dataset (Baseline Two)

Supervised Method	Feature Selection	Optimal Number of Features	Weighted Avg F1-score
Decision Tree	Chi2	33	0.7369
	ANOVA	**25**	**0.7449**
	Mutual Info	3	0.7371
Random Forest	Chi2	13	0.7205
	ANOVA	38	0.7106
	Mutual Info	**3**	**0.7530**
XGBoost	Chi2	17	0.7355
	ANOVA	29	0.7371
	Mutual Info	**3**	**0.7640**

Table 3.7: The optimal number of features for UNSW-NB15 dataset (Baseline Two)

Supervised Method	Feature Selection	Optimal Number of Features	Weighted Avg F1-score
Decision Tree	Chi2	18	0.7677
	ANOVA	42	0.7629
	Mutual Info	**12**	**0.7724**
Random Forest	Chi2	19	0.7803
	ANOVA	42	0.7785
	Mutual Info	**12**	**0.7832**
XGBoost	Chi2	19	0.7857
	ANOVA	31	0.7828
	Mutual Info	**18**	**0.7867**

Table 3.8: The PKI model results for SCVIC-APT-2021 dataset

Supervised Method	Feature Selection	Number of Features	Unsupervised Methods	Optimal Number of Clusters	Macro Avg F1-score
RF	Chi2	51	KMeans	17	0.8033
			GMM	**5**	**0.8103**
XGBoost	Chi2	49	KMeans	1	0.8092
			GMM	1	0.8092

Table 3.9: The PKI model results for NSL-KDD dataset

Supervised Method	Feature Selection	Number of Features	Unsupervised Methods	Optimal Number of Clusters	Weighted Avg F1-score
DT	ANOVA	25	KMeans	13	0.7493
			GMM	6	0.7415
RF	Mutual Info	3	KMeans	11	0.7552
			GMM	13	0.7564
XGBoost	Mutual Info	3	KMeans	12	0.7641
			GMM	**10**	**0.7642**

Table 3.10: The PKI model results for UNSW-NB15 dataset

Supervised Method	Feature Selection	Number of Features	Unsupervised Methods	Optimal Number of Clusters	Weighted Avg F1-score
DT	Mutual Info	12	KMeans	2	0.7736
			GMM	7	0.7793
RF	Mutual Info	12	KMeans	9	0.7830
			GMM	**13**	**0.7920**
XGBoost	Mutual Info	18	KMeans	15	0.7875
			GMM	4	0.7882

The PKI experiment is dependent on the previously selected features. Unsupervised techniques are used to pre-classify the selected features produced from feature selection, and then supervised methods are used to predict the final categories using the pre-classified labels and selected features. It is critical for unsupervised methods to pick the appropriate number of clusters. The validation set is utilized to determine the number of clusters for unsupervised algorithms as well. In this experiment, K-Means and GMM are two candidates for unsupervised methods. The PKI model results on SCVIC-APT-2021, NSL-KDD and UNSW-NB15 are illustrated in Table 3.8, 3.9 and 3.10. The bold rows in the tables are the highest performance that PKI model can reach. For SCVIC-APT-2021 dataset [79], the best macro average F1-score is 0.8103. The highest weighted average F1-score for NSL-KDD and UNSW-NB15 datasets are 0.7642 and 0.7920 separately. The comparison of Baseline One, Baseline Two and PKI model's results are depicted is Fig. 3.9, 3.12 and 3.13. Baseline One represents the primitive results which do not include any feature selection or unsupervised model. The original features from the dataset are fed into the supervised model directly. Baseline Two demonstrates the results after the optimal features are selected from the feature selection techniques. During the testing phase on the SCVIC-APT-2021 dataset, the DT is dropped in the following experiment from the supervised candidates list because of the limited performance on Baseline One.

Fig. 3.10 and 3.11 show the confusion matrices for Baseline One and XGBoost PKI model, respectively. The classification results for Data Exfiltration and Reconnaissance have been shown to be greatly enhanced. The misclassified data points for Initial Compromise, Lateral Movement, and Pivoting might be ignored in light of the substantial enhancement for Data Exfiltration and Reconnaissance. For this reason, the total perfor-

mance of the PKI model is superior to that of the Baseline One and Baseline Two.

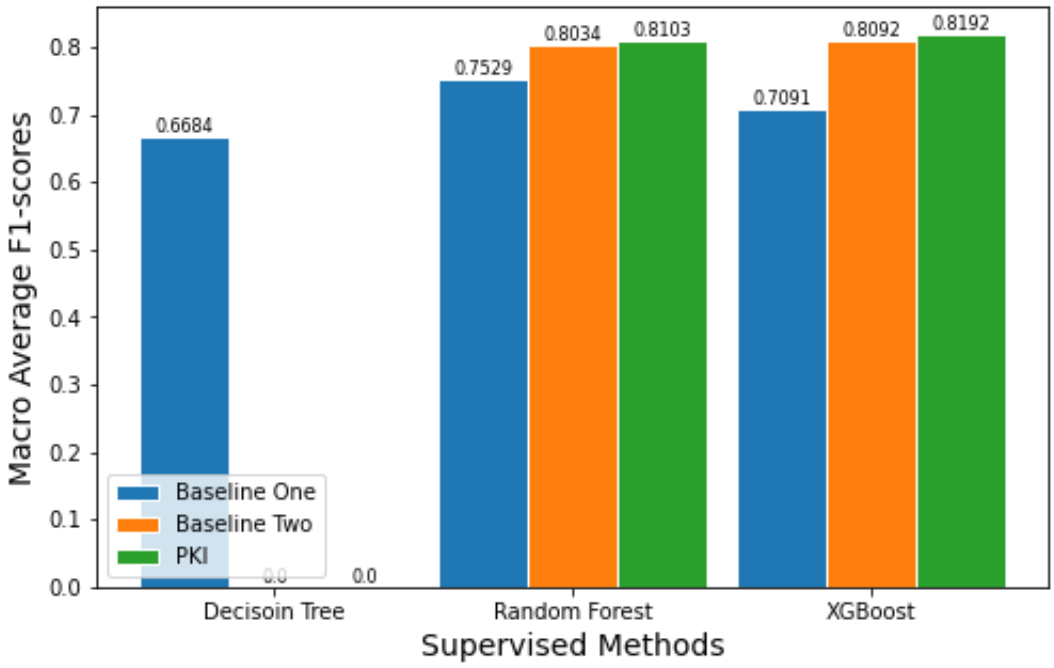


Figure 3.9: The results comparison between Baseline One, Baseline Two and PKI model on SCVIC-APT-2021

3.4 Conclusion

This chapter proposes a knowledge-based PKI paradigm for improving network attack detection performance. Comparing the SCVIC-APT-2021 dataset for detecting APT attacks and the NSL-KDD and UNSW-NB15 datasets as benchmarks to the baselines demonstrates that the PKI model significantly improves the detection performance of network attacks. While compared to the SCVIC-APT-2021 datset's Baseline One, the macro average F1-score can be increased by 11.01% when using XGBoost with PKI model. The PKI model can enhance weighted average F1-score by 5.44% and 1.35%, respectively, for the NSL-KDD and UNSW-NB15 datasets. This chapter collects and applies solely network-based features to the attack detection phase. However, the host events that occur as a result of the attacks are disregarded. This indicates that there is still room for improvement in the

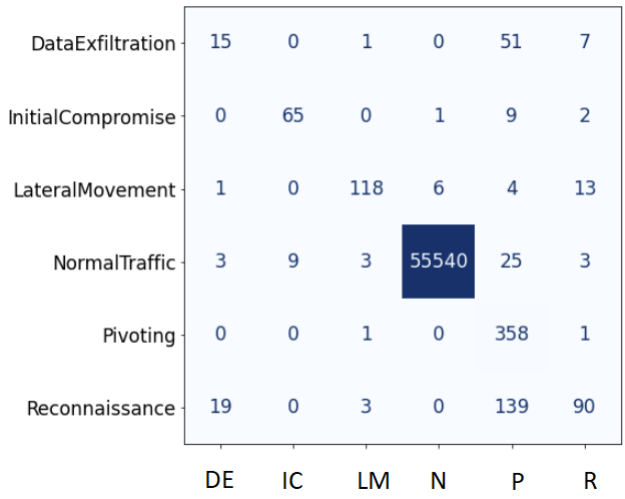


Figure 3.10: The confusion matrix for Baseline One on the SCVIC-APT-2021 dataset

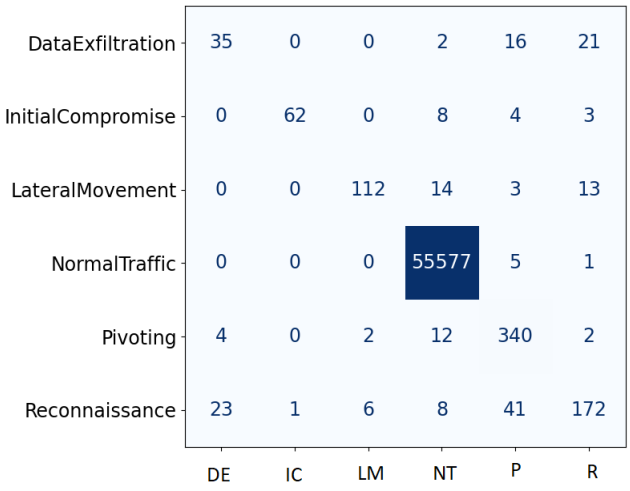


Figure 3.11: The confusion matrix for the XGBoost PKI model on the SCVIC-APT-2021 dataset

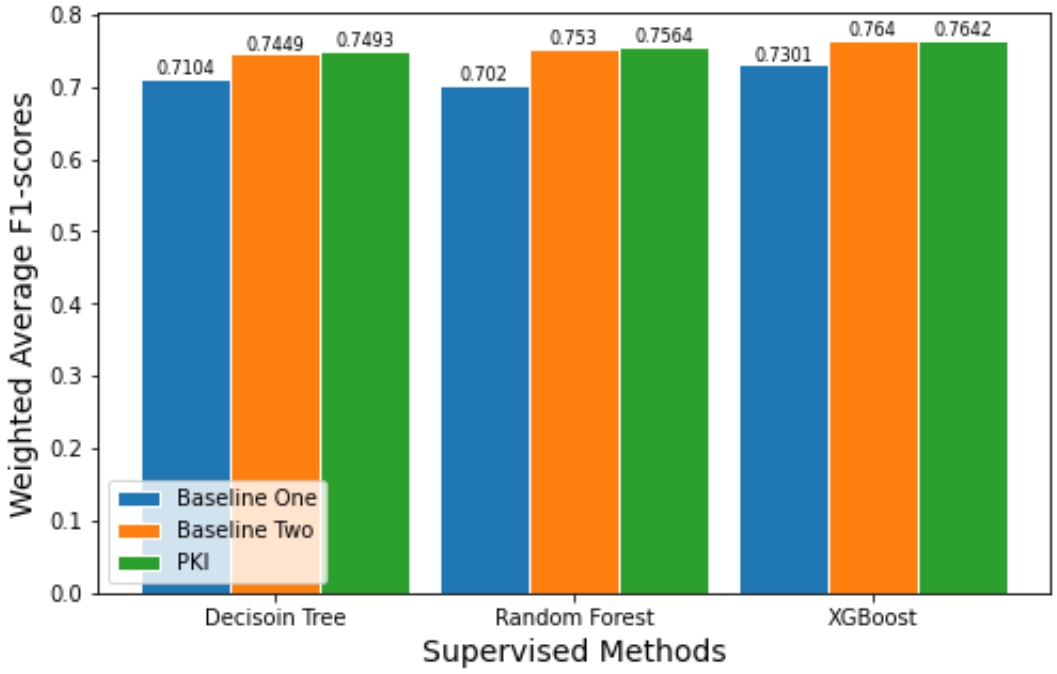


Figure 3.12: The results comparison between Baseline One, Baseline Two and PKI model on NSL-KDD

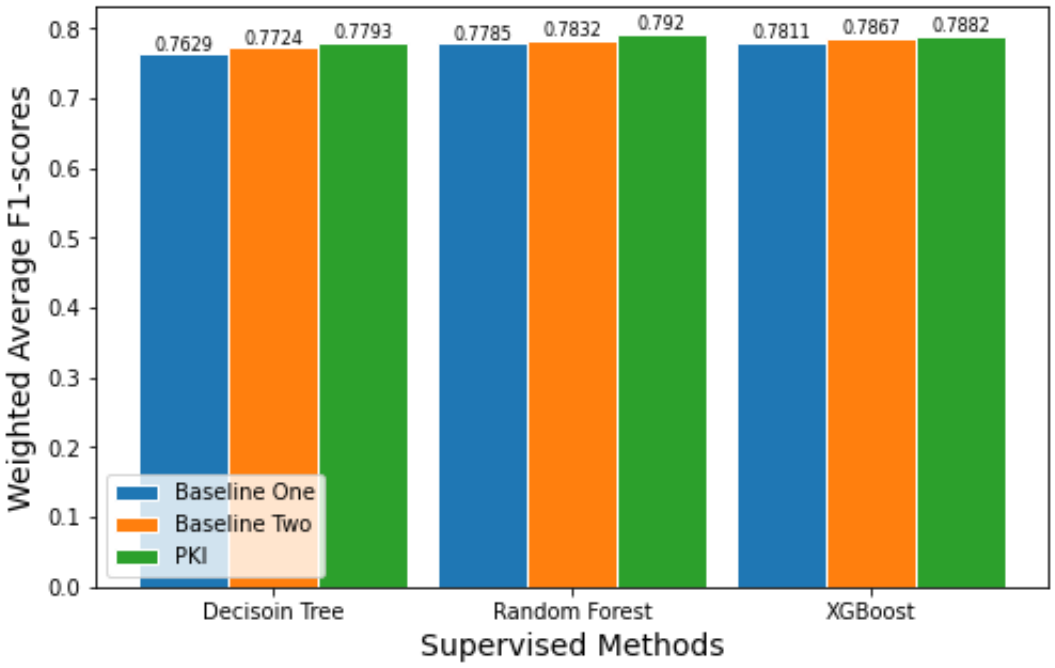


Figure 3.13: The results comparison between Baseline One, Baseline Two and PKI model on UNSW-NB15

detection of network attacks. The following chapter considers host events as part of the detection features in order to improve the model's performance.

Chapter 4

Prior Knowledge-Based Intrusion Detection on Network Flows and Hosts

By monitoring the features of network traffic, network-based intrusion detection identifies network attacks entering and departing domains and devices. By analyzing the host events and configuration snapshots created by the system, host-based intrusion detection identifies intrusions that have penetrated the host devices. The discussion of these two forms of IDS can be found in section 1.1.1. NIDS is often used for intranet import and export node, but HIDS can be used on any device. The two IDS together to provide security solutions for businesses and governments' internal networks.

Although they can cooperatively protect a network against cyberattacks at the same time, their work processes are unaffected. In other words, NIDS is solely responsible for identifying the network flow features of threats; it does not control the impact of network traffic on devices [83]. The source of the attack is ignored by HIDS, which evaluates if the device is under attack internally. This method of operation is distinct from genuine cyber-attacks. When the network attack is active, the attack packets modify the system settings on a local device, resulting in a local event record. The separate functioning modes of NIDS and HIDS, on the other hand, are unable to manage the relationship between network flow and host events, causing vital evidence of network intrusion to be overlooked [16]. We adopt the combination mechanism developed in the OCI 5G ENCQOR Project Machine Learning-based Firewall-less Security Automation for the Network Edge to boost the performance of IDS.

Furthermore, the SCVIC-CIDS-2021 dataset created from the raw data from CSE-CIC-IDS2018 is utilized to assess the efficacy of PKI and PKID models in detecting network threats.

4.1 Dataset

Researchers at the University of New Brunswick produced the CSE-CIC-IDS2018 dataset to address the drawbacks of static and one-time datasets, which cannot reflect the features of today's networks [74]. There are seven different attack scenarios in the dataset. The seven different scenarios of attacks are broken down into fourteen different attack techniques. When referencing CSE-CIC-IDS2018, publisher's created network traffic data files are not utilized. Instead, the original data, network traffic PCAP files and system event logs are utilized to design the network-based and host-based features combined hybrid dataset which called SCIVC-CIDS-2021.

First, CICFlowMeter-V3 is used to extract network flow features from PCAP files for the network flow part. The attack trace and timestamp supplied by the dataset publisher are used to label the network traffic features. 14 types of attacking techniques are used as the labels. Benign flow with the same quantity as the malicious flow is generated by picking the same percentage from each file due to the excessive number of normal flows in the original PCAP files.

Next is the feature extraction and feature engineering for host event logs. The Get-WinEvent tool in Windows PowerShell is used to read host event logs and parse them into host event features and save them into CSV files. At the same time, according to the timestamp and source-destination IP address pair, host events are synchronized with network flows in the same time period. There may be several host events during some network flow times, while other network flow periods may not correlate to any host events. Only the data points that include the host event are kept for further evaluation of the influence of host events on detection performance.

For each host event, the CSV files retrieved from host event logs contain eight numerical characteristics and a paragraph of description words. The first eight numerical host features and description words are separated into two pieces for processing.

The transformer encoder receives the numerical host features directly and encode them into a 28×8 matrix. This is because there are at most 28 host events in the time span of the same network flow. If there are less than 28 host events matching to network flow,

Table 4.1: The class distribution of SCVIC-CIDS-2021

Attack Type	Training Set	Testing Set	Total
Benign	308375	152172	460547
Bot	60767	29693	90460
DDOS-HOIC	137147	67449	204596
DDOS-LOIC-HTTP	39019	19166	58185
DDOS-LOIC-UDP	760	342	1102
DoS-GoldenEye	2271	1163	3434
DoS-Hulk	13388	6553	19941
DoS-SlowHTTPTest	10579	5351	15930
DoS-Slowloris	1394	702	2096
FTP-BruteForce	32222	15918	48140
SSH-Bruteforce	11143	5418	16561
Sum	617065	303927	920992

the field is filled with 0. The processed host numerical features are referred to as event features.

The description words are passed to the transformer encoder [143] and Bidirectional Encoder Representations from Transformers (BERT) [35], a transformer-based machine learning approach for Natural Language Processing (NLP). After processing, each data point's event message feature is a 768×100 matrix called Event Messages.

In the CSE-CIC-IDS2018 raw dataset, the PCAP files and host event logs are processed as network-based features and host-based features, respectively. Event Features and Event Messages are two types of host-based features. The structure of the dataset built using CSE-CIC-IDS2018 original data is depicted in the Fig.4.1. Flow Features, Event Features and Event Messages are fed into the machine learning model to complete a comprehensive intrusion detection. Table 4.1 illustrates the class distribution among the SCVIC-CIDS-2021 dataset.

SCVIC-CIDS-2022 is developed in order to validate the network and host integrated detection approach. It is based on the metadata of NDSec-1 obtained by the University of Applied Sciences Fulda. The raw data processing mechanism is identical to that used in SCVIC-CIDS-2021. CICFlowMeter [51] is used to extract network features from network packets. BERT receives host events and host messages for data encoding. Following processing, the shape of host event features is 2182×8, and the shape of host event messages is 512 × 768, where 512 is the maximum length of an input that BERT accepts. Table 4.2

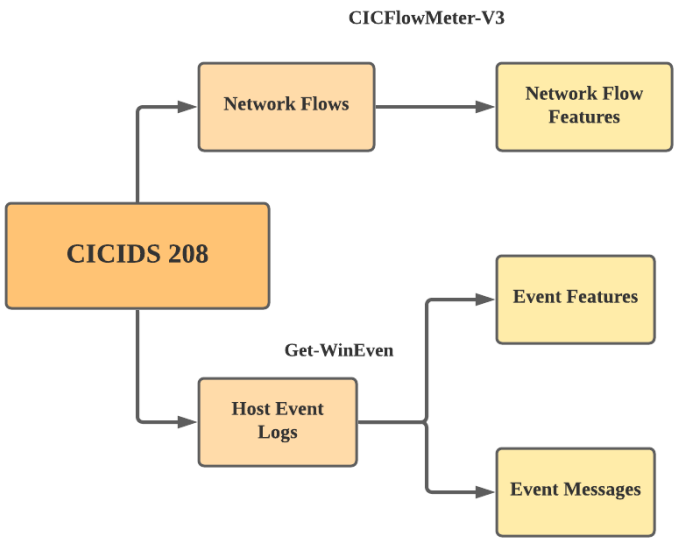


Figure 4.1: CSE-CIC-IDS2018 Dataset Structure

demonstrates the data points' distribution in SCVIC-CIDS-2022 Dataset.

4.2 Methodologies

SCVIC-CIDS-2021 dataset is generated by integrating network flow and host events, host message, using raw network flow and host logs from the CSE-CIC-IDS2018 dataset, as indicated in the preceding section. The data that is applied feature derivation, and PKI models are used to make final decisions based on this, i.e., the BERT model assesses if the system is under cyber attack based on three dimension reduced parts of features. The entire model may be separated into two halves. First of all, the Autoencoder (AE) and Gated Recurrent Unit (GRU) transform the network-based features and host-based features in each data point into dimension reduced and flattened one-dimensional vectors before being sent to the PKI model. This forms the feature derivation section. Secondly, for cyber attack detection, all of the data points are fed into the PKI model. In the subsections that follow, implementation details of the model are presented.

Table 4.2: Data Distribution of SCVIC-CIDS-2022

Attack Type	Training Set	Testing Set	Total
Botnet	52	40	92
BurteForce	2150	1006	3156
DOS	12677	6389	19066
Exploit	4	2	6
Malware	22	12	34
MISC	31	18	49
Normal	4716	2284	7000
Probe	777	307	1084
WebAttack	18	12	30
Sum	20447	10070	30517

4.2.1 Feature Derivation

In the introduction of the previous part, the network flow part of the CSE-CIC-IDS2018 dataset have 132 features. For host-based features, the event feature is a 28×8 matrix, and the event message is a 768×100 matrix in each data point. If these data are sent directly into the PKI model, it will cause too long training time and poor model performance. Feature derivation is used to overcome these difficulties and ensure that the data does not lose information. In this section, feature selection, AE and GRU will be discussed as options for feature derivation.

Data with reduced dimension is the process of generating an appropriate sub-feature set from the original feature set. This is to simplify the dataset, enhance the model performance, and speed up the training process. In this procedure, the feature selection algorithm is crucial. They compute the feature importance factor for all features in the algorithm, sort them from largest to smallest, and then pick a limited number of the highest-ranking features. The feature selection technique can perform effectively when the relation between features are reasonably independent and do not contain sequence information. However, if the features contain critical sequence information, such as a phrase, the feature selection method may lose the important sequence information. The SCVIC-CIDS-2021 dataset –created from the original data– contains sequence information in the event features and event messages. As a result, deep learning techniques are used to reduce the dimensions of the three portions of the dataset, with the goal of conserving as much data information as possible while not destroying the sequence information.

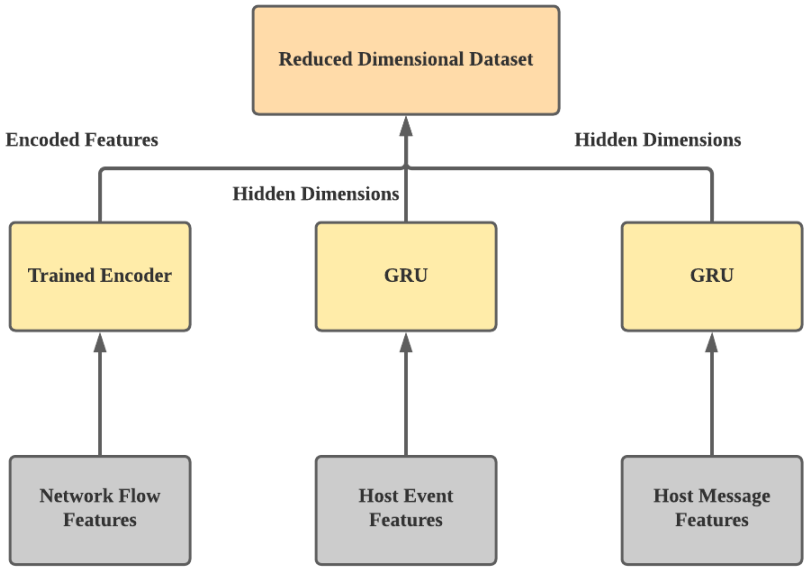


Figure 4.2: Feature Derivation

AE, as a commonly used method of encoding feature derivation, is first considered for the CSE-CIC-IDS2018 dataset. It is comprised of two parts: an encoder and a decoder [92]. The encoder is in charge of data encoding, which compresses and reduces the dimensions of a given data point to a set of hidden states. The decoder is in charge of receiving the encoder's concealed states and attempting to recover the original data using a lower amount of data. This necessitates that the AE's input and output have vectors of the same length. AE is an unsupervised algorithm. More precisely, it is a self-supervised algorithm as there is no need to use the predicted or true label to produce errors in order to update the parameters throughout the training phase. A decoder is utilized to determine the errors by comparing the restored data to the original data so to update the training parameters. Three particular designed AE are applied to each of the three components in the dataset to reduce their dimensions.

During the implementation phase, the shortcomings of AE are uncovered. AE is capable of performing effectively in the face of network flow-based features. The performance of the detection algorithm is essentially the same after AE reduces the dimensions of the data points in the training and validation sets. The performance of AE for feature derivation with host-based features is not satisfactory. AE can only decrease the dimension of host-based features to dozens of independent values by comparing the number of unique

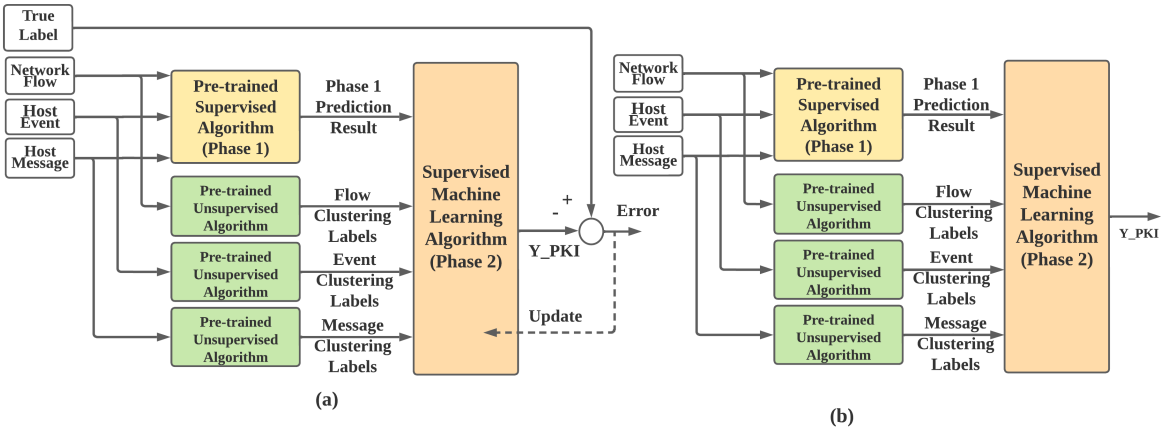


Figure 4.3: Prior Knowledge Input Model Structure

values in all data points after feature derivation. This phenomena is incompatible with the wide range of host-based features. Meanwhile, the large difference of the detection performance on the validation set before and after feature derivation supports this hypothesis. According to the outcomes, there might be two primary reasons. First, the structure of the AE is unsuitable. The dimensions of host message features start at 768×100 and are subsequently lowered from 76800 to 128 directly. Indeed, there is a possibility of information loss. Second, host-based features, particularly host messages, include essential sequence information that AE does not consider throughout the encoding and decoding process. The matrix is simply transformed into a vector, and the data points are treated as independent values. As a result, the AE needs to be replaced with another technique to reduce the dimensions of host-based features.

GRU [31] which is a type of Recurrent Neural Networks is selected to reduce the dimension of host-based features, so sequence information is taken into account. GRU is a suggested solution to the issues of long-term memory and gradients in back propagation, similar to LSTM method. The information included in all previous data points is held in the hidden dimensions from the beginning to the end. When compared to LSTM, GRU runs with fewer parameters, so provides a quicker learning rate, and almost identical model performance in other areas [153]. As a result, GRU is selected for deriving the host-based features. The number of GRU outputs is set to the number of classes in the dataset throughout the training procedure. This implies that the GRU learns by classifying data points. At the same time, the output of the hidden dimensions in the GRU maintains the

information regarding the data points after feature derivation. While the GRU classification results are close to the original values, the hidden dimensions gradually complete the data feature derivation in order to replace the original large quantity of data with less hidden dimensions. On the validation set, the performance difference of host-based features before and after feature derivation is not instantly obvious. As a result, instead of AE, GRU is a better option to reduce the dimension of host-based features.

AE and GRU are used to decrease the dimensions of network flow-based features and host-based features, respectively based on the above argument. Fig.4.2 depicts the flow chart of the feature derivation. The total dimension of each data point is around 100 after the three portions of the dataset have been treated for feature derivation. The existing dimensions are acceptable for the PKI model, especially when compared to the tens of thousands of original dimensions.

4.2.2 Prior Knowledge Input Model

To enhance the cyber attacks detection performance, PKI method is proposed in this study. The PKI model [136] incorporates existing feature information as prior knowledge into the training process. Following upon that, fewer features are required to obtain similar or better training results, reducing training complexity and increasing training efficiency [124]. Knowledge-based modelling is a common approach for collecting existing knowledge. In the proposed methodology, the pre-classification labels of unsupervised algorithms are added as prior knowledge to the training process of the dataset to achieve a better detection performance on cyber attacks.

One of the key parameters that substantially affects the performance of the unsupervised approach is the number of clusters. The clustering results provides knowledge of similarity about some group of data. Using this extra knowledge, clustering can assist the supervised method to reduce the complexity of the problem. In other cases, if clustering technique classifies data points that are highly different in the same cluster, the clustering results will have a negative influence on the final output of the supervised algorithm, resulting in performance decreasing of the model. As a result, Silhouette score, a metric with a value range of -1 to 1, is used to evaluate the clustering performance of the unsupervised method. The closer the Silhouette Score to 1, the more distinct and well-separated the clusters are. On the other hand, the closer the score gets to -1, the more incorrectly the clusters are assigned. If the value is close to 0, it implies that the distance between the clusters in the classification result is close, and hence the distribution of data points cannot be accurately represented.

The training and testing phases of the PKI model are illustrated in Fig. 4.3. The silhouette score is used before the training stage to determine the optimal number of clusters for the three parts of the dataset. The best number of clusters parameters for network flow features, event features, and host messages are not always the same because they are three independent elements. The entire dataset is fed into the supervised algorithm in phase 1 to obtain the coarse predictions for phase 2 training phase. The pre-classified labels from unsupervised methods are used as prior knowledge to train supervised machine learning algorithm in phase 2 as well. The coarse predictions from the supervised machine learning algorithm in phase 1 and the prior knowledge from the unsupervised approach are combined and fed into the supervised algorithm in phase 2 for fine classification.

To obtain testing data for phase 2, host event and message features of test data are fed into the pre-trained supervised method and the unsupervised methods in the test stage. The collected coarse results from phase 1 and the prior knowledge from unsupervised methods are processed by trained supervised machine learning algorithm in phase 2 for ultimate decision-making. The fine classification results from phase 2 demonstrate the prediction performance of the multi-class attack data.

4.2.3 Prior Knowledge Input Difference Model

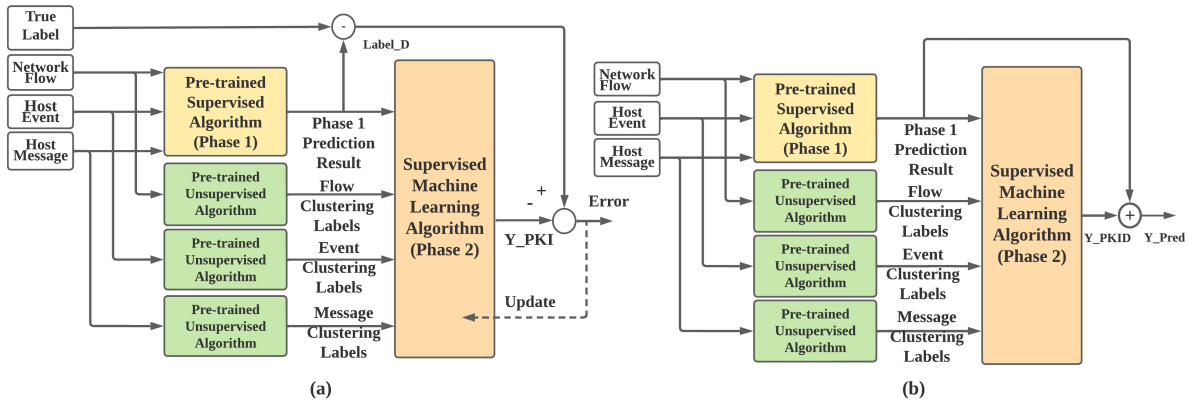


Figure 4.4: Prior Knowledge Input Difference Model Structure

To further enhance the detection model's performance, the PKID model based on PKI is presented. The PKID model also contains two phases for detecting attacks, but their purposes are distinct. The training and testing stages of PKID are illustrated in Fig.

4.4. Phase 1 and unsupervised models are trained similarly to PKI models, that is, the dimensionality reduced features are directly supplied to the ML models for training in order to get coarse classification results and prior knowledge. The distinction of PKID from PKI is in Phase 2. The real labels are utilized as the target for Phase 2 in the PKI model, whereas the differences between the real labels and the coarse results are used as the target for Phase 2 in the PKID model. In other words, Phase 2 of the PKID model predicts the gaps between the coarse classification results and the real labels, not the true category of data points. As a result, the model is referred to as the difference model.

The testing stage begins with sending the dimensionality-reduced features to Phase 1 and unsupervised models to acquire coarse classification results and prior knowledge. They are then transmitted to Phase 2 for difference prediction, which determines the difference between coarse classification results and true labels. Finally, Phase 2's output and coarse classification results are merged to get the fine classification results which are the categories of the data points.

4.3 Result

Prior to sending the dataset's features to the PKI and PKID models, AE and GRU are used to reduce the data points' dimensionality. Network-based features are sent to AE in order to extract hidden states. Host event features and host message features are delivered to GRU to reduce the dimension. DT, RF, and XGBoost are employed as supervised method candidates for final classification in this experiment. The performance of the dataset in the validation set after dimensionality reduction is utilized as a criterion for determining the ideal dimensions for three parts of the dataset for each supervised technique. Below, we use the DT's dimensionality reduction procedure to demonstrate the data dimensionality reduction process in detail.

Due to the generated dataset employs network-based features as the leading role and host-based features as an auxiliary, we begin by using AE to reduce the dimensionality of network-based features. The value of the AE hidden dimension is an array ranging from 5 to 125, with a step of 5. When the hidden dimension is set to 40, the network-based features perform optimally on the validation set. The macro average F1-score is 0.9154. In the meantime, the trained DT model's performance on the testing set is 0.8898.

Following that, the dimensionality of host event features is reduced based on the network-based features dimensionality reduction. When GRU is used to reduce dimensionality, the hidden dimension's value is an array ranging from 10 to 200 with a step of

Table 4.3: The macro average F1-score for three supervised candidates. The upper row represents the macro average F1-score and the lower row represents the correspondence optimal number of features for each supervised candidates

Supervised Model		Flow	Flow + Event	Flow + Event + Msg
DT	Macro F1-score	0.8898	0.9437	**0.9611**
	Optimal Features	40	40+10	40+10+10
RF	Macro F1-score	0.8585	0.9069	0.9269
	Optimal Features	35	35+5	35+5+40
XGB	Macro F1-score	0.8898	0.9348	0.9550
	Optimal Features	60	60+40	60+40+5

10. After dimensionality reduction, the host event features are concatenated to the hidden dimensions of network-based features. When the hidden dimension is 10, that is, when the data point has 40 hidden dimensions from network-based features and 10 hidden dimensions from host event features, DT performs the best in the validation set, with 0.9468 for the macro average F1-score. At the time, the model's performance on the test set was 0.9437.

Finally, on the basis of the prior two steps, the dimensionality reduction of host message features is conducted. GRU's hidden dimension begins at 10 and ends at 130, with a step of 10. Similarly, after dimension reduction, the host message features are concatenated to the previous two parts for joint testing. When the hidden dimension of the host message features is set to 10, the DT performs optimally on the validation set, with 0.9538 for the macro average F1-score. At the moment, the model's performance on the testing set is 0.9611.

Table 4.3 demonstrates the dimensionality reduction details for DT, RF and XGBoost. It is find that adding host event and host message features considerably improves the performance of all supervised techniques in the testing set. This demonstrates that network attack classification is beneficial from the combination of network-based features and host-based features. To simplify following procedures, we send only the best-performing DT and related dimensionality reduction features to the PKI and PKID models.

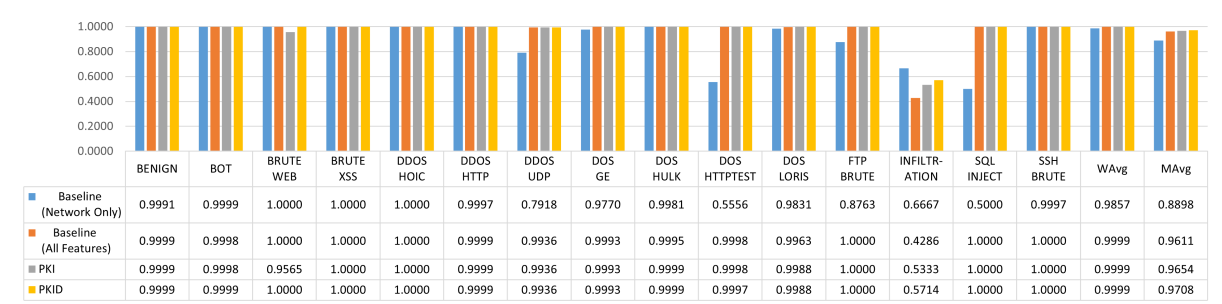
To perform feature pre-classification, K-Means is employed as unsupervised candidate for PKI and PKID. We use the Within-cluster Sum of Square (WSS) and the Silhouette Score to find the best number of cluster models for unsupervised candidates. If there is an apparent elbow in the WSS curve, the number of clusters at the elbow is selected. If the WSS curve does not have an evident elbow, the number of clusters with the highest

Table 4.4: The optimal number of clusters of K-means for network-based features and host-based features

Unsupervised Methods	Number of Clusters for Network Flow Features	Number of Clusters for Host Event Features	Number of Clusters for Host Msg Features
K-means	29	3	23

Silhouette Score is considered ideal. Appendix B demonstrates the curves of WSS and Silhouette Score. The optimal number of clusters for K-Means is depicted in Table 4.4. When the number of clusters for network features, host event features and host message features are 29, 3 and 23 respectively, the corresponding unsupervised algorithms obtain the highest silhouette score.

The macro average F1-score for the PKI and PKID models is demonstrated in Fig. 4.5. The blue baseline indicates the feature derivation results of DT only with the network flow features and the orange baseline shows the feature derivation results of DT with all three parts of the generated dataset. It is found that the macro average F1-score is increased by 7.13% when the host-based features are integrated into the attack intrusion scheme. The DDOS-UDP, DOS-HULK, FTP-BRUTEFORCE and SQL Injection detection performance benefit from the new-added host-based features. On the other hand, it is clear that both PKI and PKID models can boost the detection performance of network attacks when compared to the Baselines. The best macro average F1-score is 97.08% when the K-means-based PKID model is used for detection.



	BENIGN	BOT	BRUTE WEB	BRUTE XSS	DDOS HOIC	DDOS HTTP	DDOS UDP	DOS GE	DOS HULK	DOS HTTPTEST	DOS LORIS	FTP BRUTE	INFILTR-ATION	SQL INJECT	SSH BRUTE	WAvg	MAvg
Baseline (Network Only)	0.9991	0.9999	1.0000	1.0000	1.0000	0.9997	0.7918	0.9770	0.9981	0.5556	0.9831	0.8763	0.6667	0.5000	0.9997	0.9857	0.8898
Baseline (All Features)	0.9999	0.9998	1.0000	1.0000	1.0000	0.9999	0.9936	0.9993	0.9995	0.9998	0.9963	1.0000	0.4286	1.0000	1.0000	0.9999	0.9611
PKI	0.9999	0.9998	0.9565	1.0000	1.0000	0.9999	0.9936	0.9993	0.9999	0.9998	0.9988	1.0000	0.5333	1.0000	1.0000	0.9999	0.9654
PKID	0.9999	0.9999	1.0000	1.0000	1.0000	0.9999	0.9936	0.9993	0.9999	0.9997	0.9988	1.0000	0.5714	1.0000	1.0000	0.9999	0.9708

Figure 4.5: The macro average F1-score comparison between Baseline, PKI and PKID model

4.4 Conclusion

In this chapter, we propose a method for network attack detection that combines network-based and host-based features. The addition of host event and host message features, according to data analysis, can greatly increase performance. When only network-based features are used, the macro average F1-score for the SCVIC-CIDS-2021 dataset can reach 0.8898. When host-based features are included in the model's decision-making, the macro average F1-score can be enhanced to 0.9611. The integration of host-based features enhances the identification of attacks that can cause host events within the computer. The incorporation of unsupervised pre-classification prior knowledge considerably increases the supervised model's detection performance. Furthermore, the model's second phase predicts the difference between the pre-classification results and the real label in PKID which enables it to identify network attacks more effectively. The best macro average F1-score 0.9708 is achieved by the K-Means based PKID model.

Chapter 5

Conclusion and Future Directions

5.1 Conclusion

As network structure and communication technologies progress, an increasing amount of critical data is transferred across the network or kept in the computer after being digitized. Network security technology is highly appreciated by numerous corporations, organisations, and even individuals in order to maintain a clean and safe network environment. Unlike a standard firewall, which restricts network traffic's entry and outflow, an IDS safeguards the internal network and devices by network traffic analysis and system log warnings. IDS can be classified into NIDS and HIDS based on their operational methodologies.

The advancement of machine learning and deep learning has increased the intelligence of computers, relieving humans of time-consuming data processing duties. Without machine learning, network managers must go through system records and analyse large amounts of network data packets to glean information about threats. However, the intelligence of machine learning enables computers to identify and respond to attack traffic automatically. In comparison to human recognition, machine learning is faster and more accurate, allowing it to detect attacks in their early stages and prevent additional intrusions.

As a result, this thesis focuses on developing rapid and reliable cyber attack defence strategies through the use of machine learning and deep learning models. The research is separated into two parts. The first part analyse network traffic using machine learning to achieve the identification of APT attacks. The second part uses machine learning and deep learning to identify network attacks using network flows and host events. A knowledge-based methodology is presented to further enhance cyber attack detection performance in both parts.

For the first part, utilizing network traffic to detect APT attacks, due to a dearth of datasets for APT attacks, the majority of current research is focused on early data sets such as NSL-KDD or KDD99. These data sets comprise a single or a limited number of APT attacks, and network traffic does not match the characteristics of the current network due to the early creation period. As a result, utilising these datasets to detect APT attacks is not the optimal solution. To address the deficiencies of the early dataset, SCVIC-APT-2021 is produced based on the six stages of the APT attacks. To begin, multiple network users are deployed in two network domains that are connected by a Virtual Private Network (VPN) to imitate the network environment of an organization's internal network. Following that, the attacker utilize the attack framework and methodologies included in Kali Linux to conduct five rounds of comprehensive APT attacks on the network environment, while Wireshark collected network traffic on all workstations. CICFlowMeter-V3 processes the gathered PCAP network packet files to create a network flow dataset that can be forwarded to machine learning. Following the generation SCVIC-APT-2021, UNSW-NB15, and NSL-KDD are used as benchmarks for the proposed PKI model and Progressive PKI model, respectively. Due to the imbalance problem in all three datasets, the macro average F1-score is chosen as the model's evaluation criterion. The experimental results show that the highest macro average F1-score of PKI model in SCVIC-APT-2021 increased by 11.01% and reach up to 81.92% when compared to the supervised method. The best weighted average F1-score on the NSL-KDD and UNSW-NB15 are 76.42% and 79.20% respectively.

For the second part, because host events are typically confidential and cannot be made public, there are few studies examining the use of a combination of network and host features to identify network attacks. The thesis uses the SCVIC-CIDS-2021 dataset that is generated in the Smart Connected Vehicles Innovation Centre at the University of Ottawa with the original data from the CSE-CIC-IDS2018 dataset, including the PCAP network packet file and Windows system event logs. The PCAP network packet files are processed in the same manner as the first part. However, because event logs contain descriptive language, when they are processed, host-based features are separated into numerical event features and event message features. These two components are transformed into the feature matrices by the transformer encoder. When used with the generated dataset, the AE and GRU reduce the dimensionality of network-based and host-based features by obtaining hidden dimensions, therefore removing the effect of irrelevant features and shortening training and testing time. After dimensionality reduction, the features are submitted to the PKI and PKID models for training and testing. The test results indicate that the PKID model performs better in terms of classification, with a macro average F1-score of 96.60%, which is 7.62% higher than only utilizing network-based features.

In summary, this thesis presents a ML-based strategy for detecting network attacks

with great performance. Among them, developing the APT attack dataset and detecting different stages of it provide a foundation for future APT attack research. Furthermore, the combination of network-based and host-based features for network attack detection opens up new avenues for boosting IDS detection efficiency and accuracy.

5.2 Future Directions

While the proposed model for detecting network attacks performs well on a variety of datasets, there is still potential for improvement. The following description indicates future directions:

- When confronted with the APT attack, the SCVIC-APT-2021 dataset exhibits imbalance problem. According to the data distribution, the proportion of regular traffic is over 95%. When such a dataset is fed into a knowledge-based model, it generates a larger bias, compromising the accuracy of the prior knowledge. As a result, resolving the imbalance issue is critical for increasing the accuracy of APT attack detection and the quality of dataset. The implementation paths are classified into three distinct categories. To begin, automated, standardised APT attack scripts need to be constructed to compensate for the quantitative limitations by increasing attack traffic. Multiple APT attack attack strategies could be added to the SCVIC-APT-2021 dataset in order to expand the knowledge of the APT attack, as the initiators of the APT attack could adapt attack techniques to the real-time environment of the target network domain. Second, methods such as Random Oversampling and SMOTE can be used to produce comparable data points from the original attack data points in order to compensate for the lack of quantity. Finally, undersampling techniques like as Random Undersampling, Clustering, and others can be used to select normal traffic in order to retain the order of magnitude of normal and attack data points.

- If a mix of network-based and host-based features is employed to identify network attacks, obtaining host-based features is challenging due to the varying formats and confidentiality of host events provided by different systems. To begin, this thesis extracts host features using the Windows operating system, but the majority of modern servers run on Linux. As a result, for host events that only provide text descriptions in the Linux system, the approach we provided to extract host-based features is inapplicable, and a strategy for extracting host-based features particular to the Linux system needs to be designed. Additionally, host events are considered internal or sensitive information by the organization and are not released publicly.The

GRU technique employed in this research to extract the hidden dimensions of host events cannot be used since the data associated with the original event is unavailable. However, if GRU Auto-encoder in the form of Fig. 5.1 is applied to extract hidden dimensions, the above problem is resolved. Host events can be encoded in the internal network and then delivered to the machine learning model for attack detection. Using GRU-AE can protect the host events in the internal network from being leaked and improve its security.

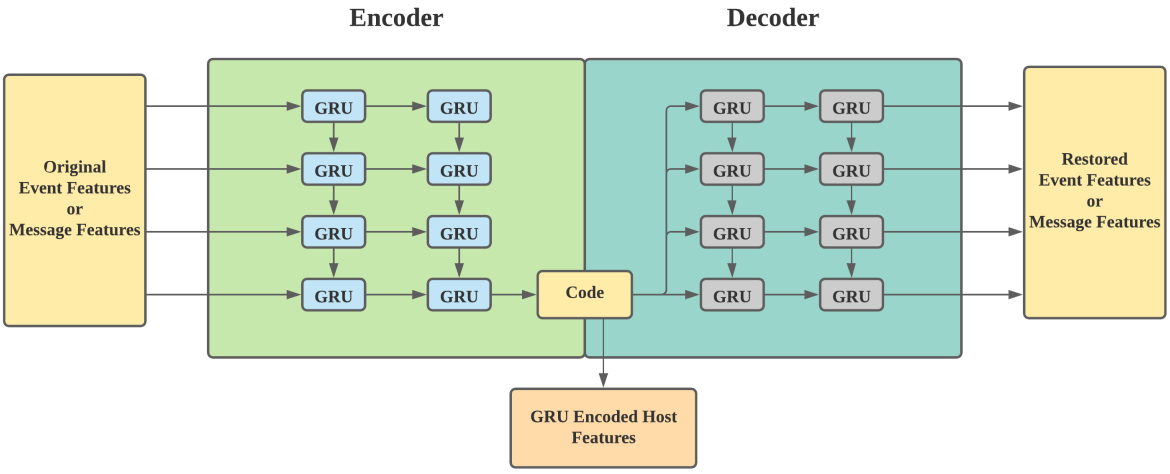


Figure 5.1: The structure of GRU Auto-encoder

- Along with advancements in the datasets, the knowledge-based model offers opportunity for growth as well. When performing network attack detection, the unsupervised method's pre-classification results are added as prior knowledge to the training and testing processes. However, there are other ways to acquire prior knowledge. For instance, the neural network layer in Fig. 1.2 can also be utilized to generate knowledge. As a result, selecting knowledge generating strategies for various models is another area of future effort and research.

References

[1] 2021 cyber attack trends mid-year report | check point software.

[2] 7.9 billion records exposed so far in 2019.

[3] Cybersecurity: What is it, and why is it an incredible career? (2021). Section: Industry Insider.

[4] Digital 2021: Global overview report.

[5] SCADA / ICS PCAP files from 4sics.

[6] ICS vulnerabilities: 2018 in review, May 2020. Library Catalog: www.ptsecurity.com.

[7] Overview of Cyber Vulnerabilities | CISA, May 2020.

[8] Jawad Ahmed, Hassan Habibi Gharakheili, Qasim Raza, Craig Russell, and Vijay Sivaraman. Monitoring enterprise dns queries for detecting data exfiltration from internal hosts. *IEEE Transactions on Network and Service Management*, 17(1):265–279, 2020.

[9] Kevser Ovaz Akpinar and Ibrahim Ozcelik. Analysis of machine learning methods in EtherCAT-based anomaly detection. 7:184365–184374. Conference Name: IEEE Access.

[10] Majjed Al-Qatf, Yu Lasheng, Mohammed Al-Habib, and Kamal Al-Sabahi. Deep learning approach combining sparse autoencoder with SVM for network intrusion detection. 6:52843–52856. Conference Name: IEEE Access.

[11] Wathiq Laftah Al-Yaseen, Zulaiha Ali Othman, and Mohd Zakree Ahmad Nazri. Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system. *Expert Systems with Applications*, 67:296–303, January 2017.

[12] A. Alshamrani, S. Myneni, A. Chowdhary, and D. Huang. A Survey on Advanced Persistent Threats: Techniques, Solutions, Challenges, and Research Opportunities. *IEEE Communications Surveys Tutorials*, 21(2):1851–1877, 2019. Conference Name: IEEE Communications Surveys Tutorials.

[13] Amar Amouri, Vishwa T Alaparthy, and Salvatore D Morgera. A machine learning based intrusion detection system for mobile internet of things. *Sensors*, 20(2):461, 2020.

[14] Giovanni Apruzzese, Fabio Pierazzi, Michele Colajanni, and Mirco Marchetti. Detection and threat prioritization of pivoting attacks in large networks. *IEEE Transactions on Emerging Topics in Computing*, 2017.

[15] Giovanni Apruzzese, Fabio Pierazzi, Michele Colajanni, and Mirco Marchetti. Detection and threat prioritization of pivoting attacks in large networks. *IEEE Transactions on Emerging Topics in Computing*, 8(2):404–415, 2020.

[16] Asmaa Shaker Ashoor and Sharad Gore. Importance of intrusion detection system (IDS). page 7.

[17] Bijoy Babu, Thafasal Ijyas, Muneer P., and Justin Varghese. Security issues in SCADA based industrial control systems. In *2017 2nd International Conference on Anti-Cyber Crimes (ICACC)*, pages 47–51, Abha, Saudi Arabia, March 2017. IEEE.

[18] T. Bai, H. Bian, A. A. Daya, M. A. Salahuddin, N. Limam, and R. Boutaba. A machine learning approach for rdp-based lateral movement detection. In *2019 IEEE 44th Conference on Local Computer Networks (LCN)*, pages 242–245, 2019.

[19] Tim Bai, Haibo Bian, Mohammad A. Salahuddin, Abbas Abou Daya, Noura Limam, and Raouf Boutaba. Rdp-based lateral movement detection using machine learning. *Computer Communications*, 165:9–19, 2021.

[20] Mustapha Belouch, Salah El Hadaj, and Mohamed Idhammad. Performance evaluation of intrusion detection based on machine learning using Apache Spark. *Procedia Computer Science*, 127:1–6, 2018. Publisher: Elsevier BV.

[21] Elisa Bertino, Ashish Kamra, and James P. Early. Profiling Database Application to Detect SQL Injection Attacks. In *2007 IEEE International Performance, Computing, and Communications Conference*, pages 449–458, April 2007. ISSN: 2374-9628.

[22] Leyla Bilge and Tudor Dumitras. Before we knew it: an empirical study of zero-day attacks in the real world. In *Proceedings of the 2012 ACM conference on Computer and communications security - CCS '12*, page 833, Raleigh, North Carolina, USA, 2012. ACM Press.

[23] Raymond C. Borges Hink, Justin M. Beaver, Mark A. Buckner, Tommy Morris, Uttam Adhikari, and Shengyi Pan. Machine learning for power system disturbance and cyber-attack discrimination. In *2014 7th International Symposium on Resilient Control Systems (ISRCS)*, pages 1–8.

[24] Joel Branch, Alan Bivens, Chi Yu Chan, Taek Kyeun Lee, and Boleslaw K Szymanski. Denial of service intrusion detection using time dependent deterministic finite automata. In *Proc. Graduate Research Conference*, pages 45–51, 2002.

[25] Guillaume Brogi and Elena Di Bernardino. Hidden markov models for advanced persistent threats. 14(4):181–190. Publisher: Inderscience Publishers.

[26] Jessey Bullock and Jeff T. Parker. *Wireshark for Security Professionals: Using Wireshark and the Metasploit Framework*. John Wiley & Sons. Google-Books-ID: DgEtDgAAQBAJ.

[27] Glenn Carl, George Kesidis, Richard R Brooks, and Suresh Rai. Denial-of-service attack-detection techniques. *IEEE Internet computing*, 10(1):82–89, 2006.

[28] Christian Cervantes, Diego Poplade, Michele Nogueira, and Aldri Santos. 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)*, pages 606–611. IEEE, 2015.

[29] Xiaoliang Chen, Baojia Li, Roberto Proietti, Zuqing Zhu, and S. J. Ben Yoo. Self-taught anomaly detection with hybrid unsupervised/supervised machine learning in optical networks. 37(7):1742–1749. Conference Name: Journal of Lightwave Technology.

[30] Bernard Lee Jin Chuan, Manmeet Mahinderjit Singh, and Azizul Rahman Mohd Shariff. APTGuard : Advanced persistent threat (APT) detections and predictions using android smartphone. In Rayner Alfred, Yuto Lim, Ag Asri Ag Ibrahim, and Patricia Anthony, editors, *Computational Science and Technology*, pages 545–555. Springer Singapore.

[31] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling.

[32] A. Creery and E.J. Byres. Industrial cybersecurity for power system and SCADA networks. In *Record of Conference Papers Industry Applications Society 52nd Annual Petroleum and Chemical Industry Conference*, pages 303–309, September 2005. ISSN: 2161-8127.

[33] Abbas Abou Daya, Mohammad A. Salahuddin, Noura Limam, and Raouf Boutaba. BotChase: Graph-based bot detection using machine learning. 17(1):15–29. Conference Name: IEEE Transactions on Network and Service Management.

[34] Gustavo De Carvalho Bertoli, Lourenço Alves Pereira Júnior, Osamu Saotome, Aldri L. Dos Santos, Filipe Alves Neto Verri, Cesar Augusto Cavalheiro Marcondes, Sidnei Barbieri, Moises S. Rodrigues, and José M. Parente De Oliveira. An end-to-end framework for machine learning-based network intrusion detection system. *IEEE Access*, 9:106790–106805, 2021.

[35] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding.

[36] L Dhanabal and Dr S P Shantharajah. A study on NSL-KDD dataset for intrusion detection system based on classification algorithms. 4(6):7.

[37] Stephanie Dick. Artificial intelligence. *Harvard Data Science Review*, 1(1), 7 2019. https://hdsr.mitpress.mit.edu/pub/0aytgrau.

[38] Cho Do Xuan, Mai Hoang Dao, and Hoa Dinh Nguyen. APT attack detection based on flow network analysis techniques using deep learning. 39(3):4785–4801. Publisher: IOS Press.

[39] Oliver Eigner, Philipp Kreimel, and Paul Tavolato. Detection of man-in-the-middle attacks on industrial control networks. In *2016 International Conference on Software Security and Assurance (ICSSA)*, pages 64–69. IEEE, 2016.

[40] Wisam Elmasry, Akhan Akbulut, and Abdul Halim Zaim. Evolving deep learning architectures for network intrusion detection using a double PSO metaheuristic. *Computer Networks*, 168:107042, February 2020.

[41] Pablo A. Estevez, Michel Tesmer, Claudio A. Perez, and Jacek M. Zurada. Normalized mutual information feature selection. *IEEE Transactions on Neural Networks*, 20(2):189–201, 2009.

[42] Michael Filimowicz and Veronika Tzankova. *Reimagining Communication: Action*. Routledge. Google-Books-ID: 291FEAAAQBAJ.

[43] Igor Nai Fovino, Andrea Carcano, Marcelo Masera, and Alberto Trombetta. An experimental investigation of malware attacks on scada systems. *International Journal of Critical Infrastructure Protection*, 2(4):139–145, 2009.

[44] David A. Freedman. *Statistical Models: Theory and Practice*. Cambridge University Press. Google-Books-ID: fW_9BV5Wpf8C.

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

[46] Robin Gassais, Naser Ezzati-Jivan, Jose M. Fernandez, Daniel Aloise, and Michel R. Dagenais. Multi-level host-based intrusion detection system for internet of things. 9(1):62.

[47] Thanassis Giannetsos, Tassos Dimitriou, Ioannis Krontiris, and Neeli R Prasad. Arbitrary code injection through self-propagating worms in von neumann architecture devices. *The Computer Journal*, 53(10):1576–1593, 2010.

[48] Vincenzo Giuliano and Valerio Formicola. ICSrange: A simulation-based cyber range platform for industrial control systems.

[49] Jie Gu, Lihong Wang, Huiwen Wang, and Shanshan Wang. A novel approach to intrusion detection using SVM ensemble with feature augmentation. *Computers & Security*, 86:53–62, September 2019.

[50] Bart Haagdorens, Tim Vermeiren, and Marnix Goossens. Improving the performance of signature-based network intrusion detection sensors by multi-threading. In *International Workshop on Information Security Applications*, pages 188–203. Springer, 2004.

[51] Arash Habibi Lashkari. Cicflowmeter-v4.0 (formerly known as iscxflowmeter) is a network traffic bi-flow generator and analyser for anomaly detection. https://github.com/iscx/cicflowmeter, 08 2018.

[52] Amir Haider, Muhammad Adnan Khan, Abdur Rehman, Muhib Ur Rahman, and Hyung Seok Kim. A real-time sequential deep extreme learning machine cybersecurity intrusion detection system. *CMC-COMPUTERS MATERIALS & CONTINUA*, 66(2):1785–1798, 2021.

[53] Peter Harrington. *Machine Learning in Action*. Simon and Schuster. Google-Books-ID: XTozEAAAQBAJ.

[54] Maaz Hasan. A Hybrid Real-Time Intrusion Detection System for an Internet of Things Environment with Signature and Anomaly Based Intrusion detection. Master's thesis, Dublin, National College of Ireland, December 2019.

[55] Mahmudul Hasan, Md. Milon Islam, Md Ishrak Islam Zarif, and M. M. A. Hashem. Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches. *Internet of Things*, 7:100059, September 2019.

[56] Hussin Hejase, Hasan Kazan, and Imad Moukadem. *ADVANCED PERSISTENT THREATS (APT): AN AWARENESS REVIEW*.

[57] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[58] Mohammed Hussein and Mohammad Zulkernine. Umlintr: a uml profile for specifying intrusions. In *13th Annual IEEE International Symposium and Workshop on Engineering of Computer-Based Systems (ECBS'06)*, pages 8–pp. IEEE, 2006.

[59] Bhupendra Ingre and Anamika Yadav. Performance analysis of nsl-kdd dataset using ann. In *2015 International Conference on Signal Processing and Communication Engineering Systems*, pages 92–96, 2015.

[60] Navjyotsinh Jadeja and Viral Parmar. Implementation and mitigation of various tools for pass the hash attack. 79:755–764.

[61] Nipun Jaswal. *Mastering Metasploit: Exploit systems, cover your tracks, and bypass security controls with the Metasploit 5.0 framework, 4th Edition*. Packt Publishing Ltd. Google-Books-ID: 5ozrDwAAQBAJ.

[62] Javad Hassannataj Joloudari, Mojtaba Haderbadi, Amir Mashmool, Mohammad GhasemiGol, Shahab S Band, and Amir Mosavi. Early detection of the advanced persistent threat attack using performance analysis of deep learning. *IEEE Access*, 8:186125–186137, 2020.

[63] Shijoe Jose, D. Malathi, Bharath Reddy, and Dorathi Jayaseeli. A survey on anomaly based host intrusion detection system. *Journal of Physics: Conference Series*, 1000:012049, apr 2018.

[64] Prabhakaran Kasinathan, Claudio Pastrone, Maurizio A Spirito, and Mark Vinkovits. 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)*, pages 600–607. IEEE, 2013.

[65] Sydney Mambwe Kasongo and Yanxia Sun. A deep learning method with filter based feature engineering for wireless intrusion detection system. 7:38597–38607. Conference Name: IEEE Access.

[66] Chayoung Kim and JiSu Park. Designing online network intrusion detection using deep auto-encoder Q-learning. *Computers & Electrical Engineering*, 79:106460, October 2019.

[67] Maria Kjaerland. A taxonomy and comparison of computer security incidents from the commercial and government sectors. *Computers & Security*, 25(7):522–538, 2006.

[68] Maria Kjaerland. A taxonomy and comparison of computer security incidents from the commercial and government sectors. *Computers & Security*, 25(7):522–538, October 2006.

[69] Gulshan Kumar, Kutub Thakur, and Maruthi Rohit Ayyagari. Mlesidss: machine learning-based ensembles for intrusion detection systems—a review. *The Journal of Supercomputing*, pages 1–34, 2020.

[70] Vikash Kumar, Ditipriya Sinha, Ayan Kumar Das, Subhash Chandra Pandey, and Radha Tamal Goswami. An integrated rule based intrusion detection system: analysis on UNSW-NB15 data set and the real time online dataset. 23(2):1397–1418.

[71] Thorsten Kurpjuhn. Demystifying the role of ai for better network security. *Network Security*, 2019(8):14–17, 2019.

[72] Arash Habibi Lashkari, Andi Fitriah A.Kadir, Hugo Gonzalez, Kenneth Fon Mbah, and Ali A. Ghorbani. Towards a network-based framework for android malware detection and characterization. In *2017 15th Annual Conference on Privacy, Security and Trust (PST)*, pages 233–23309, 2017.

[73] Arash Habibi Lashkari, Andi Fitriah A. Kadir, Laya Taheri, and Ali A. Ghorbani. Toward developing a systematic approach to generate benchmark android malware datasets and classification. In *2018 International Carnahan Conference on Security Technology (ICCST)*, pages 1–7, 2018.

[74] Joffrey L. Leevy and Taghi M. Khoshgoftaar. A survey and analysis of intrusion detection models based on CSE-CIC-IDS2018 big data. 7(1):104.

[75] Meicong Li, Wei Huang, Yongbin Wang, and Wenqing Fan. The optimized attribute attack graph based on apt attack stage model. In *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*, pages 2781–2785, 2016.

[76] Michael Liljenstam, David M Nicol, Vincent H Berk, and Robert S Gray. Simulating realistic network worm traffic for worm warning system design and testing. In *Proceedings of the 2003 ACM workshop on Rapid malcode*, pages 24–33, 2003.

[77] Huan Liu and R. Setiono. Chi2: feature selection and discretization of numeric attributes. In *Proceedings of 7th IEEE International Conference on Tools with Artificial Intelligence*, pages 388–391, 1995.

[78] Jinxin Liu, Burak Kantarci, and Carlisle Adams. Machine learning-driven intrusion detection for contiki-NG-based IoT networks exposed to NSL-KDD dataset. In *Proceedings of the 2nd ACM Workshop on Wireless Security and Machine Learning*, WiseML '20, pages 25–30. Association for Computing Machinery.

[79] Jinxin Liu, Yu Shen, Murat Simsek, Burak Kantarci, Hussein T. Mouftah, Mehran Bagheri, and Petar Djukic. A new realistic benchmark for advanced persistent threats in network traffic. *IEEE Networking Letters*, pages 1–1, 2022.

[80] Jiazhong Lu, Kai Chen, Zhongliu Zhuo, and XiaoSong Zhang. A temporal correlation and traffic analysis approach for APT attacks detection. 22(3):7347–7358.

[81] Roberto Magán-Carrión, Daniel Urda, Ignacio Díaz-Cano, and Bernabé Dorronsoro. Towards a reliable comparison and evaluation of network intrusion detection systems based on machine learning approaches. *Applied Sciences*, 10(5):1775, 2020.

[82] Ritesh K. Malaiya, Donghwoon Kwon, Sang C. Suh, Hyunjoo Kim, Ikkyun Kim, and Jinoh Kim. An empirical evaluation of deep learning for network anomaly detection. 7:140806–140817. Conference Name: IEEE Access.

[83] Claudio Mazzariello, Roberto Bifulco, and Roberto Canonico. Integrating a network ids into an open source cloud computing environment. In *2010 Sixth International Conference on Information Assurance and Security*, pages 265–270, 2010.

[84] Gaurav Meena and Ravi Raj Choudhary. A review paper on ids classification using kdd 99 and nsl kdd dataset in weka. In *2017 International Conference on Computer, Communications and Electronics (Comptelix)*, pages 553–558, 2017.

[85] Sadegh M. Milajerdi, Rigel Gjomemo, Birhanu Eshete, R. Sekar, and V.N. Venkatakrishnan. Holmes: Real-time apt detection through correlation of suspicious information flows. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 1137–1152, 2019.

[86] Douglas C Montgomery, Elizabeth A Peck, and G Geoffrey Vining. *Introduction to linear regression analysis*, volume 821. John Wiley & Sons, 2012.

[87] Stephen Muggleton. Alan turing and the development of artificial intelligence. 27(1):3–10. Publisher: IOS Press.

[88] Geoff Mulligan. The 6lowpan architecture. In *Proceedings of the 4th workshop on Embedded networked sensors*, pages 78–82, 2007.

[89] Weicong Na and Qi-jun Zhang. Unified automated knowledge-based neural network modeling for microwave devices. In *2015 IEEE MTT-S International Conference on Numerical Electromagnetic and Multiphysics Modeling and Optimization (NEMO)*, pages 1–3, 2015.

[90] K. Nar and S. S. Sastry. An analytical framework to address the data exfiltration of advanced persistent threats. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 867–873, 2018.

[91] Sheraz Naseer, Yasir Saleem, Shehzad Khalid, Muhammad Khawar Bashir, Jihun Han, Muhammad Munwar Iqbal, and Kijun Han. Enhanced network anomaly detection based on deep neural networks. 6:48231–48246. Conference Name: IEEE Access.

[92] Andrew Ng et al. Sparse autoencoder. *CS294A Lecture notes*, 72(2011):1–19, 2011.

[93] Tri Gia Nguyen, Trung V. Phan, Binh T. Nguyen, Chakchai So-In, Zubair Ahmed Baig, and Surasak Sanguanpong. SeArch: A collaborative and intelligent NIDS architecture for SDN-based cloud IoT networks. 7:107678–107694. Conference Name: IEEE Access.

[94] M. Nicho and C. D. McDermott. Dimensions of 'socio' vulnerabilities of advanced persistent threats. In *2019 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, pages 1–5, 2019.

[95] Doohwan Oh, Deokho Kim, and Won Woo Ro. A malicious pattern detection engine for embedded security systems in the internet of things. *Sensors*, 14(12):24188–24211, 2014.

[96] Chung-Ming Ou. Host-based intrusion detection systems inspired by machine learning of agent-based artificial immune systems. In *2019 IEEE International Symposium on INnovations in Intelligent SysTems and Applications (INISTA)*, pages 1–5, 2019.

[97] Mike O'Leary. FirewallFirewallIPFireIPFireIPFireFirewalls. In Mike O'Leary, editor, *Cyber Operations: Building, Defending, and Attacking Modern Computer Networks*, pages 857–896. Apress.

[98] Swati Paliwal. *Denial-of-Service, Probing & Remote to User (R2L) Attack Detection using Genetic Algorithm.*

[99] Spyridon Papastergiou, Haralambos Mouratidis, and Eleni-Maria Kalogeraki. Handling of advanced persistent threats and complex incidents in healthcare, transportation and energy ICT infrastructures.

[100] Animesh Patcha and Jung-Min Park. An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer networks*, 51(12):3448–3470, 2007.

[101] Bibudhendu Pati, Chhabi Rani Panigrahi, Rajkumar Buyya, and Kuan-Ching Li. *Advanced Computing and Intelligent Engineering: Proceedings of ICACIE 2018, Volume 1.* Springer Nature. Google-Books-ID: cHfRDwAAQBAJ.

[102] Supriyanto Praptodiyono, Moh. Jauhari, Rian Fahrizal, Iznan H. Hasbullah, Azlan Osman, and Shafiq Ul Rehman. Integration of firewall and ids on securing mobile ipv6. In *2020 2nd International Conference on Industrial Electrical and Electronics (ICIEE)*, pages 163–168, 2020.

[103] Santiago Quintero-Bonilla and Angel Martín del Rey. A new proposal on the advanced persistent threat: A survey. *Applied Sciences*, 10(11), 2020.

[104] Risto Rajala and Mika Westerlund. Business models–a new perspective on firms' assets and capabilities: observations from the finnish software industry. *The International Journal of Entrepreneurship and Innovation*, 8(2):115–125, 2007.

[105] Raihan Ur Rasool, Usman Ashraf, Khandakar Ahmed, Hua Wang, Wajid Rafique, and Zahid Anwar. Cyberpulse: A machine learning based link flooding attack mitigation system for software defined networks. 7:34885–34899. Conference Name: IEEE Access.

[106] Abdul Razzaq, Khalid Latif, H. Farooq Ahmad, Ali Hur, Zahid Anwar, and Peter Charles Bloodsworth. Semantic security against web application attacks. *Information Sciences*, 254:19–38, January 2014.

[107] José Roldán, Juan Boubeta-Puig, José Luis Martínez, and Guadalupe Ortiz. Integrating complex event processing and machine learning: An intelligent architecture for detecting IoT security attacks. *Expert Systems with Applications*, 149:113251, July 2020.

[108] Kshira Sagar Sahoo, Deepak Puthal, Mayank Tiwary, Joel J. P. C. Rodrigues, Bibhudatta Sahoo, and Ratnakar Dash. An early detection of low rate DDoS attack to SDN based data center networks using information distance metrics. 89:685–697.

[109] Alan Saied, Richard E. Overill, and Tomasz Radzik. Detection of known and unknown DDoS attacks using Artificial Neural Networks. *Neurocomputing*, 172:385–393, January 2016.

[110] Fadi Salo, Mohammadnoor Injadat, Ali Bou Nassif, Abdallah Shami, and Aleksander Essex. Data mining techniques in intrusion detection systems: A systematic literature review. *IEEE Access*, 6:56046–56058, 2018.

[111] Phurivit Sangkatsanee, Naruemon Wattanapongsakorn, and Chalermpol Charnsripinyo. Practical real-time intrusion detection using machine learning approaches. 34(18):2227–2235.

[112] Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, and Xiaowei Xu. DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN. *ACM Transactions on Database Systems*, 42(3):19:1–19:21, July 2017.

[113] Jayasree Sengupta, Sushmita Ruj, and Sipra Das Bit. A comprehensive survey on attacks, security issues and blockchain solutions for iot and iiot. *Journal of Network and Computer Applications*, 149:102481, 2020.

[114] Muhammad Shafiq, Zhihong Tian, Yanbin Sun, Xiaojiang Du, and Mohsen Guizani. Selection of effective machine learning algorithm and Bot-IoT attacks traffic identification for internet of things in smart city. *Future Generation Computer Systems*, 107:433–442, June 2020.

[115] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A Ghorbani. Toward generating a new intrusion detection dataset and intrusion traffic characterization. *ICISSp*, 1:108–116, 2018.

[116] Jivitesh Sharma, Charul Giri, Ole-Christoffer Granmo, and Morten Goodwin. Multi-layer intrusion detection system with ExtraTrees feature selection, extreme learning machine ensemble, and softmax aggregation. 2019(1):15.

[117] Srishti Sharma, Yogita Gigras, Rita Chhikara, and Anuradha Dhull. Analysis of NSL KDD dataset using classification algorithms for intrusion detection system. 13(2):142–147.

[118] Lin Shenwen, Li Yingbo, and Du Xiongjie. Study and research of apt detection technology based on big data processing architecture. In *2015 IEEE 5th International Conference on Electronics Information and Emergency Communication*, pages 313–316, 2015.

[119] Farzaneh Izak Shiri, Bharanidharan Shanmugam, and Norbik Bashah Idris. A parallel technique for improving the performance of signature-based network intrusion detection system. In *2011 IEEE 3rd International Conference on Communication Software and Networks*, pages 692–696. IEEE, 2011.

[120] Nathan Shone, Tran Nguyen Ngoc, Vu Dinh Phai, and Qi Shi. A Deep Learning Approach to Network Intrusion Detection. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2(1):41–50, February 2018. Conference Name: IEEE Transactions on Emerging Topics in Computational Intelligence.

[121] Jimmy Shun and Heidar A. Malki. Network intrusion detection system using neural networks. In *2008 Fourth International Conference on Natural Computation*, volume 5, pages 242–246, 2008.

[122] Chris B Simmons, Sajjan G Shiva, Harkeerat Bedi, and Dipankar Dasgupta. AVOIDIT: A Cyber Attack Taxonomy. page 11, 2014.

[123] Murat Simsek. Developing 3-step modeling strategy exploiting knowledge based techniques. In *2011 20th European Conference on Circuit Theory and Design (ECCTD)*, pages 616–619, 2011.

[124] Murat Simsek, Burak Kantarci, and Azzedine Boukerche. Knowledge-based machine learning boosting for adversarial task detection in mobile crowdsensing. In *2020 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–7. ISSN: 2642-7389.

[125] Murat Simsek, Burak Kantarci, and Azzedine Boukerche. Self organizing feature map-integrated knowledge-based deep network against fake crowdsensing tasks. In

GLOBECOM 2020 - 2020 IEEE Global Communications Conference, pages 1–6, 2020.

[126] Murat Simsek and N. Serap Sengor. An efficient inverse ANN modeling approach using prior knowledge input with difference method. In *2009 European Conference on Circuit Theory and Design*, pages 323–326.

[127] Murat Simsek and N. Serap Sengor. An efficient inverse ann modeling approach using prior knowledge input with difference method. In *2009 European Conference on Circuit Theory and Design*, pages 323–326, 2009.

[128] Pradeep Singh and M Venkatesan. Hybrid approach for intrusion detection system. In *2018 International Conference on Current Trends towards Converging Technologies (ICCTCT)*, pages 1–5. IEEE, 2018.

[129] Saurabh Singh, Pradip Kumar Sharma, Seo Yeon Moon, Daesung Moon, and Jong Hyuk Park. A comprehensive study on APT attacks and countermeasures for future networks and communications: challenges and solutions. 75(8):4543–4574.

[130] M Soria-Machado, D Abolins, C Boldea, and K Socha. Detecting lateral movements in windows infrastructure. page 22.

[131] Thomas F Stafford and Andrew Urbaczewski. Spyware: The ghost in the machine. *The Communications of the Association for Information Systems*, 14(1):49, 2004.

[132] Branka Stojanović, Katharina Hofer-Schmitz, and Ulrike Kleb. APT datasets and attack modeling for automated detection methods: A review. 92:101734.

[133] Sam Stover, Dave Dittrich, John Hernandez, and Sven Dietrich. Analysis of the storm and nugache trojans: P2p is here. *USENIX; login*, 32(6):18–27, 2007.

[134] Nasrin Sultana, Naveen Chilamkurti, Wei Peng, and Rabei Alhadad. Survey on SDN based network intrusion detection system using machine learning approaches. *Peer-to-Peer Networking and Applications*, 12(2):493–501, March 2019.

[135] Kazi Abu Taher, Billal Mohammed Yasin Jisan, and Md. Mahbubur Rahman. Network intrusion detection using supervised machine learning technique with feature selection. In *2019 International Conference on Robotics,Electrical and Signal Processing Techniques (ICREST)*, pages 643–646, 2019.

[136] Nima Taherifard, Murat Simsek, Charles Lascelles, and Burak Kantarci. Prior knowledge input to improve LSTM auto-encoder-based characterization of vehicular sensing data.

[137] Mahbod Tavallaee, Natalia Stakhanova, and Ali Akbar Ghorbani. Toward credible evaluation of anomaly-based intrusion-detection methods. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 40(5):516–524, 2010.

[138] Paul J. Taylor, Tooska Dargahi, and Ali Dehghantanha. Analysis of APT actors targeting IoT and big data systems: Shell_crew, NetTraveler, ProjectSauron, CopyKittens, volatile cedar and transparent tribe as a case study. In Ali Dehghantanha and Kim-Kwang Raymond Choo, editors, *Handbook of Big Data and IoT Security*, pages 257–272. Springer International Publishing.

[139] Zhihong Tian, Wei Shi, Yuhang Wang, Chunsheng Zhu, Xiaojiang Du, Shen Su, Yanbin Sun, and Nadra Guizani. Real-time lateral movement detection based on evidence reasoning network for edge computing environment. *IEEE Transactions on Industrial Informatics*, 15(7):4285–4294, 2019.

[140] Imran Ulghar, Hamid Jahankhani, and Stefan Kendzierskyj. *Blockchain Capabilities in Defending Advanced Persistent Threats Using Correlation Technique and Hidden Markov Models (HMM)*, pages 21–64. Springer International Publishing, Cham, 2021.

[141] Faheem Ullah, Matthew Edwards, Rajiv Ramdhany, Ruzanna Chitchyan, M. Ali Babar, and Awais Rashid. Data exfiltration: A review of external attack vectors and countermeasures. 101:18–54.

[142] Martin Ussath, David Jaeger, Feng Cheng, and Christoph Meinel. Advanced persistent threats: Behind the scenes. In *2016 Annual Conference on Information Science and Systems (CISS)*, pages 181–186, 2016.

[143] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need.

[144] Abhishek Verma and Virender Ranga. Machine learning based intrusion detection systems for IoT applications. 111(4):2287–2310.

[145] R. Vinayakumar, Mamoun Alazab, K. P. Soman, Prabaharan Poornachandran, and Sitalakshmi Venkatraman. Robust intelligent malware detection using deep learning. 7:46717–46738. Conference Name: IEEE Access.

[146] R Vinayakumar, K P Soman, and Prabaharan Poornachandran. Applying convolutional neural network for network intrusion detection. In *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 1222–1228, September 2017.

[147] David Wagner and R Dean. Intrusion detection via static analysis. In *Proceedings 2001 IEEE Symposium on Security and Privacy. S&P 2001*, pages 156–168. IEEE, 2000.

[148] Jin Wang, Liang-Chih Yu, K. Robert Lai, and Xuejie Zhang. Dimensional Sentiment Analysis Using a Regional CNN-LSTM Model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 225–230, Berlin, Germany, August 2016. Association for Computational Linguistics.

[149] P.M. Watson, K.C. Gupta, and R.L. Mahajan. Development of knowledge based artificial neural network models for microwave components. In *1998 IEEE MTT-S International Microwave Symposium Digest (Cat. No.98CH36192)*, volume 1, pages 9–12 vol.1. ISSN: 0149-645X.

[150] Christian Wressnegger, Guido Schwenk, Daniel Arp, and Konrad Rieck. A close look on n-grams in intrusion detection: anomaly detection vs. classification. In *Proceedings of the 2013 ACM workshop on Artificial intelligence and security*, pages 67–76, 2013.

[151] Handong Wu, Stephen Schwab, and Robert Lom Peckham. Signature based network intrusion detection system and method, September 9 2008. US Patent 7,424,744.

[152] Yang Xin, Lingshuang Kong, Zhi Liu, Yuling Chen, Yanmiao Li, Hongliang Zhu, Mingcheng Gao, Haixia Hou, and Chunhua Wang. Machine learning and deep learning methods for cybersecurity. 6:35365–35381. Conference Name: IEEE Access.

[153] Peter T. Yamak, Li Yujian, and Pius K. Gadosey. A comparison between ARIMA, LSTM, and GRU for time series forecasting. In *Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence*, ACAI 2019, pages 49–55. Association for Computing Machinery.

[154] Dingyu Yan, Feng Liu, and Kun Jia. Modeling an information-based advanced persistent threat attack on the internal network. In *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, pages 1–7, 2019.

[155] Chuanlong Yin, Yuefei Zhu, Jinlong Fei, and Xinzheng He. A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks. *IEEE Access*, 5:21954–21961, 2017.

[156] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*, 2014.

[157] Bruno Bogaz Zarpelão, Rodrigo Sanches Miani, Cláudio Toshio Kawakani, and Sean Carlisto de Alvarenga. A survey of intrusion detection in internet of things. *Journal of Network and Computer Applications*, 84:25–37, 2017.

[158] Chongzhen Zhang, Yanli Chen, Yang Meng, Fangming Ruan, Runze Chen, Yidan Li, and Yaru Yang. A novel framework design of network intrusion detection based on machine learning techniques. *Security and Communication Networks*, 2021, 2021.

[159] Chong Zhou and Randy C. Paffenroth. Anomaly Detection with Robust Deep Autoencoders. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, pages 665–674, Halifax, NS, Canada, August 2017. Association for Computing Machinery.

[160] Aaron Zimba, Hongsong Chen, and Zhaoshun Wang. Bayesian network based weighted apt attack paths modeling in cloud computing. *Future Generation Computer Systems*, 96:525–537, 2019.

[161] Aaron Zimba, Hongsong Chen, Zhaoshun Wang, and Mumbi Chishimba. Modeling and detection of the multi-stages of advanced persistent threats attacks based on semi-supervised learning and complex networks characteristics. 106:501–517.

[162] Wei Zong, Yang-Wai Chow, and Willy Susilo. Interactive three-dimensional visualization of network intrusion detection data for machine learning. *Future Generation Computer Systems*, 102:292–306, January 2020.

[163] Ünal Çavuşoğlu. A new hybrid approach for intrusion detection using machine learning methods. 49(7):2735–2761.

Appendix A

Attack Trace and Ground Truth for The SCVIC-APT-2021 Dataset

This appendix details the attack trace and ground truth used to create the SCVIC-APT-2021 dataset. There are four complete rounds of the APT attacks in the training set to produce malicious flows. The testing set is solely used to evaluate the model's detection ability against APT attacks so it requires a small amount of data. As a result, the attack data points in the testing set are generated using only one complete round of the APT attack.

The attack name, attack technique, IP address and Kali Port, as well as time, are utilized to describe the APT attack trace and ground truth. These recorded attack traces are critical information for labelling datasets and dataset tracing work. At the same time, please refer to the network domain structure shown in Fig. 3.5 to understand the attack trace demonstrated in this appendix more vividly.

The attack trace and ground truth are introduced in the following format:

Attack Name: The attack name or stage name of the APT.

Technique: The attack techniques used in this attack.

Attacker IP Address: The IP address of the attack initiator.

Victim IP Address: The IP address of the compromised device.

Kali Port: The Kali Port used by the attacker to compromise the victim.

Initial Time: The initial timestamp of the attack.

A.1 Testing Set

1. Initial Compromise

Technique: vsftpd, meterpreter

Attacker IP Address: 192.168.2.37

Victim IP Address: 192.168.2.49

Kali Port: 4455

Initial Time: 19:09:21

2. Reconnaissance

Technique: meterpreter

Attacker IP Address: 192.168.2.37

Victim IP Address: 192.168.2.49

Kali Port: 4455

Initial Time: 19:14:05

3. Pivoting

Technique: autoroute

Attacker Network Domain: 192.168.2.0/24

Victim Network Domain: 172.28.129.0/24

Initial Time: 19:19:31

4. Lateral Movement

Technique: psexec

Attacker IP Address: 172.28.129.14

Victim IP Address: 172.28.129.11

Kali Port: 4466

Initial Time: 19:22:27

5. Reconnaissance

Technique: meterpreter

Attacker IP Address: 172.28.129.14

Victim IP Address: 172.28.129.11

Kali Port: 4466

Initial Time: 19:26:39

6. Lateral Movement

Technique: psexec

Attacker IP Address: 172.28.129.14

Victim IP Address: 172.28.129.12

Kali Port: 4477

Initial Time: 19:27:31

7. Reconnaissance

Technique: meterpreter

Attacker IP Address: 172.28.129.14

Victim IP Address: 172.28.129.12

Kali Port: 4477

Initial Time: 19:30:22

8. Lateral Movement

Technique: psexec

Attacker IP Address: 172.28.129.14

Victim IP Address: 172.28.129.10

Kali Port: 4488

Initial Time: 19:39:37

9. Reconnaissance

Technique: meterpreter

Attacker IP Address: 172.28.129.14

Victim IP Address: 172.28.129.10

Kali Port: 4488

Initial Time: 19:41:56

10. Data Exfiltration

Technique: dns2tcp

Attacker IP Address: 172.28.129.14

Victim IP Address: 172.28.129.10

Kali Port: 4488

Initial Time: 19:45:33

11. Pivoting

Technique: autoroute

Attacker Network Domain: 172.28.129.0/24

Victim Network Domain: 10.147.20.0/24

Initial Time: 20:13:02

12. Lateral Movement

Technique: psexec

Attacker IP Address: 10.147.20.218

Victim IP Address: 10.147.20.237

Kali Port: 4499

Initial Time: 20:14:34

13. Reconnaissance

Technique: meterpreter

Attacker IP Address: 10.147.20.218

Victim IP Address: 10.147.20.237

Kali Port: 4499

Initial Time: 20:15:04

14. Data Exfiltration

Technique: C2 channel

Attacker IP Address: 10.147.20.218

Victim IP Address: 10.147.20.237

Kali Port: 4499

Initial Time: 20:26:22

A.2 Training Set Round 1

1. Initial Compromise

Technique: vsftpd, meterpreter

Attacker IP Address: 192.168.2.37

Victim IP Address:: 192.168.2.49

Kali Port: 4455

Initial Time: 19:30:15

2. Reconnaissance

Technique: meterpreter

Attacker IP Address: 192.168.2.37

Victim IP Address: 192.168.2.49

Kali Port: 4455

Initial Time: 19:34:21

3. Pivoting

Technique: autoroute

Attacker Network Domain: 192.168.2.0/24

Victim Network Domain: 172.28.129.0/24

Initial Time: 19:39:09

4. Lateral Movement

Technique: psexec

Attacker IP Address: 172.28.129.14

Victim IP Address: 172.28.129.11

Kali Port: 4466

Initial Time: 19:41:13

5. Reconnaissance
Technique: meterpreter
Attacker IP Address: 172.28.129.14
Victim IP Address: 172.28.129.11
Kali Port: 4466
Initial Time: 19:43:03

6. Lateral Movement
Technique: psexec
Attacker IP Address: 172.28.129.14
Victim IP Address: 172.28.129.12
Kali Port: 4477
Initial Time: 19:47:13

7. Reconnaissance
Technique: meterpreter
Attacker IP Address: 172.28.129.14
Victim IP Address: 172.28.129.12
Kali Port: 4477
Initial Time: 19:49:14

8. Lateral Movement
Technique: psexec
Attacker IP Address: 172.28.129.14
Victim IP Address: 172.28.129.10
Kali Port: 4488
Initial Time: 19:55:13

9. Reconnaissance
Technique: meterpreter
Attacker IP Address: 172.28.129.14
Victim IP Address: 172.28.129.10
Kali Port: 4488
Initial Time: 19:57:07

10. Data Exfiltration
Technique: C2 channel
Attacker IP Address: 172.28.129.14
Victim IP Address: 172.28.129.10
Kali Port: 4488
Initial Time: 19:59:17

11. Pivoting
Technique: autoroute
Attacker Network Domain: 172.28.129.0/24
Victim Network Domain: 10.147.20.0/24
Initial Time: 20:04:01

12. Lateral Movement
Technique: psexec
Attacker IP Address: 10.147.20.218
Victim IP Address: 10.147.20.237
Kali Port: 4499
Initial Time: 20:05:14

13. Reconnaissance
Technique: meterpreter
Attacker IP Address: 10.147.20.218
Victim IP Address: 10.147.20.237
Kali Port: 4499
Initial Time: 20:07:29

14. Data Exfiltration
Technique: C2 channel
Attacker IP Address: 10.147.20.218
Victim IP Address: 10.147.20.237
Kali Port: 4499
Initial Time: 20:09:43

A.3 Training Set Round 2

1. Initial Compromise

Technique: vsftpd, meterpreter

Attacker IP Address: 192.168.2.37

Victim IP Address: 192.168.2.49

Kali Port: 4455

Initial Time: 19:12:56

2. Reconnaissance

Technique: meterpreter

Attacker IP Address: 192.168.2.37

Victim IP Address: 192.168.2.49

Kali Port: 4455

Initial Time: 19:16:52

3. Pivoting

Technique: autoroute

Attacker Netowkr Domain: 192.168.2.0/24

Victim Network Domain: 172.28.129.0/24

Initial Time: 19:20:49

4. Lateral Movement

Technique: psexec

Attacker IP Address: 172.28.129.14

Victim IP Address: 172.28.129.10

Kali Port: 4466

Initial Time: 19:25:36

5. Reconnaissance

Technique: meterpreter

Attacker IP Address: 172.28.129.14

Victim IP Address: 172.28.129.10

Kali Port: 4466

Initial Time: 19:27:05

6. Data Exfiltration

Technique: C2 channel

Attacker IP Address: 172.28.129.14

Victim IP Address: 172.28.129.10

Kali Port: 4466

Initial Time: 19:30:19

7. Pivoting

Technique: autoroute

Attacker Network Domain: 172.28.129.0/24

Victim Network Domain: 10.147.20.0/24

Initial Time: 19:32:53

8. Lateral Movement

Technique: psexec

Attacker IP Address: 10.147.20.218

Victim IP Address: 10.147.20.237

Kali Port: 4477

Initial Time: 19:33:52

9. Reconnaissance

Technique: meterpreter

Attacker IP Address: 10.147.20.218

Victim IP Address: 10.147.20.237

Kali Port: 4477

Initial Time: 19:35:24

10. Data Exfiltration

Technique: C2 channel

Attacker IP Address: 10.147.20.218
Victim IP Address: 10.147.20.237
Kali Port: 4477
Initial Time: 19:38:53

11. Pivoting
Technique: autoroute
Attacker Network Domain: 10.147.20.0/24
Victim Network Domain: 172.28.128.0/24
Initial Time: 19:44:53

12. Lateral Movement
Technique: psexec
Attacker IP Address: 172.28.128.10
Victim IP Address: 172.28.128.11
Kali Port: 4488
Initial Time: 19:53:36

13. Lateral Movement
Technique: psexec
Attacker IP Address: 10.147.20.218
Victim IP Address: 10.147.20.237
Kali Port: 4499
Initial Time: 19:54:26

14. Pivoting
Technique: autoroute
Attacker Network Domain: 10.147.20.0/24
Victim Network Domain: 172.28.128.0/24
Initial Time: 19:55:48

15. Lateral Movement
Technique: psexec
Attacker IP Address: 172.28.128.10
Victim IP Address: 172.28.128.11
Kali Port: 5500
Initial Time: 19:59:39

16. Lateral Movement
Technique: psexec
Attacker IP Address: 172.28.129.14
Victim IP Address: 172.28.129.11
Kali Port: 5511
Initial Time: 20:03:41

17. Reconnaissance
Technique: meterpreter
Attacker IP Address: 172.28.129.14
Victim IP Address: 172.28.129.11
Kali Port: 5511
Initial Time: 20:04:48

18. Lateral Movement
Technique: psexec
Attacker IP Address: 172.28.129.14
Victim IP Address: 172.28.129.12
Kali Port: 5522
Initial Time: 20:10:48

19. Reconnaissance
Technique: meterpreter
Attacker IP Address: 172.28.129.14
Victim IP Address: 172.28.129.12
Kali Port: 5522
Initial Time: 20:11:50

A.4 Training Set Round 3

1. Initial Compromise

Technique: vsftpd, meterpreter

Attacker IP Address: 192.168.2.37

Victim IP Address: 192.168.2.49

Kali Port: 4455

Initial Time: 21:04:14

2. Reconnaissance

Technique: meterpreter

Attacker IP Address: 192.168.2.37

Victim IP Address: 192.168.2.49

Kali Port: 4455

Initial Time: 21:09:58

3. Pivoting

Technique: autoroute

Attacker Network Domain: 192.168.2.0/24

Victim Network Domain: 172.28.129.0/24

Initial Time: 21:15:29

4. Lateral Movement

Technique: psexec

Attacker IP Address: 172.28.129.14

Victim IP Address: 172.28.129.10

Kali Port: 4466

Initial Time: 21:21:50

5. Reconnaissance

Technique: meterpreter

Attacker IP Address: 172.28.129.14

Victim IP Address: 172.28.129.10

Kali Port: 4466

Initial Time: 21:22:18

6. Data Exfiltration

Technique: DNS Tunnelling

Attacker IP Address: 172.28.129.14

Victim IP Address: 172.28.129.10

Kali Port: 4466

Initial Time: 21:27:24

7. Pivoting

Technique: autoroute

Attacker Network Domain: 172.28.129.0/24

Victim Network Domain: 10.147.20.0/24

Initial Time: 21:41:55

8. Lateral Movement

Technique: psexec

Attacker IP Address: 10.147.20.218

Victim IP Address: 10.147.20.237

Kali Port: 4477

Initial Time: 21:43:28

9. Reconnaissance

Technique: meterpreter

Attacker IP Address: 10.147.20.218

Victim IP Address: 10.147.20.237

Kali Port: 4477

Initial Time: 21:45:52

10. Data Exfiltration

Technique: C2 Tunnelling

Attacker IP Address: 10.147.20.218
Victim IP Address: 10.147.20.237
Kali Port: 4477
Initial Time: 21:49:08

11. Pivoting
Technique: autoroute
Attacker Network Domain: 10.147.20.0/24
Victim Network Domain: 172.28.128.0/24
Initial Time: 21:57:45

12. Lateral Movement
Technique: psexec
Attacker IP Address: 172.28.128.10
Victim IP Address: 172.28.128.11
Kali Port: 4488
Initial Time: 22:02:25

13. Reconnaissance
Technique: meterpreter
Attacker IP Address: 172.28.128.10
Victim IP Address: 172.28.128.11
Kali Port: 4488
Initial Time: 22:03:47

14. Data Exfiltration
Technique: C2 Tunnelling
Attacker IP Address: 172.28.128.10
Victim IP Address: 172.28.128.11
Kali Port: 4488
Initial Time: 22:05:43

15. Lateral Movement
Technique: psexec
Attacker IP Address: 172.28.129.14
Victim IP Address: 172.28.129.11
Kali Port: 4499
Initial Time: 22:19:03

16. Reconnaissance
Technique: meterpreter
Attacker IP Address: 172.28.129.14
Victim IP Address: 172.28.129.11
Kali Port: 4499
Initial Time: 22:21:05

17. Lateral Movement
Technique: psexec
Attacker IP Address: 172.28.129.14
Victim IP Address: 172.28.129.11
Kali Port: 4499
Initial Time: 22:23:29

18. Reconnaissance
Technique: meterpreter
Attacker IP Address: 172.28.129.14
Victim IP Address: 172.28.129.11
Kali Port: 4499
Initial Time: 22:24:56

A.5 Training Set Round 4

1. Initial Compromise

Technique: vsftpd, meterpreter

Attacker IP Address: 192.168.2.37

Victim IP Address: 192.168.2.49

Kali Port: 4455

Initial Time: 19:11:41

2. Reconnaissance

Technique: meterpreter

Attacker IP Address: 192.168.2.37

Victim IP Address: 192.168.2.49

Kali Port: 4455

Initial Time: 19:15:55

3. Pivoting

Technique: autoroute

Attacker Network Domain: 192.168.2.0/24

Victim Network Domain: 172.28.129.0/24

Initial Time: 19:21:36

4. Lateral Movement

Technique: psexec

Attacker IP Address: 172.28.129.14

Victim IP Address: 172.28.129.10

Kali Port: 4466

Initial Time: 19:24:44

5. Reconnaissance

Technique: meterpreter

Attacker IP Address: 172.28.129.14

Victim IP Address: 172.28.129.10

Kali Port: 4466

Initial Time: 19:26:18

6. Data Exfiltration

Technique: C2 Tunnelling

Attacker IP Address: 172.28.129.14

Victim IP Address: 172.28.129.10

Kali Port: 4466

Initial Time: 19:28:47

7. Pivoting

Technique: autoroute

Attacker Network Domain: 172.28.129.0/24

Victim Network Domain: 10.147.20.0/24

Initial Time: 19:33:14

8. Lateral Movement

Technique: psexec

Attacker IP Address: 10.147.20.218

Victim IP Address: 10.147.20.237

Kali Port: 4477

Initial Time: 19:35:21

9. Reconnaissance

Technique: meterpreter

Attacker IP Address: 10.147.20.218

Victim IP Address: 10.147.20.237

Kali Port: 4477

Initial Time: 19:37:03

10. Data Exfiltration

Technique: DNS Tunnelling

Attacker IP Address: 10.147.20.218
Victim IP Address: 10.147.20.237
Kali Port: 4477
Initial Time: 19:39:35

11. Pivoting
Technique: autoroute
Attacker Network Domain: 10.147.20.0/24
Victim Network Domain: 172.28.128.0/24
Initial Time: 19:40:52

12. Lateral Movement
Technique: psexec
Attacker IP Address: 172.28.128.10
Victim IP Address: 172.28.128.11
Kali Port: 4488
Initial Time: 19:41:35

13. Lateral Movement
Technique: psexec
Attacker IP Address: 172.28.129.14
Victim IP Address: 172.28.129.11
Kali Port: 4499
Initial Time: 19:44:09

14. Reconnaissance
Technique: meterpreter
Attacker IP Address: 172.28.129.14
Victim IP Address: 172.28.129.11
Kali Port: 4499
Initial Time: 19:46:09

15. Lateral Movement
Technique: psexec
Attacker IP Address: 172.28.129.14
Victim IP Address: 172.28.129.12
Kali Port: 5500
Initial Time: 19:47:22

16. Reconnaissance
Technique: meterpreter
Attacker IP Address: 172.28.129.14
Victim IP Address: 172.28.129.12
Kali Port: 5500
Initial Time: 19:49:24

Appendix B

WSS and Silhouette Score Cure for SCVIC-CIDS-2021 Dataset

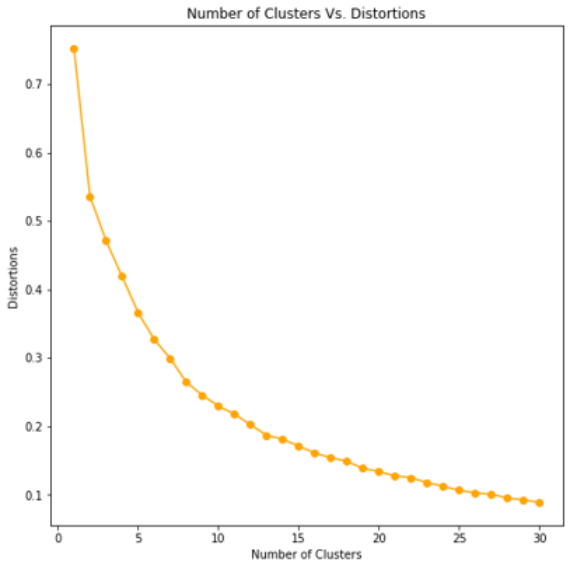


Figure B.1: WSS Score for Flow Features

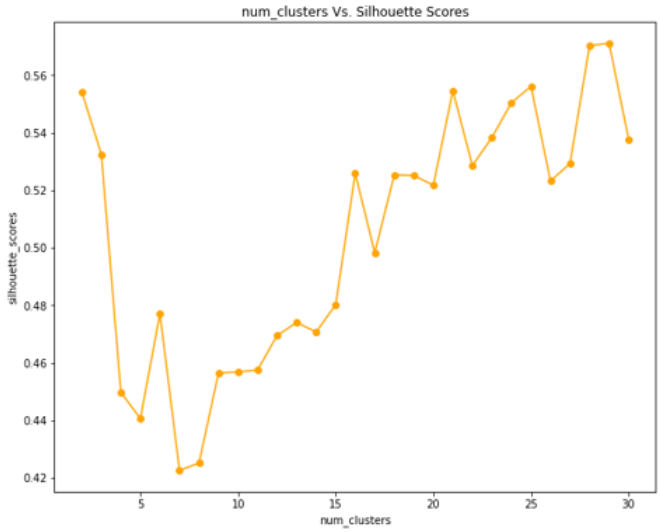


Figure B.2: Silhouette Score for Flow Features

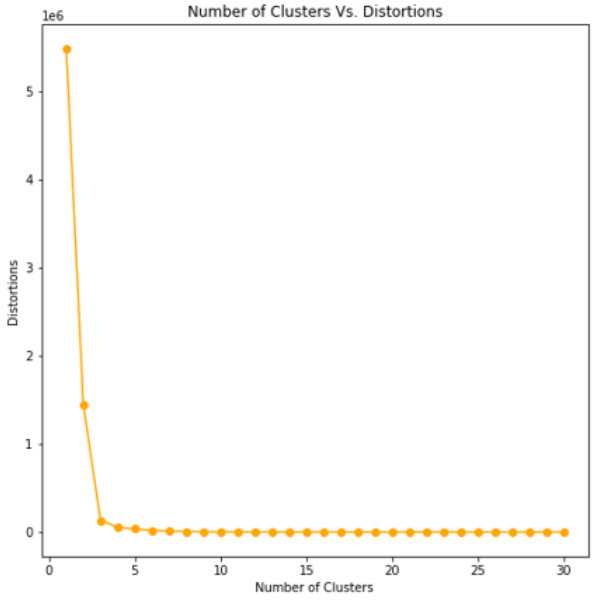


Figure B.3: WSS Score for Host Event Features

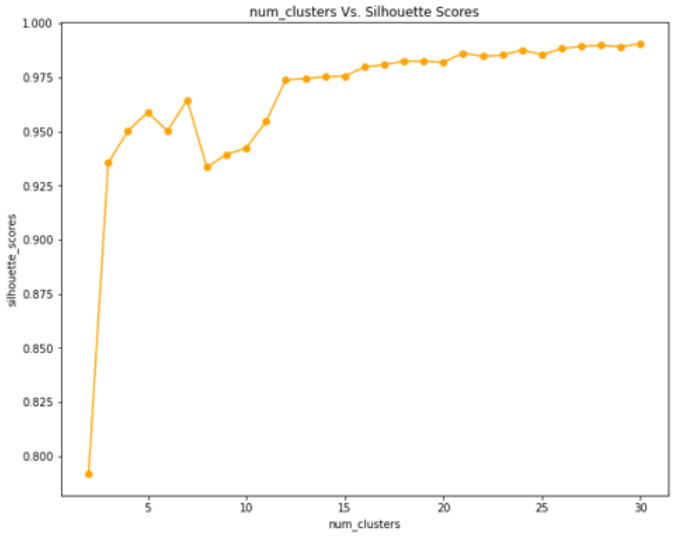


Figure B.4: Silhouette Score for Host Event Features

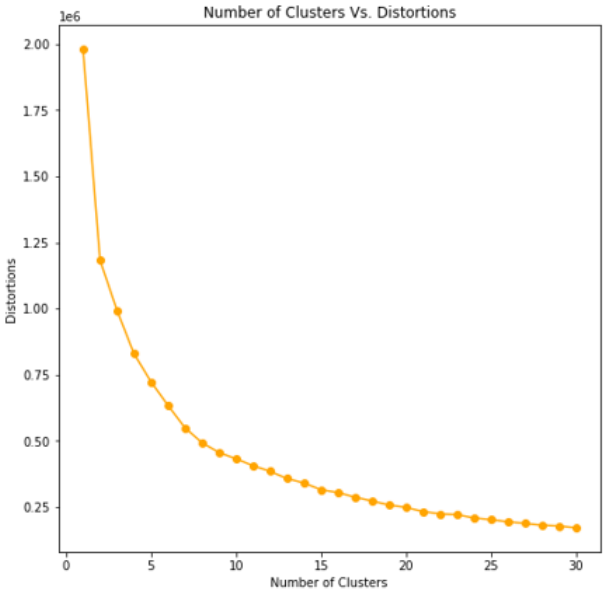


Figure B.5: WSS Score for Host Message Features

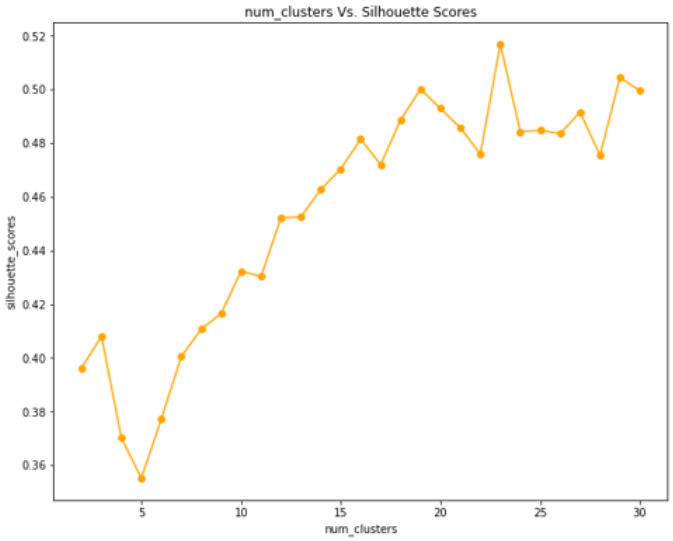


Figure B.6: Silhouette Score for Host Message Features