Data-driven framework and experimental validation for security monitoring of networked systems

Yussuf Ahmed

A thesis submitted in partial fulfilment of the requirement for the degree of Doctor of Philosophy

Faculty of Computing, Engineering and the Built Environment

Birmingham City University

UK

MAY 2021

Abstract

Cyber attacks have become more prevalent in the last few years, and several attacks have made headlines worldwide. It has become a lucrative business for cybercriminals who are motivated by financial gains. Other motives include political, social and espionage. Organisations are spending a vast amount of money from their IT budget to secure their critical assets from such attacks, but attackers still find ways to compromise these assets. According to a recent data breach report from IBM, the cost of a data breach is estimated to be around $4.24 million, and on average, it takes 287 days to detect and contain such breaches. Cyber attacks are continuing to increase, and no organisation is immune to such attacks, as demonstrated recently by the cyber attack on FireEye, a leading global cybersecurity firm.

This thesis aims to develop a data-driven framework for the security monitoring of networked systems. In this framework, models for detecting cyber attack stages, predicting cyber attacks using time series forecasting and the IoC model were developed to detect attacks that the security monitoring tools may have missed. In the cyberattack stage detection, the Cyber Kill Chain was leveraged and then mapped the detection modules to the various stages of the APT lifecycle. In the cyber prediction model, time series based fea-

ture forecasting was utilised to predict attacks to help system administrators take preventative measures. The Indicator of Compromise (IoC) model used host-based features to help detect IoCs more accurately. The main framework utilises network, host and IoC features. In these three models, the prediction accuracy of 91.1% and 98.8% was achieved for the APT and IoC models, while the time series forecasting model produced a reasonable low mean absolute error (MAE) and root mean square error (RMSE) score. The author also contributed to another paper on effective feature selection methods using deep feature abstraction in the form of unsupervised auto-encoders to extract more features. Wrapper-based feature selection techniques were then utilised using Support Vector Machine (SVM), Naive Bayes and Decision tree to select the highest-ranking features. Artificial Neural Networks (ANN) classifier was then used to distinguish impersonation from normal traffic. The contribution of the author to this paper was on the feature selection methods. This model achieved an overall accuracy of 99.5%. It is anticipated that these models will allow decision-makers and systems administrators to take proactive approaches to secure their systems and reduce data breaches.

Declaration

I declare that no material contained in this thesis has been used in any other submission for an academic award. I further declare that the work presented in this thesis entitled "Data-driven framework and experimental validation for security monitoring of networked systems" is my own, and all sources used have been appropriately acknowledged.

Acknowledgements

I want to thank the doctoral supervision team for their support during the PhD journey. Thanks to Dr Taufiq Asyhari and Prof Mark Josephs for their support. I would also like to thank Dr Syed Naqvi for his support during the early stages of the PhD. Thanks to the doctoral research college team and colleagues in the network and cybersecurity department.

Contents

List of Figures

List of Tables

Acronyms

ANN Artificial Neural Networks.

APT Advanced Persistent Threats.

ARIMA Autoregressive Integrated Moving Average.

BNN Bayesian Belief Networks.

CIS The Centre for internet security.

CKC Cyber Kill Chain.

CVSS The Common Vulnerability Scoring.

DDoS Distributed Denial of Service.

DLP Data Loss Prevention.

GQM Goal-Question-Metrics.

HIDS Host Intrusion Detection Systems.

IDS Intrusion Detection Systems.

IoA Indicators of Attack.

IoC Indicators of Compromise.

IPS Intrusion Prevention Systems.

ISMS Information Security Management System.

KNN K Nearest Neighbour.

LinearReg Linear Regression.

LNRG Linear Regression.

MFA Multiple Factor Analysis.

MTTP Mean Time To Patch.

NIDS Network Intrusion Detection Systems.

NIST National Institute of Standards and Technology.

NMAP Network Mapper.

NVD National Vulnerability Database.

OS Operating System.

PCA Principle Component Analysis.

PoP Pyramid of Pain.

RAT Remote Access Trojan.

RNN Recurrent Neural Networks.

SARIMA Seasonal Autoregressive Moving Average.

SIEM Security information and event management.

SMB Server Message Block.

SSH Secure Shell.

SSL Secure Socket Layer.

STIX Structured Threat Information eXpressions.

SVM Support Vector Machine.

TTPs Tactics, Techniques and Procedures.

VPN Virtual Private Network.

Chapter 1

Introduction

Cyber attacks are on the rise, and there is a substantial increase in the number of reported security breaches. The proliferation of internet-connected devices has substantially increased the attack surface resulting in cybercriminals exploiting vulnerabilities on these devices. Organisations are spending a substantial amount of their IT budgets on security tools to protect the confidentiality, integrity and availability of their systems and the data that reside on them. Despite these investments, cybercriminals are still finding ways to compromise these systems, demonstrating that traditional security measures such as firewalls are not enough on their own. The recent Covid-19 pandemic has also exacerbated the situation and resulted in numerous security breaches as highlighted in [1].

The presence of vulnerabilities on networked systems poses a risk that cyber attackers could exploit. Although all these vulnerabilities are difficult to eliminate due to the ever-changing threat landscape, it is important for organisations to have vulnerability scanning tools that can give them a

snapshot of their security status and the vulnerabilities that exist on these systems. Several tools are available, both commercial and open-source; these include Nessus, OpenVAS and NMAP. Other security tools that provide some assurance are antivirus and security monitoring tools such as Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) [2]. However, these security measures do not provide an overall risk or security status of the systems. Determining how secure these systems are, is still an open question and challenge.

The challenge faced by many organisations is determining the security of their networked systems and whether the current security investments is enough. Such questions are always difficult to address, and one approach suggested is the use of security metrics, although currently, there is no consensus on which metrics to use. Metrics support decision-making and can provide some degree of assurance. The questions that are often asked by senior management are: (i) How secure are we? (ii) What is our return on security investments?

Although these questions have always been difficult to answer quantitatively, security metrics could answer some of these questions. Most of the existing frameworks are subjective and answer these questions qualitatively using low, medium, and high scores. Although these are good starting points, the qualitative scores do not provide adequate measures compared to quantitative approaches.

In this thesis, a data-driven framework was developed to answer some of these questions. The model was evaluated using performance metrics to measure its effectiveness in terms of prediction accuracy. The resulting

framework will help decision-makers to make informed decisions by taking proactive approaches to prevent attacks. This framework's core components include Cyberattack stage detection, Indicators of Compromise, Cyber events forecasting, and prediction.

1.1 Motivations

Cybersecurity is at the forefront of many organisations due to the increasing complexity and volume of attacks. Organisations are spending a substantial amount of their IT budget on security tools to protect their critical systems. However, Cyber attacks are increasing, resulting in numerous security breaches being reported over the last few years. According to a recent cybersecurity breaches survey conducted in the UK [3], the number of cyber breaches is increasing, and the threat is evolving. The same report found that 46% of UK businesses experienced cyber attacks in the last 12 months.

According to a recent report by IBM, these security breaches can be very costly, and the cost of a data breach was estimated to be $4.24 million [4]. Criminals are finding ways around security defences and compromising previously secure systems. There are various sophisticated tools and attack vectors available to hackers. The changing threat landscape and complexity of the attacks make it difficult for IT professionals and the tools they rely on to detect all attacks. Some of these tools, such as intrusion detection systems, are known to generate false positives and negatives that can be tedious to triage and could result in serious security breaches if actual attacks are missed or ignored due to these false alerts. Hopefully, this thesis will support security

professionals and decision-makers to take proactive approaches to secure their systems by anticipating cyber attacks before cybercriminals exploit these vulnerabilities.

1.2 Research Questions

This thesis aims to advance current research on enhancing security monitoring of networked systems and address emerging threats and challenges affecting these systems, which led us to formulate our research questions. It is anticipated that our work will contribute to the accurate detection of cyber attacks. Detecting cyber attacks accurately and determining these systems' security status is still a significant challenge. There is a need to improve the detection accuracy of attacks and with minimal false alerts. This motivates the first question, which is answered by the proposed framework. The framework consists of models for detecting attacks, feature forecasting and IoC artefacts.

Research question 1: How could data-driven techniques be utilized to quantify the security of networked systems?

Sophisticated cyber attacks such as APTs are well planned and, on average, takes few months to discover. Modern cyber attacks are very complex, and detecting these attacks accurately, requires systems capable of detecting such attacks with minimal false positives and false negatives. This motivates our second research question.

Research question 2: How could data processing techniques be applied to improve the detection of cyber-attacks based on APT lifecycle?

Preventing cyber attacks is a crucial priority for every organisation, but most of the current solutions are reactive, such as sending alerts when a breach has occurred. Cyber attack prediction can prevent costly attacks by anticipating attacks in advance. This motivates our third research question.

Research question 3: To what extent can cyber-events be predicted based on feature engineering?

Cyber attacks are sometimes missed by the monitoring systems and can remain undetected for a long time. Detecting these IoCs accurately could result in minimising the impact of such attacks. This motivates our fourth research question.

Research question 4: How could security logs and IoC be leveraged to design an efficient cyber-attack detection in terms of accuracy and false alarm rate?

1.3 Aims and Objectives

This thesis's main objectives are to answer the research questions, but the activities also encompassed the following tasks.

1. Research on the security monitoring of networked systems
    - Identification of security enforcement and security dependent points/objects
    - Sourcing techniques and their optimisation to collect useful information for the metrics
    - Unit testing of the various individual features.
2. Research on cyber incident models

- Efficient detection metrics for cyber incidents with competitive accuracy and minimum false positives/negatives
- Investigation of semi-automatic threat classification procedures.
- Unit testing using a number of possible security attack or threat scenario (individual feature assessments).

3. Design and Analysis of the data-driven framework for cybersecurity monitoring.
    - Analyse the effectiveness of the data-driven framework using primary and secondary research
    - Optimisation of the performance parameters.
    - Testing and validation of the proposed framework (Model assessment).

1.4 Contributions

A machine learning framework for detecting cyber attacks was proposed to answer the first research question. The framework takes quantified data from the dataset features such as port numbers and goes through the various steps until the final block, which is the decision-making charts, as shown in Fig. 4.1. The machine learning model informs this framework. Next, the various blocks of the framework are covered.

The framework consists of various blocks consisting of data preparation, parameter and artefact selection, machine learning classification, and visualisation charts. The first block consists of intrusion detection datasets col-

lected from network and hosts systems. There are parameters relating to the network in the second block, such as flow data and host parameters, including event logs, while IoC variables contain artefacts collected from Windows hosts. Abstraction/aggregation is performed before feeding the data into the machine learning model. The machine learning model is used to perform the learning and classification to classify the various attacks. The parameters are then optimised until the optimum result is achieved. The result is then visualised in charts to help with decision-making.

The cyber kill chain approach was applied to detect APT attack stages using a data-driven machine learning technique to answer the second research question. Some existing detection modules proposed in work by [5] were applied and improved by performing feature extraction, feature selection, classification and adding our own proposed detection modules, which will be expanded as part of our future work.

Feature engineering was applied to predict cyber attacks using time series cyber event forecasting to answer the third research question. Machine learning forecasting techniques were applied to achieve the objectives.

Security logs and IoCs were leveraged to detect cyber attacks based on forensic artefacts left behind following the attacks to answer the fourth research question. Several features, such as widows events, were used to classify the attacks in the machine learning model.

1.5 Thesis Layout

Chapter 2 follows the introduction and presents background literature relating to this research. These topics include security metrics, vulnerabilities assessments, cyber-attack detection and classification using machine learning approaches.

Chapter 3 explores the existing cybersecurity frameworks that organisations widely adopt to protect their critical systems. These frameworks include the NIST cybersecurity framework, the ISO 27001 and the CIS 18 set of controls.

Chapter 4 presents the proposed data-driven framework and discusses its various components. In this chapter, the three models that form part of the framework were introduced. These models are cyber attack detection, prediction and indicators of compromise.

Chapter 5 discusses machine learning for detecting Advanced Persistent Threats (APT) and cover experiments performed as part of this research. The Cyber Kill Chain approach was leveraged to map the detection modules to the APT stages.

Chapter 6 presents cyber-attack events prediction using machine learning-enabled feature forecasting. This chapter will also cover experiments performed as part of this research.

Chapter 7 presents threat detection using Indicators of Compromise and discusses IoC's role in intrusion detection. The IoC experiments performed as part of this research will also be covered in this chapter.

Chapter 8 concludes the dissertation and give future directions.

1.6 Publication arising from this work

This section presents papers published/submitted to peer reviewed journals, conference proceeding/workshops and book chapters. These papers relate to materials covered in chapters 2 and 4 to 7.

Journal Paper

- Ahmed, Y., Asyhari, A.T. and Rahman, M.A., 2021. A Cyber Kill Chain Approach for Detecting Advanced Persistent Threats. CMC-COMPUTERS MATERIALS & CONTINUA, 67(2), pp.2497-2513.

- Rahman, M.A., Asyhari, A.T., Wen, O.W., Ajra, H., Ahmed, Y. and Anwar, F., 2021. Effective combining of feature selection techniques for machine learning-enabled IoT intrusion detection. Multimedia Tools and Applications, pp.1-19.

- Ahmed, Y., Asyhari, A.T. and A, Aneiba.,2021. Feature forecasting for cyber attack prediction. SN Computer Science (Submitted).

Conferences/workshops

- Ahmed, Y., Naqvi, S. and Josephs, M., 2019, May. Cybersecurity metrics for enhanced protection of healthcare IT systems. In 2019 13th International Symposium on Medical Information and Communication Technology (ISMICT) (pp. 1-9). IEEE.

- Ahmed, Y., Naqvi, S. and Josephs, M., 2018, September. Aggregation of security metrics for decision making: a reference architecture. In Proceedings of the 12th European Conference on Software Architecture: Companion Proceedings (pp. 1-7).

Book Chapter

- Ahmed, Y., Asyhari, A.T., and Siengue, D., 2021. Machine learning-based threat detection using Indicators of Compromise (IoC). Cooperative Machine Learning for Internet of Agricultural Things. Scopus (submitted).

Posters

- Yussuf Ahmed. POSTER: Cybersecurity Forensic Readiness Model. In the 3rd Cyber insurance SASIG workshop. 2019.

Chapter 2

Background

Cybersecurity is at the forefront of most organisations. They invest a substantial amount of their IT budget on security to protect their data's confidentiality, integrity, and availability. In this chapter, a review of the existing work on networked systems' security will be carried out.

2.1 Current Cybersecurity threats and challenges

Cybersecurity threats are evolving rapidly, and the security community are playing a catch-up game to protect their valuable assets from cyber intruders. Cybercriminals are using advanced techniques and sophisticated tools to perform their attacks and to hide their digital footprints. Cyber attacks have gained prominence following the media coverage of attacks involving large corporations such as Sony [6]. However, these attacks are not limited to these large companies, and all sectors are a target for these criminals,

including government institutions and healthcare facilities.

A recent report on the threat landscape by the European Union Agency for Cybersecurity (ENISA) [7], identified the top 15 attacks with malware, web-based attacks and phishing taking the top 3 slots. Phishing has been used for decades, but it has recently become a popular attack vector for deploying ransomware by tricking unsuspecting users into clicking links that trigger the attack. Although there have been concerted efforts to deal with phishing attacks both from the industry and academic researchers, these attacks continue and successfully compromise unsuspecting users. Several authors have discussed phishing and proposed techniques for mitigating such attacks [8, 9, 10, 11, 12].

Cyber attackers are also targeting networks to take advantage of the vulnerabilities that exist on these networks. Securing networks is not easy given the number of devices present in a typical network. Such devices include the Internet of Things (IoT) and other Internet-connected devices, which increase the attack surface. Due to the current Covid-19 pandemic, employees have been allowed to access corporate networks from mobile devices such as smartphone and tablets. The homeworking arrangements have complicated the matter for security professionals, given the lack of controls over these devices and these users' actions. The lack of control is one of the reasons why there was an increase in Covid-19 related cyber breaches. According to a recent report by a UK based Privileged Access Management (PAM) provider) [13], three-quarters of the surveyed decision-makers believed the shift towards remote working during the Covid-19 increased the likelihood of a cyber breach.

The main challenge for system administrators and decision-makers is determining their networks' security status and ensuring proactive measures are in place before the cybercriminals exploit vulnerabilities present in the network. In this work, several approaches were taken, including machine learning-enabled IoC detection, cyber-events feature forecasting, and improving the detection of sophisticated attacks such as Advanced Persistent Threats (APT) during the various stages of the attack lifecycle.

2.2 Cyber Attacks

Cyber attacks are often planned, and they follow a sequence of stages such as reconnaissance and exploitation before they compromise these systems. A vulnerability must exist on these systems, which a threat could then exploit. The severity of the attack will depend on the capability of the attacker and the strength of the security controls designed to protect these systems. In other cases, the motivation of the attacker plays an important role. Such motivation could be financial, political or espionage [14].

Cybercriminal will scan for vulnerabilities on the target systems and then deploy malicious payloads to exploit them. The goal of these types of attacks varies, but the most common ones are:

- Data theft including intellectual property
- Unauthorised access and modification of data
- Damage or destroy computer systems

All these actions affect the confidentiality, integrity and availability of the data. Confidentiality is ensuring only those who are authorised can access the system [15]. Integrity protection is to avoid unauthorised modification of data on transit, and at rest [16]. Availability ensures data is accessible to those who are authorised when needed [17]. Cyber attackers do not need to have physical access to the target, and most attacks are executed remotely. Detecting such attacks during their early stages can prevent serious security breaches, particularly attacks involving Advanced Persistent Threats(APT) where data exfiltration is of major concern. Several control measures are implemented to protect against cyber attacks. The most common ones are firewalls, anti-virus, Intrusion Detection Systems and Data Loss Prevention (DLP) Systems.

IDS helps with the detection of security breaches by sending alerts to systems administrators [18] although there are other challenges associated with IDS such as false negative and false positives. IDS are categorised into Host Intrusion Detection Systems (HIDS) and Network Intrusion Detection Systems (NIDS).

2.3 Intrusion Detection Systems (IDS)

Intrusion detection systems play a vital role in detecting security breaches. Firewalls and IDS have been key security solutions deployed by many organisations to protect their critical assets and preserve their systems' confidentiality, integrity, and availability.

IDS are categorised into host intrusion systems (HIDS) and network in-

trusion detection systems (NIDS). IDS uses techniques such as anomaly and signature-based detection. Anomaly-based detection analysis the traffic patterns and learns from them, while signature-based detection relies on known patterns and uses the signature to detect and prevent malicious activities [19].

Signature-based detection uses a list of known indicators of compromise (IoCs), but they have their limitation, including the inability to detect unknown attacks [20]. Another limitation is the ability of cyber attackers to modify malware by changing its signature to avoid detection.

Anomaly-based based detection provides better capability than signature-based detection by analysing the pattern of behaviours and building a picture of the unfolding attacks based on deviation from the known behaviour [21]. It is beneficial for organisations that generate large volumes of data, but the downside is higher false positives due to the threats' misclassification.

HIDS are typically deployed on systems containing critical information, although its adaptation varies depending on the organisation's need or asset owner. The purpose of HIDS is to monitor the incoming and outgoing traffic on the systems to detect malicious activities. The most commonly used HIDS include Snort, OSSEC and Splunk. HIDS uses audit trails and systems logs to detect malicious activities [22]. Fig 2.1 shows a simple network architecture with HIDS on the client machines in the internal network.

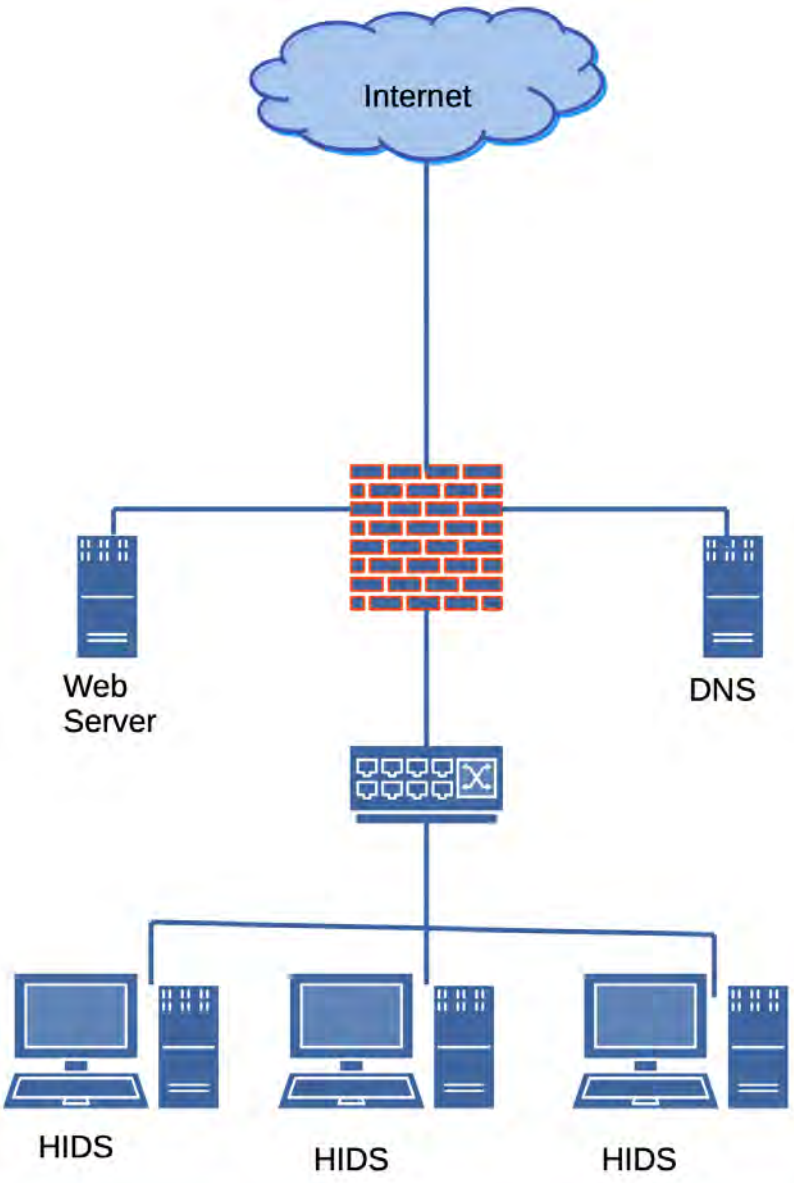

Figure 2.1: Host Intrusion Detection Systems

Network Intrusion Detection (NIDS) are widely used and deployed by many organisations. NIDS monitors the network traffic for malicious activi-

ties. NIDS can be software or hardware-based and usually has two interfaces, one for listening and the other for management and reporting. NIDS is different from the Network Intrusion Prevention Systems (NIPS) because it does not modify or alter the traffic flow. NIDS requires promiscuous network access to be enabled in order to analyse all the network traffic. They are intelligent devices that passively inspect traffic going through the network. Fig. 2.2 shows a simple network architecture with the NIDS sniffing the traffic on the perimeter firewall interface.

2.4 Intrusion Prevention Systems (IPS)

Intrusion Prevention Systems (IPS) are security systems that inspect network flow, detect and prevent exploitation of vulnerabilities, and have more advanced capabilities than IDS [23]. IPS sits directly behind the firewall, inspects files traffic between the source and destination, and takes automatic action based on the ruleset. IPS uses both signature-based anomalies to detect exploits. Researchers have shown interest in IPS due to the capability to prevent attacks rather than just report. Several authors have carried out on improving the detecting accuracy of intrusion prevention systems using deep learning approaches [24, 25, 26].

2.5 Data Loss Prevention Systems (DLP)

Most organisations carry out their daily activities in cyberspace and generate huge amounts of data that necessitate having the correct security controls

to protect this data. Data loss is a concern for many organisation due to the impact of losing confidential and sensitive data. Data loss can cause significant damages to the affected organisation's reputation and result in substantial fines from regulatory bodies. The cost of data breaches can also be higher. Data loss prevention systems detect and prevent data breaches if configured correctly. DLP can prevent the disclosure of information by blocking unauthorised data transfers by monitoring the data flow [27]. Security professionals and system administrators can develop a data loss prevention strategy. This strategy begins with prioritising and classifying the data, training employees, and implementing the DLP program. Implementing a single centralised DLP solution is the best way and avoids inconsistencies, leading to lack of visibility and potential data loss.

DLP can be used to protect data at rest, data in motion and data in use. Data at rest is data stored on computer systems, while data in motion is data that is transmitted over the network and going through the internet. Finally, data in use refers to active data executed by end-users or processes.

Insider threats are a concern for organisations, especially on data breaches relating to sensitive data, including intellectual property theft. Data can be leaked accidentally or by rogue employees and external cyber attackers. DLP can reduce the risk of these data breaches by taking proactive measure to detect and prevent data exfiltration.

2.5.1 DLP Challenges

Encryption - encryption and DLP often co-exist within the same network. Encryption is designed to protect the confidentiality and prevent eavesdrop-

ping by ensuring that only authorised users can access data. This means DLP solutions cannot read the content of encrypted data and will be unable to detect data leakage involving encrypted files or messages. This is why sometimes DLP solutions are deployed in places where the data is not encrypted.

Steganography - steganography is the process of concealing a secret message inside an object or message that is not secret [28]. Steganography is a popular technique used by cybercriminals to disguise malware. For example, DLP solutions cannot detect content hidden in images and can be used to bypass the DLP system as demonstrated in [29, 30]

DLP raises privacy concerns - DLP solutions collect a vast amount of data during the monitoring phases, which means personal data can be collected during the data collection phases, raising privacy and regulatory issues [31].

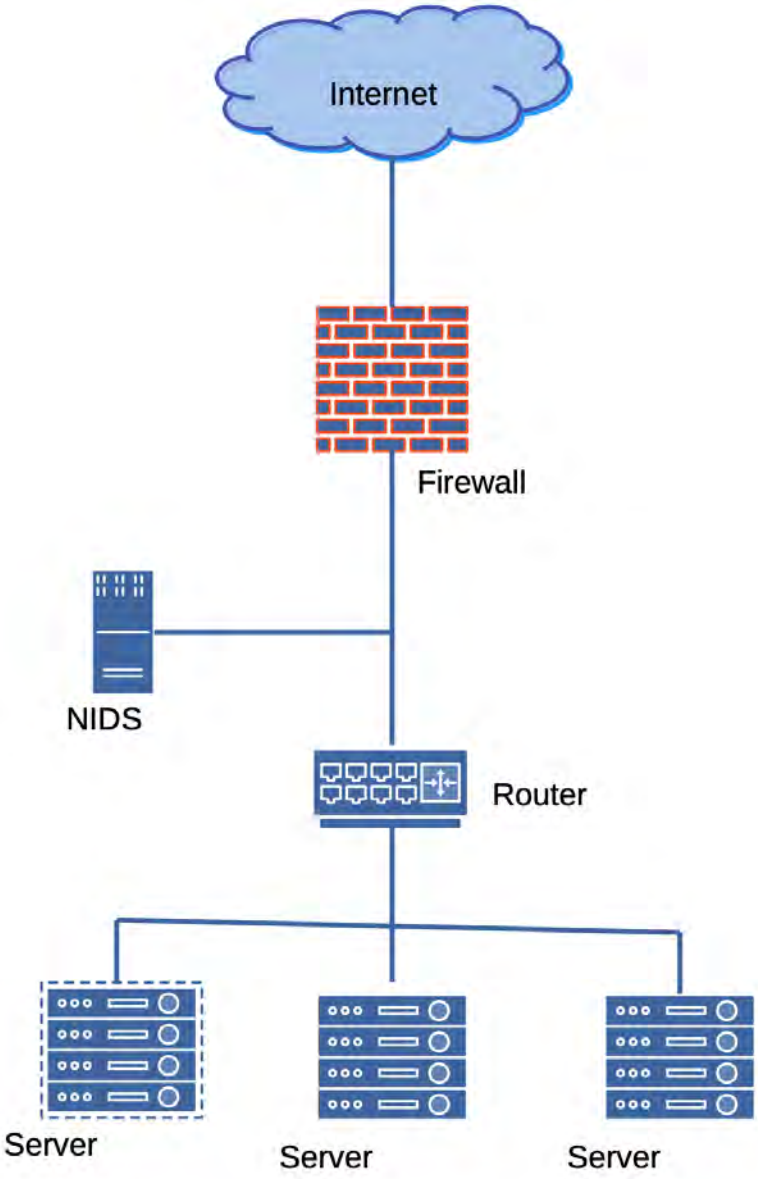

Figure 2.2: Network Intrusion Detection System

Most of the current research on intrusion detection are utilising machine learning. The following section covers machine learning approaches for in-

trusion detection.

2.6 Machine Learning Approaches for Intrusion Detection

This section provides background reviews of the machine learning approaches for intrusion detection, and some of the content derived from the author's work published in a peer-reviewed journal [32].

Machine learning is a subset of artificial intelligence and is a technique used to teach machines to learn from data and make predictions. There are four main types of machine learning methods which are: (i) Supervised, (ii) Unsupervised, (iii) Reinforcement, (iv) Semi-supervised [33].

Supervised learning trains itself based on data that was pre-prepared and labelled. Supervised learning tasks take less time to compute than other methods, and the performance results can be compared with the actual results to determine the accuracy. Labelling data can be tedious and time-consuming, resulting in overfitting depending on how the data was labelled.

Unsupervised learning ingests data and extracts meaningful feature without any human intervention. Reinforced learning is similar to supervised learning, but rather than being trained on labelled data. It learns as it goes along with trial and error. Semi-supervised is a combination of supervised and unsupervised learning. It involves a small labelled data from a large unlabeled dataset.

In this section, machine learning approaches will be discussed. There are various machine learning approaches for classifying data, but our main

focus will be on the ones used in this research. In this thesis, several intrusion detection datasets were used during the experiments. These datasets includes the CSE-CIC-IDS2018 [34], APT [5] and IoC datasets. The outcome of this work will be discussed in chapters 5-7.

Although intrusion detection systems have been effective at detecting malicious activities with known signatures, the same cannot be said on anomaly-based detection, where the systems have to learn the traffic patterns to detect unknown malicious activities. Several machine learning techniques have been proposed to improve the detection capabilities of intrusion detection systems.

Several authors proposed deep learning approaches to improve intrusion detection system based on autoencoders and recurrent neural networks (RNN) [18], [35], [36], [37]. Their work relied on the NSL-KDD dataset which is widely used for intrusion detection research. Others carried out surveys on intrusion detection based on machine learning [38], [39] to captures research in this area.

Machine learning has been widely used to improve the classification and detection of the attacks [40], [41]. Although some progress was made to improve intrusion detection using machine learning, the challenge of accurately detecting all attacks on a network remains. Several authors have explored feature engineering in order to reduce training times and intrusion detection accuracy [42], [43], [44].

A brief overview of the machine learning classifiers used in this research will be given in the next part.

2.6.1 Bayes Net

Bayes network is a probabilistic model representing a set of variables and their conditional dependencies and is also known as Bayesian Belief Network (BNN). Bayes Net represent a casual chain, and you can supply data from past events to predict the future outcome of that event [45]. It can deal with missing values when performing the predictions and has been widely used in weather forecasting and stock exchanges.

2.6.2 Naive Bayes

Naive Bayes is a classification algorithm that uses posterior probability for classification. Suppose we have $P(B|D)$ and $P(D|B)$ as the probability of B given D and vice-versa, $P(B)$ and $P(D)$ denote the likelihoods of B and D, respectively. These parameters can be linked through a Bayesian equation as used in [46].

$$P(B|D) = \frac{P(D|B)P(B)}{P(D)} \tag{2.1}$$

$$\hat{y} = \arg \max_{k \in \{1,\ldots,K\}} P(C_k) \prod_{i=1}^{n} P\ \ (x_i|C_k) \tag{2.2}$$

where C_k are label instances in the dataset, x_i are data points (instances) and $\hat{y}$ is the estimated label. For example in the case of APT, the classes C_k refer to attack stages while x_i refer to data instances of the features.

Naive Bayes is highly scalable and performs well when applied to big data, but its main limitation is the assumption of conditional independence of the features, which might result in some loss of accuracy. Naive Bayes can

also suffer from a zero probability problem which essentially means if the conditional probability is zero for a particular feature, it will not be able to predict [47].

2.6.3 Support Vector Machine (SVM)

SVM is a supervised learning algorithm for classification and uses hyperplanes to define the decision boundaries between the two data classes. Many hyperplanes can be used to classify the data, but the best choice is often considered the hyperplane representing the largest separation between the two classes. The support vectors are the nearest instances that represent the separation between the two classes [48].

The performance can be improved by using the kernel functions that are suited to your data. These kernel functions include RBFKernel, PolyKernel and StringKernel, which can be utilised to reduce some of the complexities introduced by the data, especially when data points are not linearly separable. The limitation of SVM is computation time, which takes longer to execute depending on the data's size.

The justification for using SVM is its ability to deal with classification and regression on linear and non-linear data. It is widely also used in intrusion detection. SVM is a good choice when working with smaller datasets which is the case in our experiment, and produces better results than other algorithms when dealing with small and complex data.

2.6.4 K-Nearest Neighbour

KNN is a classification and regression algorithm that calculates the distance between supplied data and the input to make predictions. It assumes similar features are located closer to each other. The challenge with KNN is finding the actual value of k, which can significantly impact classification accuracy.

The limitation of KNN is that it does not perform well with large datasets due to the time taken to compute the distance between new points and existing point, which can degrade the algorithm's performance [49].

2.6.5 Random Forest

Random forest is a supervised machine learning algorithm that combines multiple decision trees to make a forest. It is widely adopted to solve classification and regression problems, and appropriate for high dimensional data due to its ability to handle missing values and continuous, categorical and binary data [50]. Overfitting could still be a problem in Random Forest, especially with data that has noise or outliers and should be monitored. It might take a longer time to compute due to the number of trees that are generated.

2.6.6 Deep Learning

Deep learning is a subset of machine learning, and its adaptation has increased over the years due to technological advancements and availability of the large data. Deep learning uses different types of neural networks and learns from feature representations without the need of performing future

engineering, which is time-consuming [51]. Neural networks use artificial neurons that mimic the human brain and consist of input, hidden, and output layers.

There are various types of deep learning algorithms, and these include recurrent neural networks, Long Short Term Memory Networks (LSTMs), Convolutional Neural Network (CNN), Multilayer Perceptrons (MLPs), Deep Belief Networks (DBNs) and Autoencoders. Some of these algorithms will be covered next.

Convolutional Neural Network (CNN) use multiple layers and is widely used for image processing, and these layers include Convolution Layer, Rectified Linear Unit (ReLU) and Pooling Layer. The convolution layer is where the operation is performed. CNN has been applied to help classify images in fields such as satellite imagery and medical imaging. For example, several authors have proposed CNN based techniques for medical image recognition and proposed models [52, 53, 54, 55] while others have applied CNN techniques in satellite image recognition [56, 57, 58].

Long Short Term Memory Network (LSTM) is a recurrent neural network that memorises past inputs and uses them to derive future predictions. It is a popular method for time-series prediction and process sequence of data. LSTM will be one of the techniques used in our time series prediction model in chapter 6. Several authors have proposed techniques based on LSTM to model future events. For example, it has been widely used in financial predictions. In [59] the authors proposed a financial time series prediction model that chooses the features that contribute most based in their weight. Time series based predictions has been applied to various other domains such in

medical[60, 61, 62], weather [63], energy efficiency [64] and commodity pricing [65]. LSTM techniques have been applied to intrusion detection, and several researchers have carried out work in this area. In [66, 67] the authors applied LSTM on CIDDS data to create an intrusion detection model . In [68] the authors used the CICIDS217 dataset to develop a hybrid intrusion detection system that uses LSTM and CNN. In [69] the authors created a bidirectional Long-Short-Term-Memory (BiDLSTM) based on intrusion detection system to deal with challenges of false alarms using the NSL-KDD dataset. According to the authors, their proposed system produced a high accuracy when detecting the targetted attacks. In [70] the authors proposed a DoS detection method that uses LSTM and Bayes. According to the authors, their model achieved a high accuracy rate. Several other authors have used deep learning for intrusion detection based on the CICIDS2018 dataset using CNN, and LSTM [71, 72, 73]

Autoencoders are a feedforward neural network where the input is the same as the output and consists of an encoder, a code and a decoder. Autoencoders learn automatically from the supplied data through unsupervised means. Researchers have applied autoencoders in deep learning. In [36] the authors proposed a framework based on stacked autoencoder for feature engineering using support vector machine resulting in improved accuracy. In [74]the author proposed an IDS system based on a deep auto-encoder and used the KDD-CUP'99 dataset to evaluate the performance of their model. Several other authors have used autoencoders for intrusion detection, and these include[75, 76, 77, 78].

2.7 Advanced Persistent Threat Lifecycle

This section explores Advanced Persistent Threats (APTs) and related work, including current research findings. APTs are a group of sophisticated attacks targeted at organisations and governments institutions. The APT groups are well funded and often supported by organised cybercriminals or are state-sponsored. Most of these complex APT attacks were found to have been carried out by state-sponsored actors and other APT groups sponsored by organised cybercrime and primarily driven by financial gains. Attackers typically enter networks by taking advantage of human weaknesses and their susceptibility to social engineering and phishing attacks. They also enter the networks by exploiting vulnerabilities that exist on the network devices and endpoint client devices.

2.7.1 APT Background

APTs are among the most sophisticated attacks utilised by organised cybercriminals and those affiliated with nation-state actors. APTs are targeted and persistent forms of attack and may go unnoticed for an extended timescale [79]. According to FireEye, APT attacks' global median dwell time is 56 days [80]. According to a Kaspersky report on APT trends in 2021 [81], new attack vectors such as those targeting network appliances and 5G vulnerabilities are likely to occur alongside the multistage attacks, and the trend is likely to continue.

APT attackers often use multiple attack vectors to obtain or modify the information, which is even made easier by the ever-expanding attack surface

in the digitised world. For example, cybercriminals could exploit devices ranging from the Internet of Things (IoT), smart cameras, and Bring Your Own Devices (BYOD), present in most organisations.

According to a Verizon Data Breach Investigation Report, [82], there was an increase in cyber espionage involving APTs using a combination of phishing and malware. According to another report by Malwarebytes [83], organised criminals and nation-state actors linked APT groups have been using coronavirus-based phishing attacks to compromise and gain a foothold on the victim machines [84].

Recent interests have shown an increased focus to deal with APT attacks. A variety of cybersecurity measures and methodologies have been investigated to detect, monitor, and mitigate the APTs, and their impacts. Conventional cybersecurity approaches have demonstrated some limited success at detecting APTs due to their sophistication, and when they are detected, they tend to adapt very quickly and change course. Most of the APT groups are well resourced and will try every effort to achieve their goal. These motivate the recent development of machine learning and computational intelligence techniques to improve the detection of the APTs, which can then translate into timely intervention measures.

2.7.2 APT Phases

APT attacks are well planned and follow a pattern of stages. The most common attacks follow the following stages: (1) Reconnaissance, (2) Weaponisation, (3) initial compromise, (4) command & control, (5) lateral movement (6) data exfiltration.

Several industry leaders proposed attack Life Cycle frameworks for dealing with cyber threats and, in particular, APTs. These frameworks include the Lockheed Martin Cyber Kill Chain, the Diamond model, Mandiant Attack Life Cycle, and MITRE ATT&CK model.

The Lockheed Martin Cyber Kill Chain has seven stages covering the whole attack life cycle. These stages are Reconnaissance, Weaponisation, delivery, exploitation, installation, Command & control (C2) and action on objectives.

The Diamond model is another approach for detecting intrusion and has four interconnected features present in every attack. These features are adversary, capability, infrastructure, and victim [85].

The Mandiant attack life cycle consists of multiple components mapped to the various phases of the attack life cycle [86]. The industry research has its limitations, given they are not peer-reviewed and are mostly used as a platform to market their products.

The MITRE ATT&CK (Advance Tactics Techniques and Common Knowledge) is a framework used to explain the adversarial actions against the target system to gain more insight into the attackers tactics [87]. It is widely used for threat hunting and intrusion detection.

Chapter 5 of this thesis proposes a cyber kill chain approach for detecting APT leveraging machine learning methods. In the next part, the seven stages of the cyber kill chain will be discussed as shown in Fig. 2.3.

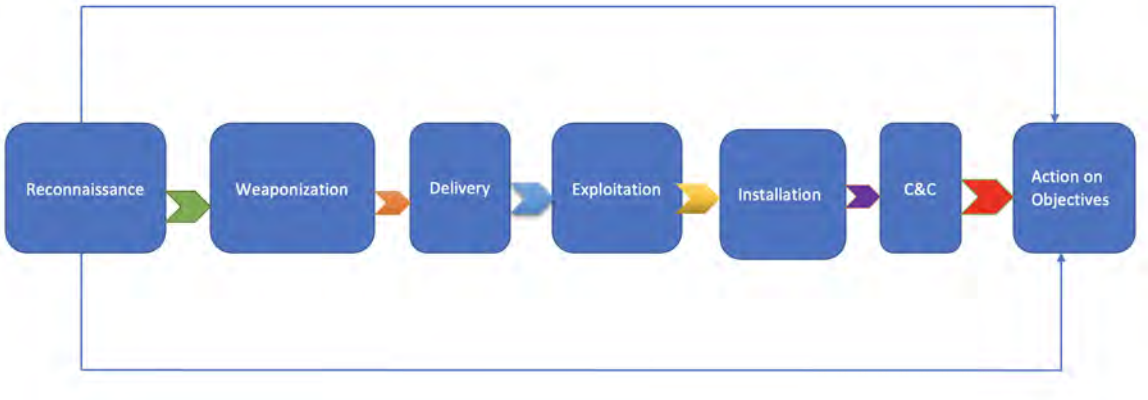

Figure 2.3: Cyber Kill Chain Stages

1. **Reconnaissance**

   Reconnaissance, which is also referred to as information gathering, is the first stage of carefully planned cyber-attacks. The goal of this phase is to find as much information as possible about the target organisation.

   The two main types of reconnaissance techniques are active and passive reconnaissance. In passive reconnaissance, the attacker has no direct interactions with the target. In contrast, in active reconnaissance, the intruder interacts with the target to obtain information that could be used during the later stages. In the reconnaissance phase, cyber attackers use various tools and techniques to obtain information on their target. These include NMAP, which can perform port scanning and OS fingerprinting.

   During the passive reconnaissance, the attacker looks for publicly available information relating to the target organisation. Such information could be found on the target public-facing websites, whois domain,

social engineering, google search and shodan [88]. In the active reconnaissance, attackers have many tools such as NMAP at their disposal.

Defensive measures - security professionals can implement many proactive measures to increase their defensive capabilities and minimise the impact of such attacks. The key to success is detecting cyber attacks during the early stages. Some of the proactive security controls that could be implemented to mitigate the risk of cyber attacks include minimising the volume of information available in the public domain, regular penetration testing to close loopholes, and user awareness training to protect against social engineering. Other security defensive measure includes URL filtering and inspection of network traffic.

2. **Weaponization**

   Attackers use the information gathered during the reconnaissance stage to create a carefully crafted malicious payload tailored to meet their requirements. The attackers usually use automated tools for packaging their malware. Remote Access Trojan (RAT) and exploits are used during the weaponisation. The RAT is designed to allow the attacker to have backdoor access to the compromised system without the victim's knowledge.

   Defensive measures - looking at the defensive capabilities, security professionals could implement proactive measures to improve their detection capabilities and prevent an attacker from moving to the next stage of the attack. The key approach to such measures success includes understanding the threat actors targeting your organisation and the

nature of the payloads they use. The features from such attacks could then be used to trigger early warning alerts. Threat intelligence feeds could also be utilised in order to learn more about the attackers' tactics, techniques and procedures (TTPs).

3. **Delivery**

   In the delivery stage, the attacker deploys the weaponised payload to the target. There are multiple means for payload delivery that are available to the attackers. These could be malicious emails, click-by downloads, watering holes or infected USB devices.

   Defensive measures - security professionals could improve their enterprise security and detect weaponised payload delivery before the attacker moves to the next stage of the attack. Such measures include; limiting administrative privileges, email filtering systems and security monitoring tools to detect anomalies such as disguised executable files and brute force attacks. Instilling a security awareness culture in the organisation can also result in employees acting more responsibly when opening malicious links or attachments. Such employees are likely to notice suspicious activities on their systems and share the information with the security professionals, resulting in prompt escalations and mitigation plans.

   Using security metrics to measure these security mechanisms' effectiveness can help improve the detection capabilities. Such metrics could include the number of security incidents involving employees who have taken the security awareness training. This will measure the effective-

ness of the security training programme in reducing security incidents rather than taking a compliance-based approach that only records the number of employees who have taken the security training.

4. **Exploitation**

   In this phase, the attackers exploit vulnerabilities on the target system, although a vulnerability must exist on this system for successful exploitation to occur. This vulnerability could be a known vulnerability or zero-day. Once the target is compromised, the attacker will get notifications and prepares for the next stage of the attacks.

   Defensive measures - it is crucial to implement a patch management mechanism to minimise the attack surface and reduce the number of vulnerabilities available to the attackers. Known and high-risk vulnerabilities are likely to be exploited by attackers. It is important for security professionals to use metrics such as Mean Time to Patch (MTTP) to measure their response times and reduce the window of exposure. Other measures include using security monitoring tools such as intrusion detection/prevention systems, limiting the use of administrative privileges and detecting brute force attacks which could be an attempt to escalate privileges.

5. **Installation**

   Once the target is exploited, the next stage for the attackers is to execute the malware. In most cases, the malware will self install and replicate depending on the goal of the attacker. In other cases, the attackers might also need to perform privilege escalation and install

a backdoor to maintain persistence. To avoid detection, the attackers often use a dropper and downloaders, which can be used to disable the security monitoring tools such as antivirus in order to avoid detection during the malware installation [89].

Defensive measures - security professionals could implement several measures to defend against attacks relating to this stage. Such security measures include; up to date antivirus systems to detect malware, firewalls, intrusion detection/prevention systems and to limit user rights. Administrators should configure these monitoring tools to send alerts when anomalies such as disguised executable files and other forms of attacks such as brute force are detected.

6. **Command and Control**

Attackers get management control of the target and establish a backdoor to maintain persistent access. The attackers can then perform lateral movement and exfiltrate data. A two-way communication channel is established between the attacker and the compromised target. It is also possible for them to perform other actions such as authorised modification and destruction of the data depending on their motives.

Defensive measures - security professionals could implement measures for detecting and blocking malicious IP addresses, malicious SSL certificate, malicious domain flux and established malicious connections such as ToR communications. Once the command and control stage is detected, there should be defensive mechanisms to block access to the external command and control servers and quarantine or isolate the

infected hosts. The proactive measure includes network layering and enhanced monitoring.

7. **Action on Objectives**

   In this stage, the attackers will execute several other tasks to achieve their goals. Depending on the attackers' motives, several actions could be performed. These include data ex-filtration, modification or destruction, privilege escalations, credential harvesting, SSH attacks, internal reconnaissance, internal spearphishing and anti-forensic actions. To disguise their activities, the attacker could use ToR onions or encrypted channels such as Virtual Private Network(VPN).

   Defensive measures - some of the defensive security measures which security professionals could implement include; encrypting data at rest, using Data Loss Prevention (DLP) mechanisms, blocking communications with suspicious URLs and blocking communications with commands and control servers.

   Internal reconnaissance is one of the important steps for cyber-attackers. They tend to scan for valuable assets to help them achieve their goal by looking for the easiest path to obtain privileged and persistent access to the network. Detecting these internal scans and correlating them with the other steps in the cyber kill chain can help detect the APTs and implement corrective measures to mitigate the risks. Internal scans can be collected from the network monitoring tools, which can be configured to send alerts to the network administrators and security teams. Another solution proposed by some researchers is the deployments of

deceptive honeypots or honeynets.

2.7.3 APT Research and Findings

APT received attention from both the industry and academic researchers, although most of APT's information is from the industry. The extensive research by industry leaders such as FireEye [90], and Kaspersky [91] led to the discovery of many APTs, including those used by nation-state actors that are difficult to detect. For example, FireEye published research on APT41, which is linked to nation-state actors and used for espionage and financial gains [92].

Several other APT groups have been detected, and some of them are still active and keep developing new malware variants. For example, Kimsuky, linked to nation-state actors, has been active since 2013 but recently deployed new malware. The primary goal of the group is cyber espionage.

Another APT group whose goal is cyberespionage was reported in 2018 [93]. This APT group was found to be active since 2012 and distributing malware targeted at Window and Android platforms. Their infrastructure includes watering hole servers, phishing domains and command & control servers. APT27 is another active group linked to nation-states and primarily used for espionage, although they adapted their tactics recently and started deploying ransomware. They targeted five online gambling companies and encrypted their servers. According to a security firm, the hackers reached their target through a third-party provider [94]. In 2019 hackers from the APT27 compromised SharePoint servers belonging to two governments in the middle east.

APT28 focused on intelligence gathering and espionage. Security teams have attributed this group to nation-state actors due to the information they are after and their targets [95]. The Lazarus group are perceived to be state-sponsored and have been very active. Some of the hacking activities they were involved in include the AppleJeus campaign, which targeted cryptocurrency trading platform users and their systems [96] and the attack on Sony Pictures [97, 98, 99]. Other high profile attacks by APT groups include the alleged interference in the US election in 2016 in which a staff member's account was compromised through a phishing email sent by the APT group [100]. APT36 linked to state-sponsored actors orchestrated campaigns exploiting people's fears on Covid-19 and sending malicious documents purported to be from health authorities and then deploy the Crimson Remote Administrator Tool (RAT) on their target systems [101]. The list of various APT groups and their alleged affiliation can be found in [102].

APT attacks have also been gaining interest from academic researchers, and several authors have published articles on this subject. In [103], the authors surveyed APTs and proposed a taxonomy for APT defence classification. Similarly, [104] carried a survey on APTs and reviewed some of the known APT groups' activities but did not cover defensive or detective technical measures. This work mostly relied on publicly available data on APTs shared by the industry, although they described such sources' limitations.

Another APT attack life cycle methodology was proposed in [105]. The authors proposed four stages which were, prepare, access, resident, and harvest. In the preparation stage, the attackers gather information relating to the target. The access and harvest stages broadly encompass the step

involved in compromising the target. According to the authors, the most common attack vectors for APT include watering hole and spearfishing. In [106], the authors discussed the tools and techniques available to the attackers and linked them to the various stages of the Cyber Kill Chain, but their review was more generic. It could have benefited from evaluating specific APT attacks or groups. Similarly, in [107], the author proposed a taxonomy for banking Trojans based on the Cyber Kill Chain. In another work [108], the authors leveraged the Cyber Kill Chain to break down complex attacks and built a picture of the APT attackers' tactics, techniques, and procedures (TTPs). The authors analysed over 40 APTs to build their proposed taxonomy.

In [109], the authors proposed an approach for detecting APT using fractal methods based on a k-NN algorithm, which, according to the authors, resulted in a reduction in false positives and false negatives. In a similar report [110], the authors performed experiments to detect the stages of APT attacks. They used the NSL-KDD dataset and selected Principal Component Analysis (PCA) for feature sampling.

In [111], the author categorised the APT attack lifecycle into five phases: reconnaissance, compromise, maintaining access, lateral movement, and data exfiltration. In [112], the authors categorised APT phases into reconnaissance, delivery, exploitation, operation, data collection, and exfiltration and proposed an APT detection methodology.

In [5] the authors proposed a machine learning-based framework for detecting APT attack stages and built detection modules. The authors expanded on this work by applying Hidden Markov Chain (HMM) model to

determine the likely sequence of APT stages [113]. According to the author, their proposed approach was able to predict the sequence of APT stages with a prediction accuracy of at least 91.80% which is an improvement from the previous work. Our focus is on their first work, which contained the bulk of the experiment, including the detection modules and resultant dataset. That dataset was also the most recent dataset when this research began.

Deep learning approaches have gained interest from APT researchers. Several works have been carried out to improve APT detection. However, accurate detection still remains a challenge due to the ever-changing threats landscape and the complexity of APT attacks. Deep learning methods such as autoencoders have been applied to large datasets and produced some good results. Such autoencoders include stacked and convolution autoencoders. In [114] the authors developed a deep learning stack to APT detection, but their approach was not evaluated on actual data collected from production environments. Reference [115] combined individual deep learning networks such as CNN, LSTM and Multilayer perceptron to help analyse and detect APT attacks. In [116] the authors used Bayesian network, C5.0 decision tree and deep learning to detect APT attacks using the NSL-KDD dataset, and the deep learning model produced the highest accuracy. In [117] the author proposed autoencoder based deep learning approaches to detect APT attacks in cloud-based computing and achieved a high accuracy result. Reference [118] applied autoencoders with deep learning trained on lateral movement to the Los Alamos National laboratory dataset and achieved some good results. While progress has been made on APT detection, this is still an area that is developing, and research is limited compared to other cyber attack types.

The need for a model capable of detecting the various stages of the APT lifecycle more accurately is still an ever-pressing one.

2.7.4 Summary and Gaps

This section discussed the various stages of the Cyber Kill Chain (CKC) and possible defensive security measures for each stage. APT attacks are very sophisticated and are considered one of the significant challenges faced by security professionals. APT attackers often have backdoor access to the network over a significant period before they are detected. In chapter 5, the dataset will be reconstructed, and the CKC approach is applied by mapping the detection modules and alerts types to the stages of CKC.

APTs are real threats facing many organisations and are very difficult to detect due to careful planning and are well-funded organised groups. According to recent research, it takes 56 days on average to detect APT attacks, which demonstrates the intrusion detection systems are having limiting success at detecting APT attacks. Therefore, there is a need for new APT detection approaches with minimal false positive and false negatives. In chapter 5 of this thesis, a cyber kill chain approach was leveraged to detect APT attacks during the various stages of the APT lifecycle. The existing detection modules and newly proposed ones were applied to different APT stages to improve the detection.

2.8 Cyber attack prediction and forecasting

In the section, an introduction to time series forecasting and attack prediction will be provided. This section's content is derived from the work of the author submitted to a peer-reviewed journal [119].

Cybercriminals' threat is ever-increasing, and despite significant investments by organisations, they are still getting beyond the security defences. No organisation is immune to these kinds of attacks, as demonstrated recently by the attack on a major cybersecurity firm compromised through SolarWinds updates [120]. One of the challenges in predicting cyberattacks is the sophistication of the attacks and the evolving techniques.

Traditional security defences such as firewalls are no longer enough on their own. There is a need to move away from reliance on reactive measures and move into predictive methods before cybercriminals can cause damage. According to a recent report by IBM [4], the average cost of a cyber breach is $4.24 million while the average time to identify and contain a breach is 280 days. Protecting computer systems and the confidentiality, integrity and availability of the data they contain has been a priority for every organisation. However, cybercriminals are still finding ways to compromise these systems.

The current trends show that cyber-attacks are likely to increase due to the vast number of internet-connected devices, which increase the attack exposure. These devices range from tiny sensors, smart cameras to mobile devices and found in most organisations. Organised cybercriminals are also using Advanced Persistent Threats (APT), which are difficult to detect and defend against using traditional security measures. However, there have been several work to improve APT detection, such as in [5, 121, 32]. There is a need

for predictive approaches that can help system administrators better prepare themselves for potential attacks and preempt such attacks before they occur. Predictive analytic are proactive measures and can allow resources to be targeted where they are needed most and put appropriate countermeasures to mitigate the risks from these attacks. Analysing the attack patterns will help predict the incoming events and help with implementing corrective measures.

2.8.1 Research findings on forecasting and predictions

Cyber attacks are on the rise, and the last few years have seen a substantial increase in the severity and intensity of attacks. Researchers from industry and academic circles have been working on improving attack prediction. This work aims to forecast cyber-attacks based on certain events or features observed in the network using a data-driven approach and anticipate cyber-attacks before they occur.

Cyber attacks can be passive or active, and they usually follow specific steps to achieve their goal. These steps include identifying the target, detecting vulnerabilities, exploiting the systems and maintaining access. These activities can leave digital artefacts, which can then be used to predict attacks before monitoring sensors even detect them. Several methodologies, such as the Cyber Kill Chain, describe how attacks could be detected or stopped during various stages of the kill chain. Forecasting cyber-events can also be used to predict events before they even occur by analysing the pattern of the features and attributing it to an attack type based on the characteristics.

There are several surveys on cyber attacks forecasting and prediction. In [122, 123, 124], the authors provided an extensive survey on current research

on cyber attack prediction and forecasting. In [125], the authors used cyber-event from an operational environment that analysts verified, and they used that data to perform forecasting for malware events. In [126], the authors performed forecasting for Distributed Denial of Service (DDoS) attacks using text stream from Twitter feeds. In [127], the author used hacker's behaviour and sentiments analysis of their posts to predict malicious cyber-events. Similarly, [128] used sentiment analysis in social media to predict cyber attacks. In [129], the author used cybersecurity-related keyword searches from Twitter and the dark web to predict cyberattacks.

Time series have been used to forecast events that are likely to occur in the future. Several machine learning techniques were used based on neural network and autoregressive time series models. In [130], the authors applied time series techniques to financial data using four models: multiple linear regression in excel, multiple linear regression in Weka, Autoregressive in R and Neural network. They found Weka's linear regression outperformed the other three. In [131], the authors used the ARIMA time series to predict future attacks based on historical data about cyber incidents. In [132], the authors used time series techniques to build a predictive model for detecting vulnerabilities in common internet browsers such as Internet Explorer, Firefox, Safari, Chrome and Opera. In another study, the authors performed a statistical analysis of security breaches between 2005-2017 and proposed for both the inter-arrival times and breach sizes of hacking breach incidents to be described in the stochastic process rather than probabilistic distribution. [133].

Several forecasting methods are widely adopted, including ARIMA, Lin-

ear regression, SMOreg, Gaussian process, and Multilayer perception. ARIMA is a statistical method that uses time series to predict future trends. It has been used widely to model economic, financial, energy consumption, and weather forecasts [134, 135, 136, 137] for a long period. However, there are some limitations with ARIMA model and in particular, the difficulty of modelling non-linear relationships [138]. It is considered to be a univariate time series forecasting model. It performs better when the time series data is stationary hence why it essential to check the stationary of the data before performing the forecasting [139]. ARIMA has several other variations such as ARIMAX and SARIMA (Seasonal Autoregressive Moving Average) [140].

Linear regression is one of the base learners available for forecasting and is available in Weka. It is a model used to predict the relationship between two variables and measures their association [141]. Typically this relates to a dependent variable and one or more independent variables. SMOreg is another tool available in Weka and provides effective algorithms for solving Support Vector Machine (SVM) problems. SMOReg can model regression and prediction with non-linear data more effectively [142].

The Gaussian process is a supervised machine learning approach designed to solve regression, and probabilistic classification problems [143]. It is also one of the base learners available in Weka and used for forecasting. Multilayer Perception contains multiple layers of neurons and is sometimes referred to as feed-forward ANN (Artificial Neural Networks). Its layers consist of the input layer, hidden layer and output layer. Except for the input node, each node is a neuron that uses a non-linear function [144].

Several authors discussed the application of deep learning techniques such

as LSTM, as discussed in section 2.6.6. LSTM is one of the popular methods that are widely used in time series forecasting.

2.8.2 Summary and Gaps

Time series data forecasting is not new, and it is widely adopted in fields such as weather forecasting and stock predictions. Forecasting has been gaining traction in cyber attack prediction, although this is still an emerging area [145]. Although several predecessors [129, 146, 122, 147, 126] have carried out work based on time series prediction and forecasting, its application in cyber attack detection has been limited compared to other well-established fields such as weather and stock prediction. The limited work that exists in the cyber domain are mostly on sentimental analysis of hackers behaviour based on social media feeds and the Darknet, or on single attacks such as Denial of Service (DoS) attacks and malware, variant [129, 126, 147, 146, 148]. Most of the existing works are limited by the quality of the datasets, which impacts the prediction accuracy. This work utilises a large dataset with multiple attack labels to make the time series forecasting within a specified time frame to address these challenges. Our opinion is that this work will significantly improve cyber attack prediction and reduce cyber breaches by performing forecasting based on the data's observation and trends, giving the system administrator the chance to put proactive countermeasures in place.

2.9 Indicators of Compromise (IoC)

In this section, Indicators of compromise and related work, including current research findings, are explored. The content of this section is derived from the work of the author submitted to a peer-reviewed book chapter [149].

The rapid growth of the internet and the proliferation of smart devices has increased the attack surface. Cybercriminals are exploiting these vulnerabilities using high-tech tools which are not easily detected by the monitoring tools. IOCs are artefacts left behind by attackers following their malicious activities such as malware execution. Incident response teams use IOCs to detect abnormal activities and piece together the attackers' digital trails.

The last decade has seen a substantial increase in the number of reported cyber-attacks. Although organisations have invested heavily in security measures, cybercriminals are still finding ways to go beyond these security measures and compromise previously secure systems. The need to link events together and to build the attackers profile is a pressing one.

Most cyber-attacks leave behind forensic artefacts, which can be used to determine the type of attack and build the bigger picture using the attackers' trails. Given the rapid growth of the internet and the proliferation of smart devices capable of connecting to the internet, it is just a matter of time before cybercriminal breach the network security defences.

Attackers have access to many freely available tools that can be used to perform actions ranging from reconnaissance to covering their tracks. Although preventative security measures are crucial in securing systems, it is also essential to have a detection capability. Such measures can help to contain security breaches and to remove the threats.

IoCs are widely used to detect security breaches and other malicious activities that have occurred in the organisation. For example, it is common for security personnel to look at the logs and determine malicious activities such as known malicious IP addresses, domains, file hashes, URLs and login irregularities, including failures. The information gained from these IoCs can then be used to increase the network's security defences by understanding their patterns and then tuning the rules on the security monitoring systems to detect similar attacks in the future [150]. The IOCs that are detected following a breach or shared by the security community can be deployed to the sensors to detect such attacks in the future during the early phases of the attack. Cyber attackers also change their tactics regularly to avoid detection, and security professionals need to have a clear understanding of threat facing their organisation.

IoCs are mainly categorised into Network-based, host-based and Email indicators [151]. Network-based IoCs include IP addresses, domain names and URLs. Host-based indicators include malware names, registry keys and malicious file hashes. The network and host-based IoCs can be categorised further according to their impact on confidentiality, integrity and availability. For example, changes in the traffic pattern involving malicious IP addresses could be a sign of data exfiltration, affecting confidentiality. Other IoCs that can affect confidentiality are unauthorised changes to file permission and privilege escalations. IoCs that affect integrity include configuration changes and other unauthorised changes initiated by privileged accounts. IoCs affecting availability includes malformed packets and high CPU usage.

In a typical organisation, extensive volumes of data are generated daily.

Making sense of this data can be an enormous task for Information Technology (IT) professionals. The monitoring tools also generate lots of false positives and false negatives, which amplifies the challenges. This information must also be correlated in order to focus on genuine attacks. IoC provides IT professional or security teams with the chance to find out attacks that security monitoring systems may have missed.

Several predecessors have carried out work on IoC and proposed various approaches. Similarly, industry leaders have also researched IoCs and provided tools such as OpenIOC [152]. There exist other community-driven platforms such as IoC bucket [153] which people can share threat intelligence. Some industry leaders have also been at the forefront of sharing information on IoCs, including tools. For example, FireEye recently released a tool for scanning IoC related to CVE-2019-19781, which affects Citrix application vulnerabilities [154].

Although some improvements were made in the intrusion detection capabilities, their accurate detection is still a significant challenge due to the ever-changing cyber threat landscape. Detecting cyber attacks in the early stages can significantly reduce the impact of security breaches hence why there is concerted effort to improve the detection capabilities of IoCs in order to both contain the breach and reduce the overall impact of the threat.

Reference [155] performed a survey of threat intelligence sharing and categorised IoCs into atomic, computed and behavioural. For example, the atomic provides a specific characteristic of the IOC, such as IP address, while the computed contains information derived from a group of IoCs such as the hash keys. The behavioural is derived from both the atomic and the

computed IOCs, providing the full description of the threat. Reference [156] carried out a survey on threat intelligence on the more sophisticated emerging threats and provided an extensive overview of the available literature on IOCs.

In [157], the author proposed improving IoC detection using the Graph Convolutional Network approach. In [158], a machine learning framework for cyber threats attribution was proposed. The framework was designed to build a profile of the attacker based on the patterns of attacks. The authors' used high-level indicators of compromise to attribute the attacks to cybercriminals. Reference [159] presented a method of analysing threats based on the IoC used for the cyber attacks. Reference [160] proposed IoC-Miner, a framework for extracting threat intelligence's based IoC sourced from public information-sharing platforms.

Reference [161] discussed tools such as the Open Indicator of Compromise (OpenIOC), which provides a standard format for describing IoCs. Other threats sharing mechanisms, including STIX and VERIS, have enabled security professionals to share IoCs with trusted stakeholders in a more standardised format. Reference [162] proposed a system for detecting IoC and generating the IoC descriptions using OpenIOC data sharing format. Their classifier was trained on a dataset containing 150 IoCs and 300 non-IOC using Support Vector Machine. in another article, the authors' [163] proposed an approach for collecting IOC from web pages. The setup also includes security onion virtual machines.

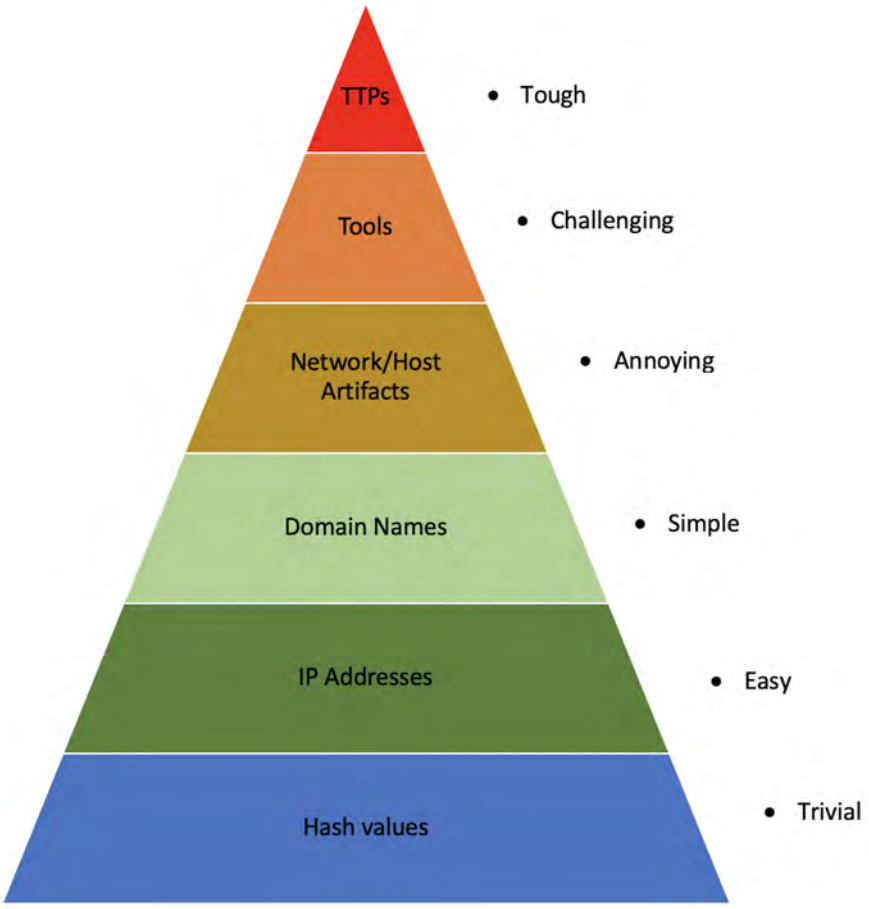

Figure 2.4: Pyramid-of-Pain [164]

Several other authors have researched IoC behaviour about threat hunting, with the most notable being the Pyramid of Pain (PoP) proposed by Bianco [164] as shown in Fig. 2.4. In the Pyramid of Pain, the author categorised the IoCs according to their difficulty. Hashes were listed as trivial and easier to detect because each file or malware has a unique hash key, making it easier to recognise given there are minimal chances of hash collision. On the other hand, hashes change even with the slightest modification to the file

and malware creators often use that to avoid detection. Even though new hashes are generated regularly, they can easily be stopped once the hashes are recognised. The IP addresses were classed as easy in the PoP framework, given that they can be easily attributed to the attacker. However, cybercriminals use proxy servers and Virtual Private Networks (VPNs) to hide their IP addresses, making cyber incident investigation work very difficult.

Next, the domain names were classed as simple in the PoP framework because they are easy to track since domain names have to be registered. However, attackers use Fast-flux DNS, which is associating many IP addresses to a single domain to hide or protect the primary server. Cyber attackers have also been using Fast-flux to mask their botnets. Network/Host artefacts were classed as annoying in the PoP framework. The IoCs artefacts relating to networks or host are not easy to detect due to the sophistication of the attackers' tools and the amount of traffic generated by these systems, making it difficult to distinguish between genuine attacks and benign activities. The Tactics, Techniques and Procedures (TTPs) were classed as tough in the PoP framework and sat on the Pyramid top. TTPs is where the pattern of activities relating to the threat actors is analysed. These threat groups are sometimes well funded and can change their tactics and techniques at short notice if detected, including creating a new payload to exploit their target.

2.9.1 Summary

Researchers carried out lots of work on IoC, but despite these efforts, correctly identifying IoC and particularly during the early stages, is still a significant challenge. The detection challenges are due to many factors, including

the attackers' motivation, the tools they use, and the sophistication of the malware. APT groups are challenging to detect due to the abundance of resources they have to create a crafted malware, and according to a recent report, it takes on average 56 days to detect APT attacks. However, by then, these APT groups may have achieved their goal, including data exfiltration.

There has been improvement in sharing detected IoCs over the last few years due to the availability of threat sharing platforms and acknowledgement of the benefits of sharing IoCs by the security community.

Chapter 7 of this thesis leveraged machine learning-based threat detection to detect attacks using Indicators of Compromise (IoC). Windows event logs and other features such as those of account activities were used to classify the attack types.

2.10 Security Metrics

This section contains a review of security metrics, and some of the content derived from the authors' published work [150, 165].

The advances in technology experienced over the last few decades have resulted in the manufacture of small devices capable of connecting to the internet coupled with processing power; however, this has not been matched by a considerable solution to support the resulting infrastructure.

Researchers and businesses have turned to security metrics to gain insight information on their systems and use that information to secure them. According National Institute of Standards and Technology (NIST), security metrics are tools that are designed to help with decision-making and improve

performance, and accountability [166]

Security metrics has been well researched over the last few years. However, there is no consensus on what metrics to use, and most of the current work relates to organisational metrics and are compliance-based.

2.10.1 Security Metrics and its role in decision-making

Security metrics are means of measurement that can be used to demonstrate the security level of an organisation. Metrics provide insight information and helps decision-makers to make informed decisions. A report by Thycotic [167] highlighted the failure of organisations to implement cybersecurity metrics resulting in a failure to review the performance of their security measures. According to another report by McAfee lab, [168] most organisations realise how good their security defences are once they are breached.

In another report by the US cybersecurity research and development strategic plan [169], there is a need to develop quantifiable security metrics that can be used to measure attackers effort and capability of the defensive mechanism to withstand the attack. Several other authors have described cybersecurity metrics quantification to be a major challenge and often listed it as a major a hard problems to solve [170], [171], [165], [172]. In [171], the authors performed a survey on security metrics and proposed some metrics based on the attack-defensive interactions. Such metrics include: (i) System vulnerability (ii) Defensive strength, (iii) Attack severity, (iv) Situation understanding.

Although this is a developing area, one of the challenges that are hampering the implementation of security metric is the lack of an agreed mechanism

for aggregating security metrics that can assess the overall security posture of all the systems in the organisation [173]. In [174] the authors developed a set of metrics for attack and defence interaction and evaluated the effectiveness of Moving target techniques. The techniques they implemented were network topology shuffle and software diversity. Several other authors discussed moving targets, including the effectiveness of MTD techniques [175, 176]. In [177] the authors created ontology-based security metrics and attack goals determined by several interconnected features. Their ontology has top classes of metrics such as attack metrics, topological metrics, event metrics, attacker metrics and response metrics.

2.10.2 Security Measurement Methodologies

There are several security measurement methodologies and these include:

- The Goal-Question-Metrics (GQM) paradigm is a top-down methodology used to design security metrics programs and provides structured methods for metrics according to the organisations' requirements. GQM is based on three levels: Conceptional level (goal), Operational level (questions) and Quantitative level (Metric) [178]. The GQM + strategies extended the GQM and provide a link between the business goals and measurement programs to help with decision-making [179]. In the GQM method, the organisation defines the goal they want to achieve, followed by a question for each goal, and finally, the metrics required to measure each question.

- Attack graph - metrics based on attack graphs are mainly used to find

the number of paths an attacker could take to exploit vulnerabilities on the system and are qualitative, relying on experts' subjective judgments. [180].

- The Common Vulnerability Scoring (CVSS) framework assigns a numeric value to a vulnerability according to the severity level ranging from one to ten, with one being the lowest and ten the highest. The CVSS has three metric groups, which are base, temporal and environmental groups. The basic metrics contain the characteristic of the vulnerability that does not change over time and consists of exploitability and impact metrics. The temporal metric group contains characteristics of a vulnerability and changes over the lifetime of the vulnerability. The Environmental metric group represents the characteristics of a vulnerability that relate to a particular system, network, or user environment [181].

  Although the CVSS has metrics such as exploitability and impact, which can be found within the base score and environmental metrics, it has its own limitations. The vulnerabilities on CVSS are assigned to individual vulnerabilities and not aggregated, which can be a challenge for organisations and security researchers. Another limitation of CVSS is its inability to consider the link between the vulnerabilities. Attackers often exploit multiple vulnerabilities in order to achieve their ultimate goal.

Most of the current security measures are reactive and rely on metrics such as Mean Time To Detect (MTTD) and Mean Time To Respond (MTTR)

following a security breach. The proactive approaches are missing from such metrics. Hence, there is a need for security measures that can forecast attacks and help decision-makers implement defensive security mechanisms that could preempt the attacks before the attackers exploit them.

2.10.3 Categories of Security Metrics

Researcher in academia and the industry have been working on developing various security metrics. The Centre for internet security (CIS) classified metrics into Management, Operations and technical [182]. The NIST grouped the metrics according to their implementations, effectiveness/efficiency and impact [183]. The implementation metrics track the progress of implementing the security program, including individual security controls and the policies and procedures that underpin the program. The effectiveness/efficiency metrics track the performance of security controls in terms of their defensive capabilities and how well they protect the critical assets. The impact metrics covers the impact the security program had on the overall security inline with the expectation of the business. Typically the impact is measured in terms of risk reduction and risk avoidance achieved as a result of implementing the security program.

2.10.4 Security Metrics Types

There are several types of security metrics, and these include the following:

- Network-based Metrics

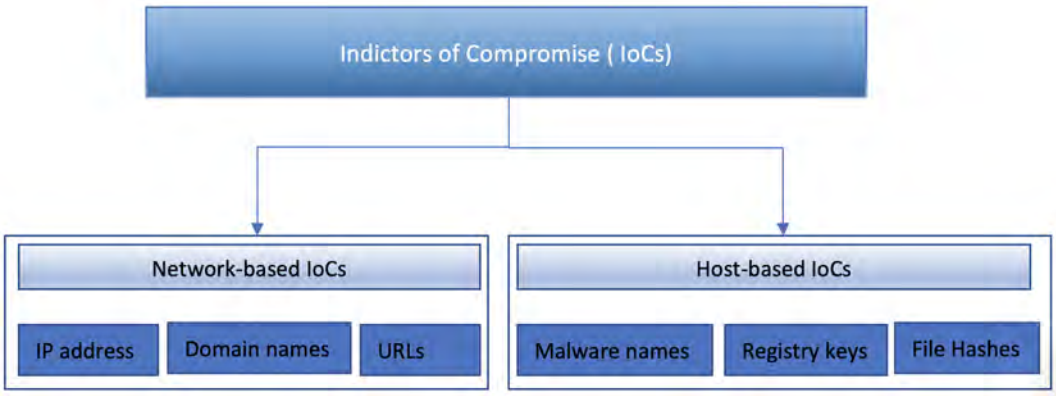

Figure 2.5: Network and Host-based IoC Categories

These metrics are driven by network monitoring tools such as intrusion detection systems (IDS), firewalls and intrusion prevention system (IPS). Fig. 2.6. shows the categories and network and host-based metrics.

- Host-based Metrics

  Host-based metrics are driven by security logs and monitoring that detect activities at the host level. These tools include antivirus and Host Intrusion Detection Systems (HIDS). The HIDS provides visibility on the activities taking place in the system, which adds an extra layer of security [184].

- Risk-based Metrics

  The reliance on compliance metrics which are tick-box exercises, is not enough due to the evolving threat landscape, and organisations are instead embracing risk-based approaches to deal with the impact of emerging threats. Risk-based metrics can give direction on the level of

risk and its trajectory. These metrics can quantify the overall impact of the threat on the business, including deviation from the strategic goals. These metrics can be used as an early warning to help decision-makers take corrective measures to mitigate the risk.

- Vulnerability Metrics

  Vulnerability is a weakness in a system that a threat can exploit [185]. Vulnerabilities pose a serious threat to systems security, although this depends on their severity and impact. Most organisations invest in vulnerability scanning tools to detect existing vulnerabilities. While it is a good security measure, the task of remediation and how quickly these are implemented relies on the system administrators. The cyber attackers can take advantage of these vulnerabilities and exploit the systems if prompt actions are not taken to resolve the issues. The existence of vulnerabilities often provides an indication of the security level of systems, and the time taken to mitigate these vulnerabilities can matter depending on the severity and criticality of the assets to business operations.

  Several vulnerability metrics are used to provide insightful information to decision-makers and system administrators. The most widely used metric is the Common Vulnerability Scoring Systems (CVSS). The vulnerabilities are scored from the lowest to the highest ranging from 1 to 10. The list of vulnerabilities and their assigned score can be found in the National Vulnerability Database (NVD). The CVSS uses three metric categories, which are Base Score, Temporal Score and Environ-

mental Score. These metrics provide an aggregation of attributes such as exploitability, impact and exploit code maturity.

Some of the other vulnerability metrics that can be used to measure the performance of the security measures include Mean Time To Detection (MTTD), Mean Time Time Compromise (MTTC), Mean Time To Patch (MTTP) and Exposure time. MTTD cover the time taken by the security monitoring tools or analysts to detect the vulnerability. MTTC is the time taken by an attacker or penetration tester to break into the system by bypassing the security defences. MTTP is the time taken to patch a vulnerability. Exposure time metric is the duration between detection and application of the patch and could be used to measure how quickly the system administrators are applying patches.

Several other metrics groups can be used to measure the security of systems and other functions, as demonstrated in work by [150]. These include:

- Indicators of Compromise (IoC) Metrics

  Indicators of compromise are artefacts left behind by attackers that can be used to identify malicious activities that have taken place on the target system [163]. Sophisticated attacks such as Advanced Persistent Threats (APTs) take a long time to be detected, and data breaches might have occurred by the time they are detected. IoCs are mainly categorised into Network-based, Host-based and Email indicators and can be used to piece together the activities of the attack and to build the digital trail. Once IoCs are detected, they can be shared with the security community to mitigate future attacks. The sharing platforms

include STIX and OpenIOC [161]

- Indicators of Attacks (IoA) Metrics

  Indicator Of Attack (IoA) is a proactive measure that can be used to reveal an attack that is in progress before the indicators of compromise become visible [186]. Security professionals can leverage IOAs to disrupt attackers before they exploit the systems by implementing mitigating security measures. For example, IoAs could be used to prevent attacks such as Phishing and ransomware, which has become a popular attack vector deployed by cybercriminals. Utilising IoA metrics such as lateral movements and many failed authentication attempts could help organisations detect attacks taking place and implement proactive security measures.

  Searching for the indicators of attacks requires both detective and preventative measure for dealing with cyber attacks. Examples of IOA include internal hosts communicating with known malicious destination and network scan originating from internal hosts.

- Risk Assessment metrics

  Risk assessment is an essential task in every organisation's risk management strategy due to the evolving threat landscape and the increasing attack surface. Most organisations rely on interconnected devices to perform their daily tasks, but these devices introduce vulnerabilities that cyber attackers could exploit. The purpose of risk assessment is to inform decision-makers and support risk responses by identifying the threats facing the organisations, vulnerabilities, impact and likelihood

of the threat causing harm [187].

There is a need to understand what needs to be protected, which can be achieved through the risk assessment process. The first step in the risk assessment process is to perform asset identification and classification in order to apply the appropriate security measures to critical assets. A risk register is created as part of the risk management process and contains the risk description, risk owner and actions taken, among other information. Organisations with the cyber risk management process often reduce data breaches by identifying the risk they face through their risk assessment and implement corrective measures. They are also more prepared to deal with eventual security breaches should they occur if they have already identified their critical assets and processes.

Risk assessment is a continuous process due to the ever-changing threat landscape. Risk metrics can play an important role in ensuring cyber risks are prioritised and adequate countermeasures are assigned to protect critical assets. Risk assessment metrics that could be utilised include:

(1) Percentage of risk with critical or severe rating. This metric will help prioritise and direct resources to deal with the most urgent threats affecting the critical assets.

(2) Percentage of assets that are not monitored. This metric will help with monitoring the coverage and determine existing gaps.

(3) Insider threats metrics. This metric includes accidental and malicious incidents attributed to insider threats.

- Penetration testing metrics

  Penetration testing involves finding vulnerabilities on systems and attempting to exploit the weaknesses [188]. Penetration testing can be costly due to the costs involved in hiring skilled workers and the chances of unintentional breakdown or data breaches during the exploitation stage when penetration testers are trying to compromise the systems. Due to these challenges, organisations tend to perform penetration testing on a quarterly or bi-annual basis. However, the threat landscape could change very quickly, making such tests redundant within a short period.

  To fill this gap, newer penetration testing tools that operate inside the network have been developed by companies. For example, Firedrill developed the AttackIQ [189] which is a tool that performs automated testing to determine the enterprise security posture, but these tools are not widely adopted. One way to maximise these tests is to use penetration testing metrics. Such metrics include but not limited to:

  (1) Percentage of penetration tests that discovered high risks. This metric could be used to measure the performance of the existing security controls and their detection capability.

  (2) Penetration testing intervals. New threats can arise between the various penetration testing intervals. Shorter testing intervals are preferable in order to increase the chances of detecting new threats following the previous test.

  (3) Mean Time To Fix (MTTF). These metrics will show the average

time taken to fix the vulnerabilities identified during the penetration testing and allow senior managers to measure the capability and average response times of the technical teams.

- Red and blue teaming metrics

  Red teaming is a simulated form of attack in which skilled teams attempt to penetrate the security defences and compromise the systems. Organisations usually employ the service of red teams in order to test the maturity of their security controls [190]. After the red team assessment, organisations will have a list of attack vectors they are vulnerable to and corrective measures to mitigate such risks.

  Red teams should be complemented with Blue teams whose role is to defend against attacks and bolster the security defences. The purpose of a Blue team is to defend the organisation against both Red teams and real attackers. Enlisting the service of Red teams can be expensive, and organisations should try to maximise this potential to improve their security. Red and Blue teams could also be utilised during the optimisation stages after new security programs are deployed. Organisations can use metrics to measure how well they are integrating the outcome from the Red team assessment to improve the strength of their security mechanisms. Such metrics include the skills and knowledge of the attackers.

- Resilience Metrics

  Resilience is the ability for a system to adapt and continue to provide functionality in the face of an attack [191]. One of the most prevalent

attacks deployed by cyber attackers is ransomware which encrypts the files and makes them unavailable to the users until a ransom is paid. Resilience will enable organisations to withstand adversarial attacks and ensure continuity of critical services [192]. The following are some of the metrics that could be used to measure resilience in an organisation. These metric include Mean Time to Failure (MTTF), Mean Time to Repair (MTTR and a availability of offline and tested backup

- Threat intelligence metrics

  Cyber threats are constantly changing, and attackers employ sophisticated tools and techniques to bypass security defences. These cyber attackers are known to be sharing techniques using the dark web to mask their identities and digital trails, which reduces the effectiveness of the traditional security mechanisms. Threat intelligence sharing plays an important in defeating such attacks and implementing mitigating controls.

  Threat intelligence is about finding information relating to the attackers and their techniques. Such information can include the attackers tactic, techniques and procedures (TTPs), motivation, and targets. Threat feeds are becoming popular, and organisations have realised the importance of threat sharing and leveraging the expertise of the security community. An organisation can use security metrics to measure their threat intelligence capability and determine how well they are prepared in the event of an attack. Such metrics include:

  (1) Number of known threats groups targeting your organisation or

sector at any given time. This metric could be used to measure how well you are capable of dealing with threats from these communities. Information about these groups could be obtained from in-house threat intelligence teams or vendors and the security community.

(2) Access to vendor threat intelligence report directly related to your organisation. Vendors have huge capabilities and resources, including threats sharing with their industry partners. Having access to these threat intelligence feeds will provide your organisation with an edge over the attackers.

(3) Metric from Indicators of Compromise (IoC). The IoCs can be used to identify or attribute to particular attack types.

2.11 Summary

In this chapter, we discussed security metrics and their role in helping to secure organisations against potential cyber attacks by measuring the effectiveness of the security measures. The ultimate goal of security metrics is to support decision-making and to provide accountability. There are no sets of agreed security metrics, although several bodies such as CIS have proposed their own metrics, and organisation can implements metrics that suit their goal and business objectives. Metrics can be obtained from sources such as vulnerability scanners, intrusion detection/prevention systems, and other network monitoring systems such as Antiviruses, Firewalls and SIEMS. Although metrics from these systems are valuable, they have their limitations, such as false alarms, which can impact organisations in terms of threats that

may have been misclassified and the large number of resources required to triage these alerts. For example, an antivirus system might show metrics such as the total number of malware detected and in a given month but does not show the number of malware not detected in that same period. Some of the malware not detected by the antivirus system can be obtained through forensic audits, and IoCs left behind following an attack or malware infection reported by end-users.

This thesis uses performance metrics such as detection accuracy and false alarm rate, which can help decision-makers evaluate their security systems' effectiveness. Machine learning models are utilised to forecast cyber events, detect the stages of cyber attacks such as APTs and perform reactive approaches based on indicators of compromise, which are forensic artefacts left behind following an attack.

Next, some of the key cybersecurity frameworks will be discussed and summarised to show how they relate to our work.

Chapter 3

Cybersecurity Frameworks

Cyber attacks and associated risks have been a concern faced by many organisations. Several frameworks and industry best practices are widely adopted to help organisations secure their critical assets and be prepared for eventual security breaches through better resilience and recovery.

The most common frameworks for security assurance are (1) Centre for Internet Security (CIS 18), (2) ISO 27001, (3) ISO 27004, (4) Cyber Essentials, (5) NIST Cyber Security Framework, (6) NCSC Cyber Assessment Framework.

3.1 Centre for Internet Security (CIS 18)

These are the top 18 control sets proposed by the Centre for Internet Security (CIS). They are operational controls that provide system administrators with a means to secure their systems and protect against common cyber attacks [193]. The CIS 18 controls are more prioritised and actionable controls that

are easier to implement than the more comprehensive frameworks such as ISO 27001 and NIST Cybersecurity framework. These controls range from inventory and control of enterprise assets to penetration testing.

CIS Control No.	**CIS Top 18**
1	Inventory and Control of Enterprise Assets
2	Inventory and Control of Software Assets
3	Data Protection
4	Secure Configuration for Enterprise Assets and Software
5	Account Management
6	Access Control Management
7	Continuous Vulnerability Management
8	Audit Log Management
9	Email Web Browser and Protections
10	Malware Defenses
11	Data Recovery
12	Network Infrastructure Management
13	Network Monitoring and Defense
14	Security Awareness and Skills Training
15	Service Provider Management
16	Application Software Security
17	Incident Response Management
18	Penetration Testing

Table 3.1: CIS18 controls [193]

3.2 ISO 27001/2

The ISO 27001 is an Information Security Management System (ISMS) framework whose objective is to apply appropriate countermeasures to protect critical assets and reduce the threats' impact. The purpose of ISMS is to protect the confidentiality, integrity and availability of information assets. The European Union Agency for Cybersecurity (ENISA) [194] described six steps involved in the development of ISMS. These include (1) Definition of security policy, (2) Definition of ISMS scope, (3) Risk Assessment, (4) Risk Management, (5) Control selection, (6) Statement of Applicability (SOA).

ISO 27001 is a widely adopted industry best practice for information security and allows organisations to demonstrate their commitment to continuous protection of the information they hold through regular audits. There is a revalidation audit to ensure organisations are continuously complying with the requirements, and any non-conformance is highlighted and must be corrected before the certifications are reissued. The uptake of ISO 27001 is higher in larger organisations due to the need to protect the vast amount of personal data they hold or process, demonstrate information security best practice and comply with legal and regulatory requirements.

The ISO 27002 provides guidance on applying the controls and helps organisations select the appropriate controls during the implementation of 27001. The ISO 27002 provides more detailed information about the controls sets and allows users to select the controls that mitigate the risks based on the risk assessment results. Although both the ISO 27001/2 are used together, it is only the ISO 27001 that organisations are certified against.

3.3 ISO 27004

The ISO 27004 is a standard for evaluating the performance of the Information Security Management Systems (ISMS) to fulfil the requirements for ISO 27001. The current version of this standard is ISO 27004:2016 and provides guidance on monitoring and measurements of the ISMS [195].

The foundation of ISO 27004 is ISO 27001, one of the primary standards for managing information security. The ISO 27004:2016 focus on security metrics for measuring the performance of the ISMS. Metrics support decision-making and allow senior executives to gain insight into how the security mechanisms are performing.

This standard provides guidance on how to construct information security programs, including what to measure. The framework provides several examples of security measures and how the effectiveness of these measures could be assessed. ISO 27004 can be used by any organisation regardless of its size.

3.4 Cyber Essentials

Cyber essentials is a simple and basic framework proposed by the National Cyber Security Centre (NCSC). The framework consists of five controls which organisations can use to secure their systems and provide a security baseline [196]. These five controls are (1) Firewall, (2) Patch management, (3) Malware protection, (4)Access control, (5) Secure configuration.

3.4.1 Cyber Essentials Controls

1. **Firewall**

   Boundary firewall and internet gateways protect the organisation and can be effective if appropriately configured. Cyber essentials provide a set of basic checklists which organisations can use to achieve a basic level of protection. A firewall monitors traffic and can stop malicious traffic from entering the network.

2. **Patch Management**

   Patch management is keeping software and systems up to date. Cyber attackers are known to be exploiting vulnerabilities on these systems. Although there are zero days vulnerabilities, most of the security breaches are associated with exploiting known vulnerabilities where a patch is already available.

   Cyber essentials require organisations to have a mechanism for implementing patch management and gives guidance on some of the best practices, which include: Ensuring software is supported and licensed, Removing legacy systems that are no longer supported, Patching high and critical risks within 14 days of a patch release.

3. **Malware Protection**

   The protection of their systems is at the forefront of most organisations. The malware protection mechanism is one of the key investments to make in order to protect these systems from threats such as Ransomware which has become a prevalent tool for cybercriminals in

recent years.

4. **Access Control**

   Access control provides authentication and authorisation. Organisations should have access control to ensure user privileges are assigned appropriately and misuse is prevented. Administrative privileges should be limited and monitored in order to avoid intentional or unintentional risks. Cybercriminals are known to target accounts with administrative rights that help them execute malware and perform privilege escalations and lateral movements.

5. **Secure Configuration**

   Secure configuration allows systems to be more secure and more resilient to cyber risk. Hackers are known to exploit misconfigurations on systems. Standardising the configuration of the systems can also make managing and pushing security patches much more manageable.

   Cyber essentials provide guidance on implementing secure configuration, and these include:

3.5 NIST Cyber Security Framework (NSF)

The NIST cybersecurity framework is a comprehensive framework that guides organisations on identifying, preventing, detecting, and responding to cyber-attacks in a timely manner. This framework references other frameworks such as ISO 27001, CIS, COBIT 5 and NIST SP 800-53 Rev4.

The framework is designed for critical infrastructure but is flexible enough to be implemented in other sectors. It consists of five core functions, and each is divided into categories and sub-categories [197]. The core functions are (1) Identify, (2) Protect, (3) Detect, (4) Respond, (5) Recover.

3.5.1 Identify

The primary goal of the identify function is to determine the inventory of all assets and the risk they face.

The key categories of this function are (1) Asset Management, (2) Business Environment, (3) Governance, (4) Risk Assessment, (5) Risk Management Strategy, (6) Supply Chain management.

3.5.2 Protect

The goal of the protect function is to guide the implementation of security mechanisms to protect critical assets from harm.

The key categories of this function are (1) Identity Management, Authentication and Access control, (2) Awareness and training, (3) Data Security, (4) Information Protection Process and Procedures, (5) Maintenance, (6) Protective Technology.

3.5.3 Detect

The detect function provides guidance on the detection of anomalies and other malicious activities in the network. Detecting anomalies can help capture attacks in their early stages, allowing system administrators to deploy

countermeasures to mitigate the risks.

The key categories of the detect function are (1) Anomalies and Events, (2) Security Continuous Monitoring, (3) Detection Process.

3.5.4 Respond

Despite best efforts, it is possible organisations security defences will be breached at some point. The respond function provides guidance on how to deal with such eventuality should it occur. It is crucial to have effective and tested response plans to be in place to limit the impact of such threats on the organisation.

The key categories of the response function are (1) Response planning, (2) Communications, (3) Analysis, (4) Mitigation, (5) Improvements.

3.5.5 Recover

The recover function provides guidance on how to bring services back to the operational state following a disruption. This function also encompasses disaster recovery and business continuity plans.

The key categories of the recovery function are (i) Recovery Planning, (2) Improvements, (3) Communications.

3.6 NCSC Cyber Assessment Framework (CAF)

NCSC Cyber Assessment Framework (CAF) is a framework that provides an approach of assessing the impact of cyber risk on critical functions and contains four objective and 14 principles [198].

- Objective A: Managing Risks

  This objective deals with managing risks to critical systems and functions. It contains four principles which are governance, risk management, asset management and supply chain.

- Objective B: Protecting against Cyber Attacks This objective deals with the application of security measures to protect critical functions consist of six principles which are service protection policies and procedures, identify and access controls, data security, system security, resilient network and systems and staff awareness training.

- Objective C: Detecting Cyber Security Events This objective deals with cyber threat detection capability and consists of two principles which are security monitoring and proactive security event discovery.

- Objective D: Minimising the impact of cybersecurity incidents This objective deals with the organisations' capability to minimise the impact of security incidents and consists of two principles which are response and recovery planning and lesson learnt

3.7 Summary

Cybersecurity frameworks play an important role in improving the security posture of organisations. Some of these frameworks, such as ISO 27001 and NIST cybersecurity framework, are comprehensive and, if implemented correctly, can reduce the number of security breaches and increase resiliency.

The CIS 18 controls are operational metrics that are much easier to implement compared to the ISO 27001 and the NIST Cybersecurity Frameworks, while Cyber essentials provide five basic sets of controls to protect against common online threats.

Although these frameworks can improve the security posture if implemented correctly, they mainly provide guidelines, and their effectiveness depends on how well they are implemented. Most of the frameworks listed above have some relations to the work carried out in this thesis, but they do not provide a mechanism for measuring security assurance quantitatively and the effectiveness of the security measures without manual intervention and allocation of substantial resources to scope the work. The framework that is closest to work carried out in this thesis is the ISO 27004 standard which provides metrics-based measurements. However, these are manually computed and rely on the human factor, which can introduce errors. This thesis will improve these frameworks by adding automated capture of potential security attacks and issues as part of the algorithm, which will pave the way for automating cybersecurity intervention and proactive detection. In this thesis, machine learning-based techniques were used to identify attacks and their stages correctly. The evaluation metrics used include detection accuracy and false alarm rate, covering both false positive and false negatives.

Chapter 4

Proposed Framework

Enterprises are relying on protective measures to protect their critical systems, but these measures are shown to be less effective at dealing with an advanced and complex cyberattack. Therefore, comprehensive solutions capable of predicting, preventing, detecting and responding to these cyber threats are needed. It is anticipated that this framework will address some of these challenges by introducing predictive, detective and forensic capabilities.

In this chapter, our proposed framework for the security monitoring of networked systems is presented. The main focus is the machine learning aspect of the framework, where a rigorous analysis of the results occurs. The performance of the key components is evaluated. Such components include the cyber attack detection and prediction using the Cyber Kill Chain approach to detect the stages of advanced persistent threats, time series forecasting for attack prediction, and indicators of compromise to collect forensic artefacts. The frameworks components will be discussed in the following sections.

4.1 Framework Blocks

The framework consists of five blocks that summarise the key activities undertaken in this research, as shown in Fig. 4.1. The first block consists of the network systems where data is collected using probes. Overall, there were four major experiments in the research. The first and second experiments utilised a publicly open IDS dataset [5, 34] where the dataset providers ran probes and provided a machine learning dataset with the various features. The labs are setup in the third and fourth experiments, and simulated attacks are performed to collect the datasets involving Indicators of Compromise(IoC). The second block contains the primary datasets, which are APT [5], CSE-CIC2018 [34] and the IoC datasets. The exploited vulnerabilities used during the datasets' creation are checked to confirm the exploits and tools used in these experiments. Log data was collected from these hosts and downloaded as part of the CIC2018 and IoC datasets. The logs provide essential information such as the timestamps and the events that took place on these systems.

The third block contains the network parameters, host parameters and IoC variables derived from the dataset. These parameters are primarily features relating to networks and hosts. The network features include flow data such as packet lengths, while host features include events ids. The IoC artefacts can be used to identify the presence of threats on the affected systems. Abstraction, optimisation and aggregation were performed on these parameters and artefacts to fine-tune the features before feeding them to the machine learning model for classification.

The central part of the framework is the fourth block related to machine

learning. In this block, the prepared and constructed datasets are fed into the machine learning model, where classification is performed. Feature engineering is also performed here to select the best features to maximise the efficiency of the model. It is a continuous process where the parameters are optimised until optimal results are achieved. This research used four datasets: APT, CSE-CIC2018 and two IoC datasets to create our machine models. The APT dataset captures realistic stages of an APT attack that is highly sophisticated and challenging to detect. Dataset reconstruction was performed, and the Cyber Kill Chain was then applied to detect the various stages of the attack. The work of the APT dataset provider [5] was used as a baseline to expand on the machine learning aspect of their work, as shown in Fig. 4.1. The machine learning algorithms used for classification are SVM, Random Forest, BayesNet, Naive Bayes and KNN. The APT stage detection will be covered in chapter 5. The second dataset, the CSE-CIC2108, is a realistic cyber defence dataset with various attack labels performed on a secure network. The dataset used for attack prediction and time series forecasting is covered in more detail in chapter 6. The final dataset is the IoC datasets created from the networks set up during our experiments and discussed in chapter 7. The IoC datasets contain the artefacts comprising Windows events and other features to piece together attacks on this system.

4.2 Framework Modules

Fig. 4.2 shows a high-level overview of our machine learning framework's components taking place in this part of the framework (block 4). The three

models that were created will contribute to the networked system's effective security monitoring and allow the system administrators to secure their networks. The three components shown in Fig. 4.2 are (i) attack stage detection, where a model to detect advanced persistent threat was created; (ii) attack prediction, where a model for predicting cyber events using time-series forecasted features was created, (iii) the IoC detection detection, where a model for detecting IoCs was created. System administrators can feed these detected IoCs to their monitoring systems such as Security Information and Event Management Systems (SIEMS) and Intrusion Detection/Prevention Systems(IDS/IPS) to correctly detect them in the future and alert the system administrators or take preventative actions. System administrators can build a module for each of these three components shown in Fig. 4.2 and deploy it on the network to enhance their security defences and determine their network monitoring tools' accuracy.

The fifth and final block of the framework is the visualisation charts. This charts display the model's performance result using metrics such as accuracy, precision, recall, and the false alarm rate. This visualisation chart will allow decision-makers to have a high-level overview of the security status and determine how well their security controls performed by linking it to the detection accuracy and then performing proactive measures to prevent attacks.

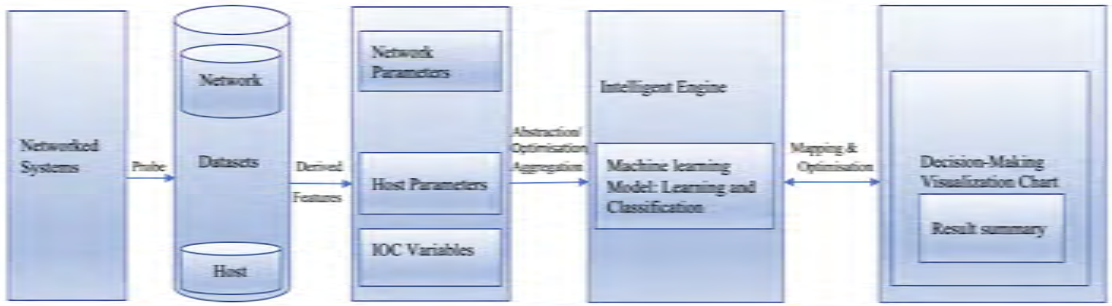

Figure 4.1: Framework for Security Monitoring of Networked Systems

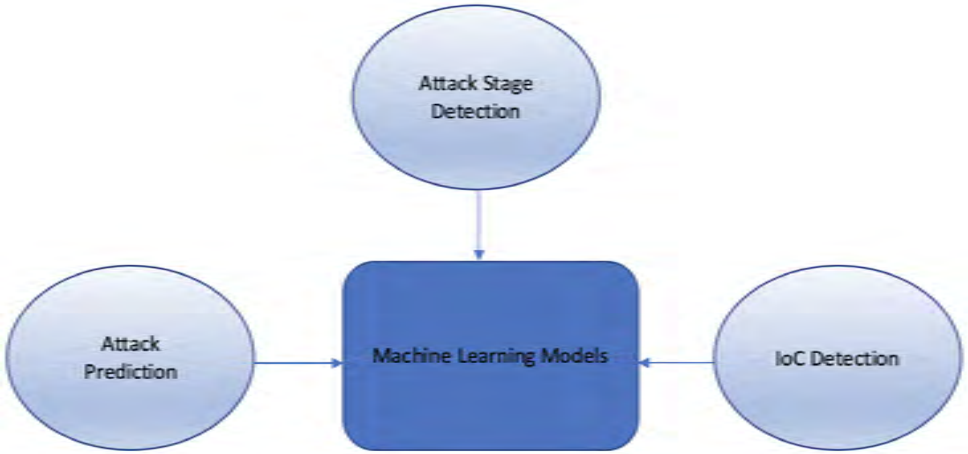

Figure 4.2: Framework Component Modules

4.3 Summary

In this chapter, the definitions of functional blocks in the framework was given. Machine learning was leveraged to detect the stages of sophisticated attacks such as APT. Time series forecasting was applied to predict cyber-attacks before they occur by looking at the pattern of cyber events, which should give system administrators time to perform proactive approaches to mitigate the risk that may be realised from incoming attacks. Finally, IoC artefacts from hosts were used to build a picture of attacks that the security monitoring systems may have missed. Such artefacts can lead to these attacks being detected and reduce their adverse effects by allowing system administrators to act promptly. It is anticipated that the combination of attack stages detection, cyber-attack event forecasting, prediction, and IoC detection proposed in this framework will reduce cyber breaches by allowing the system administrators to implement corrective and preventative measures to mitigate these risks.

Chapter 5

Machine learning for APT attack detection

In this chapter, the proposed cyber kill chain approach for detecting advanced persistent threats will be presented. most of the content of this chapter was derived from the author's work, which was published in a peer-reviewed journal [32].

While several predecessor works have investigated machine learning for APT detection and mitigation, there have been various shortcomings in their effectiveness for wider uses. These include: (i) a lack of reliable publicly-open APT datasets, (ii) a lack of alignment with Industry-informed practice on the available dataset construction, (iii) limited experimental works to evaluate the learning algorithm effectiveness. This work proposes to advance the machine learning application for APT detection by addressing the latter two shortcomings from the previous works. Our main contributions include:

- Building upon a recently proposed APT dataset in [5], The industry-

informed framework of Cyber Kill Chain was leveraged to reconstruct a dataset that captures realistic APT stages. Data intelligence was employed via machine learning that exploits possible patterns within the reconstructed dataset.

- Feature extraction was performed via Multiple Factor Analysis (MFA) to overcome the limitations in the number of features in the APT dataset.

- Feature selection techniques were utilised to select optimal features for the APT stage detection and classification.

- An in-depth and rigorous analysis of the experimental results was performed to assess the trade-off of the classifiers' performance using a variety of performance metrics.

This experiment improved on the work [5] by leveraging the Cyber Kill Chain approach for dataset reconstruction to enhance the attack stages and associated alert types that are critical to accurate detection of APT stages. The alert grouping was refined such that one alert could correspond to multiple attack stages in order to remove direct linkage between stages and alert types as found in [5].

Despite recent progress on APT research, there is limited experimental evidence of explicit association and linkage between existing APT datasets and the corresponding machine learning with the Cyber Kill Chain data modelling. This work attempts to address this gap by reconstructing a recently shared APT dataset. However, the original dataset has its own limitation, including the limited number of features. Machine learning was used to gain

insight into the data and to perform rigorous analysis of the experiments. This work can provide a foundation for future provisioning of automated APT detection and classification with minimised human intervention.

5.1 Cyber Kill Chain Informed Modelling

The Cyber Kill Chain (CKC) was developed by Lockheed Martin's and consisted of seven stages covering the whole attack life cycle, as discussed in section 2. In this section, the dataset will be reconstructed and map the detection alerts to the CKC stages. We then perform feature extraction and feature selection to improve the detection of the stage detection model. Fig. 5.4 shows the data preparation stages.

Fig. 5.1 depicts the APT stages proposed by the original dataset providers [5] alongside our work based on the Cyber Kill Chain. This work builds on their work as a baseline to expand on the machine learning aspect of their framework with the aim of improving the overall detection accuracy. The methodology proposed by these authors covers six stages, but they only considered four of these as detectable stages. These stages are (i) Point of entry, (ii) C&C Communication, (iii) Asset/data Discovery, (iv) Data exfiltration [5]. Their proposed detection modules are Disguised exe File Detection (DeFD), Malicious File Hash Detection (MFHD), Malicious Domain Name Detection (MDND), Malicious IP Address Detection (MIPD), Malicious SSL Certificate Detection (MSSLD), Malicious Flux Detection (MDFD), Scanning Detection (SD) and Tor Connection Detection (TorCD) [5] as shown in Fig 5.3.

According to the original dataset providers [5], the Disguised exe File Detection (DeFD) module is designed to detect files whose extension have been manipulated to avoid detection. Malicious File Hash Detection (MFHD) is designed to detect malicious files download on the systems based on a blacklist of malicious hashes. Malicious Domain Name Detection (MDND) is designed to filter malicious domains based on blacklisted domains. Similarly, the Malicious IP Address Detection (MIPD) module detects malicious IP addresses and is based on blacklisted IP addresses. The Malicious SSL Certificate Detection (MSSLD) module aims to detect malicious SSL certificates based on blacklisted SSL certificates. The Scanning Detection (SD) module aims to detect port scanning. The Tor Connection Detection (TorCD) module aims to detect connections to networks by looking at the source and destination of the traffic and referencing against a list of known Tor servers.

Fig. 5.3 depicts that their stages and detection modules fall within the delivery, command & control, and action on objectives stages of the CKC. Their point of entry stage, which corresponds to the delivery stage of CKC, is broad and can be matched to the other stages. However, in our opinion, it is more suitable for the delivery stage of the CKC, given that this was considered the initial point of compromise. In the next part, each of the seven CKC stages and their assigned detection modules will be discussed, including those proposed by the APT dataset providers and others from our proposed work.

5.1.1 Reconnaissance Detection

In the reconnaissance stage, the attackers gather information about the target. Although there were no proposed detection modules for reconnaissance in the work by the original dataset providers, given the importance of this stage, it is crucial to have a detection module to prevent such attacks or detect them in the early stages.

Therefore this work proposes detection modules which can be used detect reconnaissance attacks. These detection modules will be built in our future work. Our proposed detection methods for this stage include: (i) OS fingerprinting [199], (ii) Port scanning [199], (iii) Alerts on robot.txt access which can reveal restricted paths [200], (iv) DNS enumeration [201], (v) DNS honey tokens [202].

- OS fingerprinting

  Attackers can perform OS fingerprinting to find more about the target system and create payloads designed to target vulnerabilities on that particular operating system. Likewise, system administrators can perform OS fingerprinting to determine the security of their networks.

  The OS detection module will monitor traffic and logs to detect the scan and notify the system administrators, who can then use this information to harden their system. Source IP addressed will also be checked against a blacklist of IP addresses.

- Port scanning

  Scanning is a common task performed by cyber attackers during the

reconnaissance stage. Some of the most common tools used include NMAP for scanning open ports and services running on them.

The port scanning detection module will monitor traffic to detect malicious port scanning by looking at triggers such as patterns of the scans and malicious source IP address.

- Alerts on robot.txt

  The robot.txt file can reveal restricted information, including disallowed directories, if it is not configured correctly. This detection module will monitor the contents of robot.txt files and files accessible to the crawlers.

- DNS enumeration

  A typical DNS server contains a list of all computers, IP addresses and services running on them. This helps the attackers build a map of your network, which they can use in the attack. The DNS enumeration detection module will monitor triggers such as suspicious requests from non-DNS servers and sources with excessive requests. It will also notify systems administrators when suspicious activities are detected .

- Honey tokens

  Honey tokens can be accounts or other resources designed to trap cybercriminals and shadow their activities. The proposed honey tokens detection module will monitor these activities, and any attempt to access the accounts or resources will trigger an alert which will also be sent to the systems administrators

5.1.2 Weaponization Detection

During this stage, the attackers use the information that was obtained during the reconnaissance stage to create a carefully crafted malicious payload. The attackers usually use automated tools for packaging their malware. Remote Access Trojan (RAT) and exploits are used during the weaponisation.

The original APT dataset provider's work did not create a detection module for this stage in their framework, given that the attackers will not be interacting with the target at this stage. We agree with the dataset providers and have not assigned any alert to this stage in our CKC informed model.

5.1.3 Delivery Detection

Malicious actors deploy a weaponised payload to the target during the delivery stage. There are multiple means for payload delivery available to the attackers, including malicious emails, click-by downloads, watering hole [203], or infected USB devices [204]. The authors of the original APT dataset called it the point of entry in their proposed APT lifecycle. They used the detection methods in Fig. 5.3 to detect their APT steps. However, this was expanded further in our work, considering some of the sophisticated APT attacks, such as Stuxnet, were delivered using infected USB sticks [204]. Infected USB drives, malicious links [205], and injection attacks [206] were added to the list of alerts in our proposed detection methods and are planning to build the detection modules in our future work.

- Disguised exe file

  This module detects files whose extension have been manipulated to

avoid detection.

- Malicious file hash

  This module detects malicious files that have been download on the system based on a blacklist of malicious hashes. Hashes are calculated for each new files, and their MD5, SHA1 and SHA256 keys are crossed checked against the blacklist.

- Malicious domain name

  This module checks for connections to malicious domains based on a blacklist of known malicious Domain names. DNS requests are filtered and cross-referenced against the blacklist

- Infected USB devices

  USB devices have been used to spread malware between computers, whether connected to the internet or offline. USB devices have been used to spread malware in complex attacks such as Stuxnet. Our proposed detection module for infected USB will trigger alerts if actions such as auto-runs are detected.

- Malicious links

  Malicious URLs are commonly utilised by cybercriminals including in phishing attacks. Our proposed detection modules will check this link against blacklists and other indicators.

- Injection attack

Injection attacks are common in web applications, and APT attackers use methods such as SQL injection and Cross-site scripting to establish a foothold on their target system. The proposed module will monitor malicious activities on web application and logs for triggers of scripting attacks, including inserting unexpected parameters in a user input field to try and expose personal information.

5.1.4 Exploitation Detection

A vulnerability must exist first before a malicious payload can be executed successfully. In work by the APT dataset provider, the authors did not directly specify a detection module and alerts for the exploitation stage. However, their point of entry stage may overlap with this stage. In this stage, two alerts from the original authors and three from our list were added, namely: (i) Brute force detection, (ii) Pass hash detection alerts, (iii) Task schedule, (iv) Scripting, (v) PowerShell [207]. Next, a brief description of the proposed detection modules will be given.

- Task schedule

  Task schedule runs in privileged mode and can be exploited by attackers, as demonstrated in a recent exploit targeting Window 10 task scheduler addressed in CVE-2019-1069[208]. New variants of malware and some APTs are targeting Task scheduler to deploy their programs in privileged mode.

  The proposed detection module will monitor the task scheduler for any new, unexpected or hidden tasks. It will also be configured to reference

auto-start applications to detect application such as malware that was scheduled to auto-run. Other tools such as Auto-run for Windows could also be utilised.

- PowerShell

  Powershell is a Microsoft tool that system administrators widely use. However, it has become a favourite tool for cybercriminals in recent years due to its capability to run in memory without writing to disk and bypassing the security monitoring tools.

  It is also a trusted application which means it can execute scripts without being blocked. Attackers are known to use PowerShell to execute the malware on their target systems and using tools such as invoke-Mimikatz, Powercat and PowerSploit. Attackers are known to use PowerShell to create scripts that automate data exfiltration. The proposed detection module will monitor Powershell activities to detect malicious and unexpected tasks in order to detect attempts to exploit systems through PowerShell.

- Brute force

  In brute force, the attackers try all possibilities and combinations until they find a working one. The main motivation behind brute force attacks is to gain access to restricted information or resources. Our proposed module for brute force detection will monitor log files and look at brute force triggers such as repeated failed account login attempts.

- Pass hash detection

Pass hash is a post-exploitation technique used by cybercriminals to steal credentials and perform lateral movement. Attackers extract hashes using tools such as Mimikatz. This detection module will monitor account activities using Windows events to detect pass hash attacks.

- Scripting

  Scripting attacks are more common in web applications, and APT attackers use methods such as SQL injection and Cross-site scripting to establish a foothold on their target system. The proposed module will monitor malicious activities on web applications and logs for scripting attacks, including inserting unexpected parameters in the user input field to expose personal information.

5.1.5 Installation

Attackers execute the malware during the installation stage of the CKC. In this stage, One alert from the detection methods proposed by the APT dataset provider [5] and a further two from our list were added: privilege escalation and injection attack alerts, as shown in Fig. 5.3.

- Brute force

  Attackers try all possibilities and combinations to achieve their goal. The main motivation behind brute force attacks is to gain access to restricted information or resources. This brute force detection module will monitor log files and look for brute force triggers such as repeated failed account login attempts.

- Privilege escalation

  Privilege escalation allows attackers to gain elevated permissions. Attackers take advantages of vulnerabilities such as misconfigurations to achieve their goal. Our proposed detection module will trigger alerts if activities such as unauthorised access to endpoints or anomalies relating to accounts are detected.

- Injection attacks

  Injection attacks are common in web applications, and APT attackers use methods such as SQL injection and Cross-site scripting to establish a foothold on their target system. The proposed module will monitor malicious activities on web application and logs for triggers of scripting attacks, including inserting unexpected parameters in the user input field to try and expose personal information

5.1.6 Command and Control

In this stage, the detection methods proposed by the original APT dataset provider [5] were used, which are: (i) Malicious IP address, (ii) Malicious SSL certificate, (iii) Malicious domain flux detection, as discussed in section 5.1.

- Malicious IP address

  The malicious IP Address Detection (MIPD) module detects malicious IP addresses and is based on blacklisted IP addresses. Malicious IP addresses are filtered and alerts sent to system administrators.

- Malicious SSL certificate

  The Malicious SSL Certificate Detection(MSSLD) module aims to detect malicious SSL certificates based on blacklisted SSL certificates. The connections are filtered and checked against a list of blacklisted certificates.

- Malicious domain flux detection

  Domain flux is a technique closely associated with botnets that generate a large number of domain names randomly to maintain access and protect their Command and Controls servers [209].

  This detection module will trigger alerts when events such as high volume of DNS queries resolved against a particular IP address.

- Tor Connection

  Tor provides anonymity; the cybercriminals use Tor to hide their activities and to avoid detection. The Tor detection module will detect connections from the Tor network using a list of servers from the Tor network. All traffic will be checked for the source and destination address and compared against the Tor server list.

5.1.7 Action on Objectives

This module refers to the final part of the Cyber Kill Chain. The dataset providers [5] used Tor connection alerts and scanning as their detection methods, and DNS tunnelling detection was added from our list.

- DNS Tunnelling

Attackers encode malicious data in DNS queries and responses and then exploit these for their malicious activities [210]. Our proposed detection module will monitor traffic to look for the pattern in the data and check the frequency of the DNS request.

- Tor Connection

  The Tor detection module will detect connections from Tor network using a list of servers from the Tor network. All traffic will be checked for the source and destination address compared against the Tor server list.

- Scanning

  Cybercriminals use scanning tools to scan for open ports and services during the reconnaissance stage. The same scanning is also performed during the final stages of the attack to find other systems of interest in the internal network and perform lateral movement. This module will track scanning activities based on predefined rules and threshold, and alerts will be sent to the system administrators.

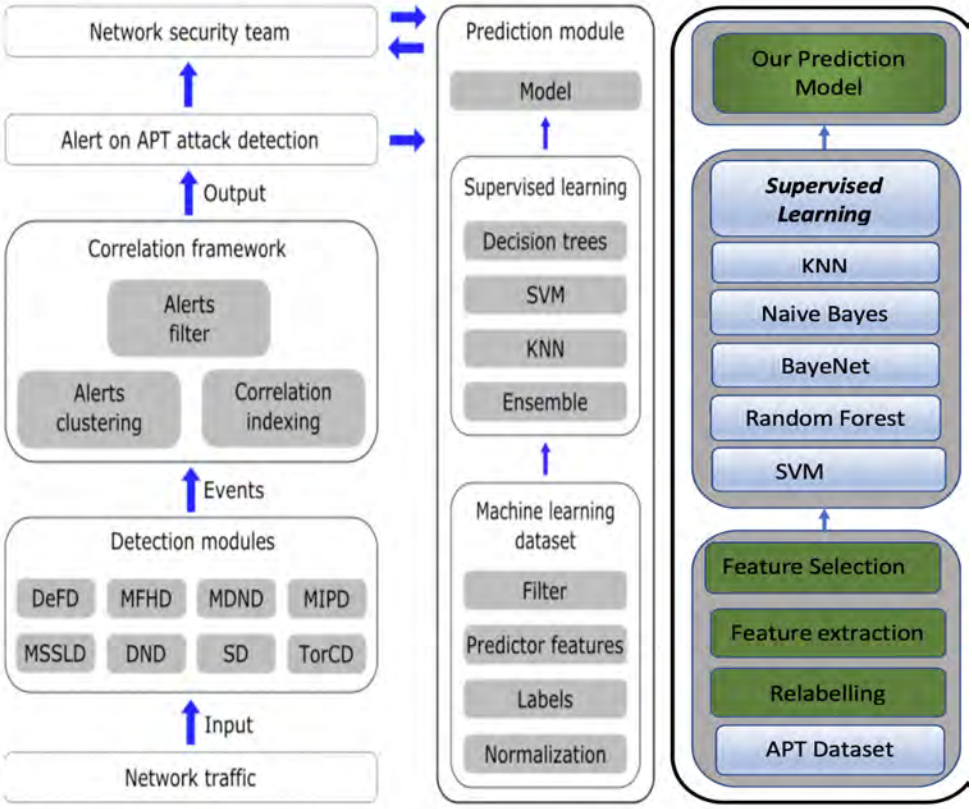

Figure 5.1: MLAPT [5] alongside our work

Cyber Kill Chain	APT Lifecycle [5]	Detection Methods [5]	Our Proposed Work	
			APT Lifecycle	Detection Methods
Reconnaissance	Intelligence Gathering	None	Reconnaissance	Robot.txt access [176] OS fingerprinting [175] DNS enumeration [177] DNS honey token [178] Port scanning [175]
Weaponization		None	Weaponization	
Delivery	Point of entry	Disguised exe file Malicious file hash Malicious domain name	Delivery	Disguised exe file [5] Malicious file hash [5] Malicious domain name [5] Infected USB [180] Malicious URL [181] Injection attacks [182]
Exploitation			Exploitation	Brute force attack [5] Pass has detection [5] PowerShell [183] Task Schedule Scripting
Installation			Installation	Brute force [5] Privilege escalations Injection attacks [182]
Command and Control (C&C)	C&C Communications	Malicious IP address Malicious SSL Malicious Domain Flux	Command and Control (C&C)	Malicious IP address [5] Malicious SSL [5] Malicious Domain Flux [5] Tor connection [5]
Action on Objectives	Lateral movement Asset/Data recovery Data exfiltration	Tor connection Scanning	Action on Objectives	Tor connection [5] DNS Tunnelling [186]
			Internal Reconnaissance	Scanning [5]

Figure 5.2: APT alerts mapped to the CKC, demonstrating state-of-the-art assignment with experimental machine learning and comparison with our work

In the next part, the data preparation stages will be covered and how the alerts align to the Cyber Kill Chain stages.

5.1.8 Data preparation stage

The data preparation stage is one of the most important tasks in the creation of machine learning datasets. The original APT dataset consists of 8 features and 3676 observations mapped to a label comprising 6 APT stages proposed

by the authors. The dataset was then reconstructed, performed feature extraction and selection during the dataset preparation stage, as shown in Fig. 5.3. In the next part, we are going to explain our feature extraction and selection process.

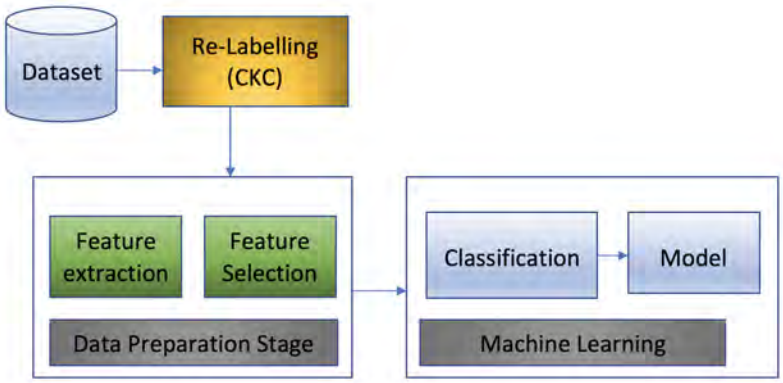

Figure 5.3: Data Preparation Stages

5.1.9 Feature Extraction

Feature extraction represents the task of obtaining a set of features from sample data and enhancing the classifier's performance [211]. In our experiment, we considered several feature extraction methods, including Principal Component Analysis (PCA), Multiple Correspondence Analysis (MCA), and Multiple Factor Analysis (MFA). PCA denotes a method for reducing large datasets dimensionality while minimizing information loss using linear combinations (weight average) of a set of variables [38]. MCA is another statistical technique best suited for tables with individuals described by several qualitative variables [212]. MFA is a PCA variation, making it possible to analyze more than one data table representing a group of variables collected

on the same observations [213]. Given that this dataset contains qualitative and quantitative variables, MFA was selected as our feature extraction method. However, other feature extraction methods such as Stacked Autoencoders (SAE) could be used to extract features. This research explored a low complexity machine learning process that can achieve sufficient performance, including faster speeds. SAE will involve neural networks that introduce complexities such as resource constraints and is more suited to unsupervised tasks.

5.1.10 Feature Selection

Feature selection refers to selecting only the most important features based on their ranking to reduce complexity, remove noise, and increase the model's efficiency. The feature selection process's objective is to build a less complex but comprehensive model without compromising accuracy [214] by removing redundant or less relevant futures. This work selected Information Gain (IG), Gain Ratio (GR), and OneR as the feature selection methods, as shown in Table 5.3. The main reason for selecting these methods is that they all provide scores and rank features according to their relevance.

5.2 Attack Stage Classifiers

5.2.1 Feature Selection

This work applied a set of machine learning classifiers, namely BayesNet, Naive Bayes, Support Vector Machine (SVM), Random Forest, and KNN.

These classification techniques were discussed in Section 2, which is the background section. Weka machine learning tool was used for this experiment.

5.2.2 Analysis and Discussions

This section discusses the steps involved in the setup of our experiment and the analysis of the results, starting with the performance evaluation metrics, which will be used to examine the model's effectiveness. This will be followed by the experiment setup and a reflection of the results.

5.2.3 Evaluations Metrics

The performance of the model was investigated using performance metrics. These metrics include accuracy (Acc), detection rate (DR), F-measure (F1), and false alarm rate (FAR). The accuracy score is a reflection of the effectiveness of the algorithms used. The detection rate is the number of actual stages detected over the total number of stages detected in the dataset. The measurement for these metrics is defined in Eqs. (5.1)–(5.7) as used in [215].

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \tag{5.1}$$

$$DR = \frac{TP}{TP + FN} \tag{5.2}$$

$$Precision = \frac{TP}{TP + FP} \tag{5.3}$$

$$FAR = \frac{FP}{TN + FP} \tag{5.4}$$

$$FNR = \frac{FN}{FN + TP} \tag{5.5}$$

$$F\_1 = \frac{2TP}{2TP + FP + FN} \tag{5.6}$$

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{5.7}$$

Table 5.1: Results of feature selection with Naïve Bayes classifier

Feature selection	Acc (%)	DR (%)	FAR (%)	F_1 (%)
OneR	91.1	91.1	1.3	91.2
Gain Ratio	90.5	90.5	1.2	90.6
InfoGain	87.3	87.3	1.9	87.4

Herein TP and TN refer to True Positive and Negative, respectively, while FP and FN denote false positives and negatives. DR refers to Detection Accuracy. Table 5.1 shows an example of the results obtained using the Naïve Bayes classifier.

Table 5.1 shows the result from the performance metrics using Naive Bayes. In this example, the highest prediction accuracy of 91.1% was ob-

tained with features from OneR, while the lowest FAR of 1.2% was obtained with features from GainRatio. The highest detection rate was achieved with OneR. The results from the F1-measure showed features from OneR scored the highest results with a prediction of 91.2%.

5.3 Experimental Setup

This experiment began by examining the original dataset to understand the various features and observations. The following steps were then performed to reconstruct and relabel the original dataset.

- Removed feature "alert id" from the original dataset. This was a redundant feature that was not contributing to our model, leaving us with 7 features.
- Perform classification based on the 7 features using Naive Bayes, Bayes Net, k-NN, Random Forest, and SVM classification algorithms. These will be our baseline results.
- Extract a further 7 features using MFA from the original 7, giving us a total of 14 features.
- Perform classification on the 14 features using the same classifiers.
- Selected the top 10, 7, and 5 features in turns and performed the classification.

Table 5.2: Numerical experiment scenarios

Experiment stages	**Description**
Dataset-CKC stage labelling	labelled dataset mapped to CKC stages
Dataset-CKC+MFA Extraction	labelled dataset mapped to CKC stages and feature extraction
Dataset-CKC+MFA+FS1	labelled dataset mapped to CKC stages, feature extraction and feature selection1(FS1)
Dataset-CK+MFA+FS2	labelled dataset mapped to CKC stages, feature extraction and feature selection2(FS2)
DatasetCK+MFA+FS3	labelled dataset mapped to CKC stages, feature extraction and feature selection3(FS3)

5.4 Results and Discussions

The main aim of this research was to improve the detection accuracy of the APT stages. A dataset on APT was used, which other researchers shared. The main challenge faced was the limited number of features on the dataset. This limitation was addressed by performing feature extraction and selection techniques. Our experiment set the threshold of a satisfactory outcome to be 84.9% for the prediction accuracy based on the original APT dataset provider's work. Our results achieved a prediction accuracy of 91.1%, which was more than the threshold. The results of our feature extraction and selection processes will be discussed in the next part, followed by our classifier results.

5.4.1 Results from Feature Extraction and Selection

The original APT dataset contains 8 features and 1 label. The "alert_id" was removed, and resulting in 7 features. R package's FactoMinerR was used to convert the non-numerical features to categorical features before extracting the feature. The final features were 14 features in total, including 7 extracted features. Information gain, Gain ratio, and OneR feature selection techniques were then used to choose the features that contributed most to our model. All the features were selected from the start, including the extracted ones. The features were then reduced gradually until the optimal level.

The features were ranked from highest to lowest using the techniques described above. The results showed that 14 of the features had a value greater than zero, which means the MFA feature extraction technique successfully

extracted the features relevant to the model. All our extracted features had a value greater than zero. Further feature selection processes were then performed until the final 5 features were left. Table 5.3 shows the top 5 features from the InfoGain, GainRatio, and OneR feature selection methods. Features from OneR produced better results, followed by the features from GainRatio and then InfoGain. The top 5 features from OneR consist of two original features and three extracted features, while the top 5 features from GainRatio consist of three original features and two extracted features. The top 5 features from InfoGain are all original features, but their prediction accuracy was less than the other two method's features. The results from the experiment stages were compared, and it was found that our feature extraction and selection processes improved the model's prediction accuracy.

Table 5.3: Selected features used across all the selected classifiers

Feature Selection Methods	Selected top 5 Features
OneR	1,6,13,14,9
GainRatio	1,6,10,5,13
InfoGain	1,5,6,3,7

5.4.2 Classifier Results

The classification of the remaining 7 features was performed once the data relabelling and the removal of redundant features were completed. Table 5.4

shows the classifier's results, including their prediction accuracy. This result will be our baseline. The result shows that the highest accuracy score of 87.43% was obtained with the SVM classifiers.

Our next step was to perform classification on the 14 features, including the 7 extracted features. The result shows that the highest prediction accuracy of 87.87% was obtained with the SVM classifier, as shown in Table 5.4. We then performed the classification using the top 10 features consisting of 4 original and 6 extracted features, which shows our extracted features are relevant to the model. The original features are feat1, feat2, feat6, and feat8, while the extracted features are feat9, feat10, feat11, feat12, feat13, and feat14. The result showed improvements in accuracy compared to the 14 features. The highest accuracy of 91.41% was obtained with SVM.

After analysing the top 10 features' classification results, the top 7 features were then selected and performed further classifications. The 7 features in the ranking were feat1 and feat6 from the original dataset and feat9, feat11, feat12, feat13, and feat14 from the extracted features. The results show a slight decrease in the accuracy results compared to the top 10 features. The top-performing classification algorithm was Bayes Net, which had a prediction accuracy of 90.85%. Finally, the top 5 features were selected according to their ranking score, and the highest prediction accuracy of 91.1% was obtained with Naive Bayes. Tab. 5.2 shows the experiment stages and the corresponding description. There are five stages in total, which start with the relabelled dataset until the final stage, consisting of the relabelled data, extracted features, and top selected features. In this table, CKC stands for the Cyber Kill Chain, and FS stands for feature selection. In FS1, FS2, and

FS3, the top 10, 7, and 5 features were selected, respectively. Fig. 5.4 shows the experiment stages and the selected features, along with the results obtained from the classifiers. From the figure, it is evident that the prediction accuracy is affected by the number of features, as shown in Fig. 5.5.

Experiment	Features (%)	NB (%)	BN (%)	k-NN (%)	RF (%)	SVM (%)
Dataset-CKC stage labelling						
	7	87.31	82.15	83.58	84.32	87.43
Dataset-CKC+MFA Extraction						
	14	82.02	83.14	83.45	80.65	87.87
Dataset-CKC+MFA+FS1						
	10	87.43	91.35	89.61	87.87	91.41
Dataset-CK+MFA+FS2						
	7	88.24	90.85	89.86	87.68	90.79
Dataset-CK+MFA+FS3						
	5	91.07	90.73	89.05	87.43	90.79

Figure 5.4: APT-Feature selection and accuracy results

Herein NB and BN refer to Naive Bayes and Bayes Net, respective, while RF and SVM denote Random Forest and Support Vector Machine as shown in Table 5.4.

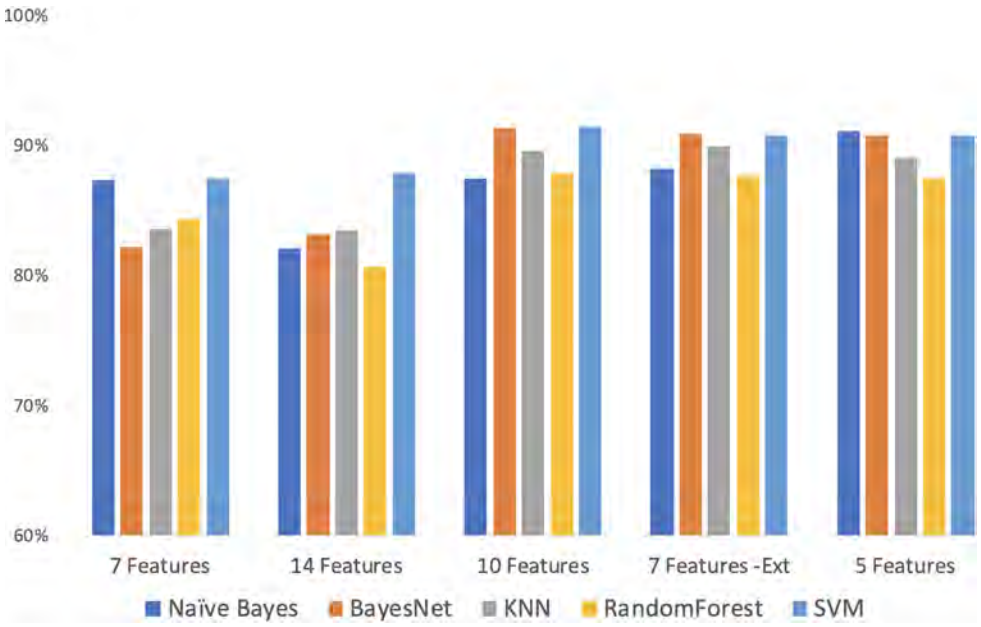

Figure 5.5: Classifier accuracy rates under various numbers of selected features for classification

The following parameter were applied when performing the classification using the various algorithms.

In Bayes net the Estimator was set Simple Estimator -A 0.5, this search parameter is used for estimating conditional probability, the SearchAlgorithm was set to K2 _P -S BAYES, InitAsNaiveBayes:True, MarkovBlanketClassifier:false and maxNrOfParents was set to 1. In KNN the K value was set to 5. In Random Forest the parameters were set to RandomForest -P 100 -l 100 -num-slots 1 -K2 -M1.0 -V0.001 -S1 with Numiterations set to 100 and the Numfeatures set to 2. The parameter is SVM was set to C=1.0, FilterType:

Normalize training data and Kernel set to Polykernel -E 1.0 -C 250007.

In this work, our proposed approach has been studied to detecting APTs relevant attacks using the Cyber Kill Approach. Further research may consider applying the Cyber Kill Chain concept to securing specific areas of IoT-enabled applications.

5.4.3 Summary

This section discussed our work on the machine learning approach for detecting Advanced Persistent Threats using the Cyber Kill concept. As part of this experiment, we performed feature extraction and selection techniques to increase the features and tune the model using the best features. This next subsection will explore feature extraction and selection techniques further using deep learning techniques such as auto-encoders and wrapper method for feature selection. We published a paper on this work in a peer-reviewed journal [216]. My main contribution to this paper was on the feature selection methods. We then combined the features from the three algorithms to obtain the best features for the model. In the next subsection, we explore the main work of this paper and my main contribution to the article. A conclusion that summarises the key points will be provided towards the end of this chapter.

5.5 Machine learning techniques for improving intrusion detection

Intrusion detection is an area that has been widely researched for many years but the accurate detection of cyberattacks remains a challenge. Recent research demonstrated advances in intrusion detected [217], [38], [218], [219]

Machine learning approaches for intrusion has yielded some good results depending on the quality of the dataset used and classification algorithms, which perform better under various scenarios. Most machine learning datasets start with data preparation, followed by feature extraction or selection before the data is fed to the model.

In the earlier APT stage detection, feature extraction and feature selection methods such as MFA, Information gain and Gain ratio were used. Machine learning classifiers such as SVM, KNN and Random Forest were applied. To expand on that work, we collaborate with other colleagues on expanding the use of feature selection in intrusion detection. The resulting work explored feature extraction based on deep learning, used a wrapper-based classifier for feature selection and Artificial Neural Networks (ANN) for classification [216].

My main contribution to this paper was in the feature selection methods. In the paper, an effective feature selection technique was leveraged to improve intrusion detection. The proposed approach used deep feature abstraction in the form of unsupervised auto-encoders to extract more features. Wrapper-based feature selection techniques were then utilised using Support Vector Machine (SVM), Naive Bayes and Decision tree to select the highest-

ranking features. Artificial Neural Networks (ANN) classifier was then used to distinguish impersonation from normal traffic.

5.6 Feature Selection

Feature selection is a technique widely used to rank features according to their relevance. The two main categories of feature selection are wrapper-based and filter-based. The primary reason for feature selection is to reduce noise and optimise the performance of the model. Fig. 5.6 depicts a high-level overview of the feature combining process.

Wrapper methods for feature selection are widely used to deal with classification problems and reduce noise by selecting the optimal features for the model. In this paper, we proposed a method for selecting features using three different algorithms and then combined the features. The wrapper based techniques used in this paper were Support Vector Machine, C4.5 and Naive Bayes.

The total number of features in the dataset was 204, and we selected the top 20 features. We selected the 7 top-ranked features from SVM and C4.5 and then the top 6 features from NB, which gave us a total of 20 top features. To reduce duplication due to the features overlapping, we skipped to the next feature if it already appeared in the top features for any of the other algorithms used. The classification was then performed for the combined features using Artificial Neural Networks, and we achieved an overall accuracy was 99.95%.

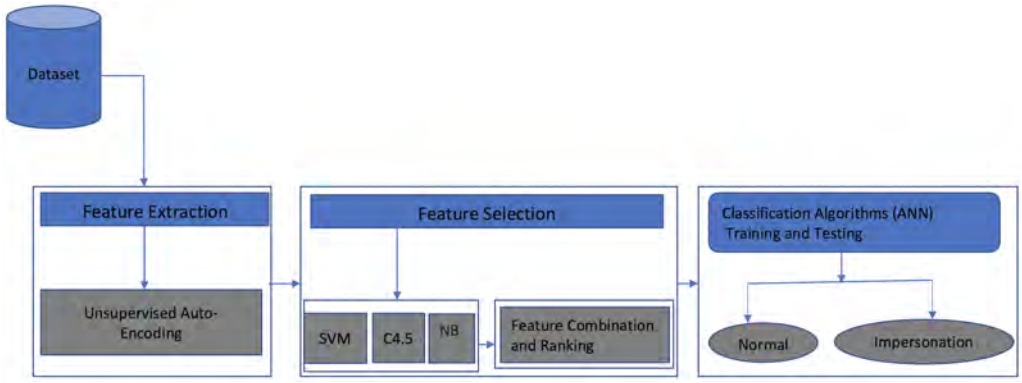

Figure 5.6: Architecture of combining best feature and classification

5.7 Conclusion and Key Points from Our APT experiments

This chapter discussed machine learning approaches for detecting APT attacks by leveraging the Cyber Kill Chain concept. APT attacks are challenging to detect and can cause substantial damages. Although some progress was made on APT detection, it remains a challenge, given the time it takes to detect the APT attacks. According to a recent report by FireEye [80], it takes on average; 56 days to detect APT attacks. We used the APT dataset provider's work [5] as a baseline to expand on the machine learning aspect of their work. In this work, we also used feature extraction and selection techniques to increase our model's efficiency. The APT dataset used had a limited number of features. We overcame that limitation by performing feature extraction techniques using Multiple Factor Analysis (MFA), which doubled the number of features. We then performed feature selection using Information Gain (IG) and Gain Ratio (GR) methods to deliver our model's

most relevant features. We selected the top 5 feature IG and GR methods and obtained a detection accuracy of 91.1%.

To explore the feature selection techniques, we published another paper on "Effective combining of feature selection techniques for machine learning-enabled IoT intrusion detection". In this work, we utilised feature extraction using deep learning techniques. We then performed feature selection using the wrapper methods, selected the top 20 features, and obtained an overall detection accuracy of 99.5%.

We hope our proposed alert modules mapped to Cyber Kill stages will help detect APT attacks and reduce the cost of data breaches. Furthermore, the feature selection and extraction concept explained in the second paper can also be applied to APT stage detection in the context of the Internet of Things (IoT).

Chapter 6

Feature forecasting for cyber attack prediction

This chapter presents our work on cyber attacks events forecasting and prediction based on time series data. Some of the content of this chapter is derived from the author's work, which was recently submitted to a peer-reviewed journal [119]. In this work, we used a recently released dataset that contains several attack types on a realistic secure network and then constructed times-series models with tuned parameters to assess the effectiveness of the various time series forecasting techniques such as Linear Regression, Sequential Minimal Optimization for regression (SMOreg) and Long Short Term Memory (LSTM) to forecast the cyber events.

Time series data forecasting is not new, and it is widely adopted in fields such as weather forecasting and stock predictions. Forecasting has been gaining traction in cyber attack prediction, although this is still emerging area [145]. While several predecessors have carried out work on cyber attack

prediction and forecasting, there are limitations on the datasets' quality, which are mainly derived from honeypots [129, 148] and social media feeds [126, 147, 146]. Such work often looked at a single attack, such as Denial of Service (DoS) and malware variant. Most of these existing works on forecasting does not provide enough information to help implement proactive approaches due to the dataset limitations. To overcome these challenges, we have used a large dataset with multiple attack labels [34]. We then performed event forecasting to predict cyber attacks within a specific time frame. The main contributions of our research are as follows:

1. Perform time series resampling based on original data to make sure we have equally-spaced samples for prediction.

2. Perform and evaluate time series forecasting based on linear regression, SMOreg and LSTM.

3. Evaluate the performance of the forecasted events using the metrics MAE and RSME.

4. Use time-series data to forecast cyber-attack events within a specified period.

6.1 Cyber Event Forecasting Model

In this work, we created a time series based cyber event forecasting model. The model's primary goal is to predict cyber-attack events and help decision-makers take proactive measures to protect their networked systems. To

achieve this research's objectives, we used a publicly available dataset containing various attack types on a realistic and secure network. The distribution of the data was Benign 64%, SSH-Bruteforce 18% and FTP-Bruteforce 18%. The dataset providers selected the attack types based on McAfee report on the most common attacks [220].

The dataset consists of several attack types collected over five days, and our focus was on data collected over 24 hours. The dataset contains three labels which are SSH-Bruteforce, FTP-Bruteforce and Benign data. The dataset consisted of 79 features and 1048575 observations. Data resampling will be performed based on 30 seconds intervals, and from there, the data was partitioned into training and test portions.

Forecasting is not new, and it has been widely used over the years to predict customer trends, energy consumption, weather patterns and stock forecasting. Several authors have carried research on forecasting energy consumption in smart meter environment [221, 222, 223, 224, 225]. Weather forecasting has been used for many years, and we are used to checking the weather forecasts regularly. Several authors carried research on weather forecasting using machine learning techniques to improve the accuracy of the prediction [226, 227, 228, 229].

Although time series forecasting techniques are well established in weather forecasting and stock prediction areas, it has not been widely explored to predict cyber-events due to the challenges of the ever-changing threat landscape and the volume of data exchanged, which can be quite high even for a small network. Cyber-events forecasting is a proactive measure that can help with early detection and help decision-makers make informed decisions,

and take corrective measures to mitigate potential threats. In this work, we will focus on cyber-events forecasting to help anticipate future attacks and determine whether an attack is likely to occur at a given time based on a combination of certain features and events. The purpose is to help technical teams and decision-makers to intervene before cybercriminals execute their malicious activities and compromise their systems. Fig. 6.1 shows the data preparation stages, including the forecasting.

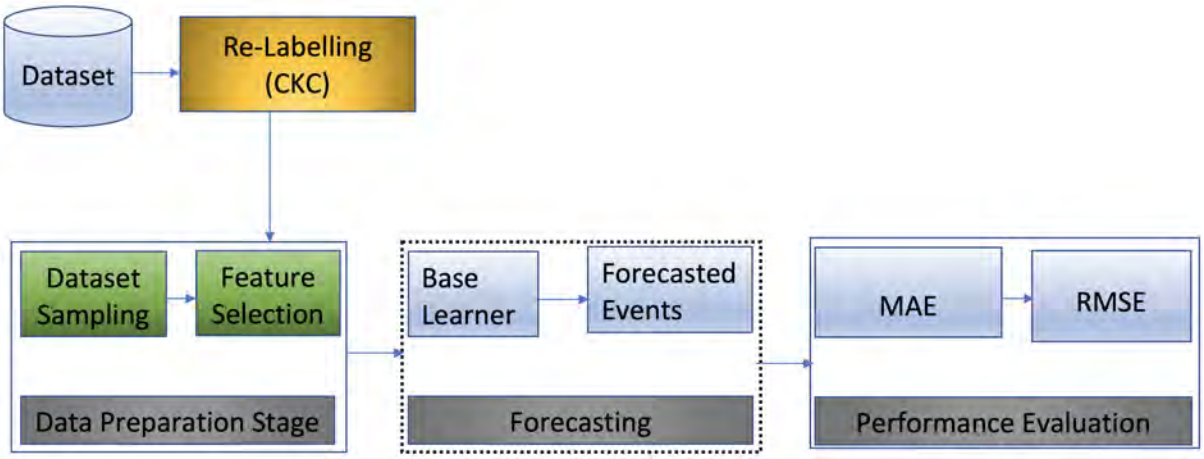

Figure 6.1: Forecasting Stages

6.1.1 Data Preparation

This work used the CSE-CIC-IDS2018 dataset [34], a recently released public dataset on intrusion detection. We initially went through the dataset and explored the various features and labels. The sequence of the dataset preparation and experiment stages are (i) Dataset preparation and feature selection, (ii), Time series forecasting, (iv) Performance evaluation, as shown in Fig. 6.1.

Our focus was on the portion of the original dataset collected over 24 hours and consisted of 79 features and 1048575 observations. The next step was to perform data selection based on a time interval of 30 seconds and ended up with 79 features and 1084 observations. The time-series data was then checked to ensure it was stationary.

6.1.2 Feature selection

Feature selection was applied to reduce the number of features and increase the model's efficiency using Information Gain (IG) selection methods that use ranking based on their relevance to the model. The top 21 features were then selected based on IG. These features were selected based on their ranking and contribution to the model. The first 884 portions of the data were selected as training data and the remainder 200 as the test data.

6.2 Experiment Setup

The CSE-CIC-IDS2018 dataset was used in this experiment, and data resampling and time series forecasting were then performed. This was a large dataset collected over five days and consists of seven attack types which are (i) Bruteforce, (ii) DoS attack (iii), Web attack, (iv) Botnet attacks, (v) Infiltration, (vi) DDos, (vii) Heartbleed. Our research's primary focus was data collected over 24 hours containing SSH-Bruteforce, FTP-Bruteforce and Benign data.

The victim network consists of five departments with 450 computers and 30 servers, and while the attack network consists of 50 machines. The cap-

tured data includes network traffic and log files from each host in the network, making it a comprehensive dataset. Time series were used to collect the data at regular intervals of 30 seconds which reduced the dataset to 1084 observations. The dataset was then divided into training and test portion.

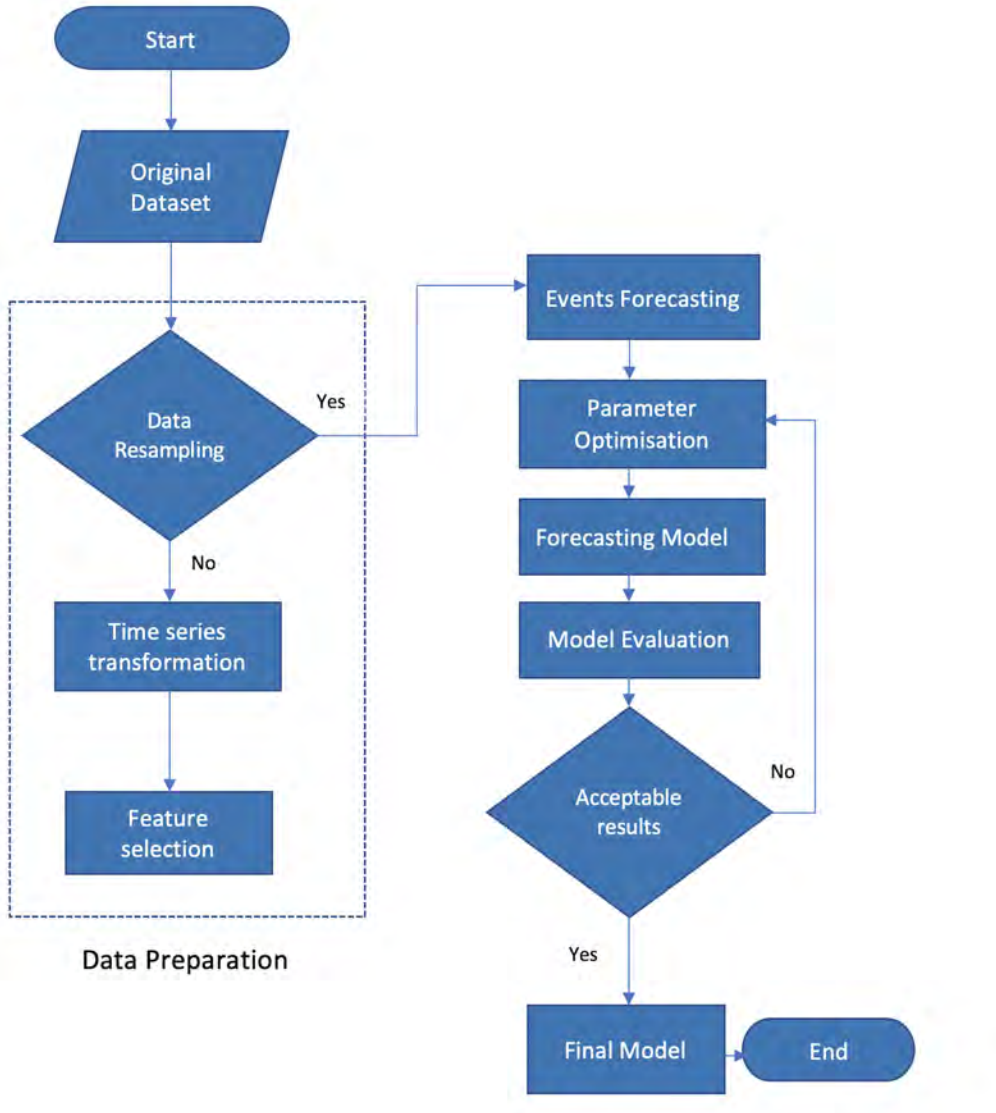

Figure 6.2: Flow chart - forecasting model

The flow chart in Fig 6.2 shows the steps and algorithms used in the experiment. We believe that the flow chart will be more representative to capture the step that was done in applying machine learning.

6.2.1 Experiment overview

Time-series resampling was performed based on the original data to ensure equal sampling for the prediction. The resampling process reduced the dataset to 1084 observations during the data preparation stage, and from there, we reserved 884 observations for training and the reminder 200 for testing. The next step was to perform the forecasting based on the subsequent 200 events. WEKA machine learning tool was used to perform the experiments. During the experiment, several base learners such as Linear regression, SMOreg, LSTM, Gaussian and multilayer perceptron were explored but eventually selected Linear regression, SMOreg and LSTM for our experiment. The next step was to perform the cyber event forecasting using these base learners in turns and evaluate the performance using metrics such as Mean Absolute Error (MAE) and Root Square Means Error (RSME). MAE gives the absolute value of the error and is useful when determining the difference between the actual and predicted values. RSME is another metric for evaluating the performance of models and is used to show how far the predictions are from the actual values using Euclidean distance. This work will be using both of these metrics to measure the performance of our model.

6.3 Time Series Forecasting

This work performed forecasting on time series data to predict attacks based on events' patterns. In the forecasting, the time series data were collected at the regular intervals of 30 seconds, reducing the dataset to 1084 observations. The following base learners were explored during the experiment (i) Linear regression, (ii) SMOreg, (iii) LSTM, (iv) Gaussian Processes (v) Multilayer Perceptron. The next step was to forecast 200 events using each of the base learners described above.

The events that were forecasted through linear regression, SMOreg and LSTM were then chosen. Next, the parameters used in the forecasting experiment will be listed. In Linear regression the parameters used were: (i)attributeSelectionMethods was set to M5 methods, (ii) eliminateColinearAttributes was set to true, (iii) ridge - the default value was selected. In SMOreg the following parameter were set (i) the c value = 2.0, (ii) Kernel = PolyKernel, (iii) RegOptimizer = RegSMOImproved, (iv) filtetype = Normalize training data. In LSTM the parameter were set to (i) Activation function = ActivationReLU, (ii) Number of outputs = 3, (iii) gate Activation function = ActivationSigmoid. The result of the experiment can be found in figures 6.2 and 6.2.

6.4 Performance Evaluations

In this work, the metrics Mean Absolute Error (MAE) and Root Square Mean Error (RSME) were used to evaluate the forecasted data's performance. These metrics will be discussed in sections 6.4.1 and 6.4.2

6.4.1 Mean Absolute Error (MAE)

The metric Mean Absolute Error (MAE) was used in this experiment to evaluate the times series forecasted data's performance. Table 6.1 shows the result of the MAE. In Table 6.1, we have selected the top five forecasted features obtained with linear regression, SMOreg and LSTM. The results show that SMOreg forecasted features or events performed better than the Linear regression and LSTM predictions by producing the lowest MAE value although linear regression was not far behind. The equation below shows how MAE is calculated as used in [230]

$$MAE = (\frac{1}{n}) \sum_{i=1}^{n} |y_i - x_i| \tag{6.1}$$

Herein y_i = prediction, x_i = actual value and n = total number of data points.

Target Feature	**Linear Regression**	**SMOreg**	**LSTM**
Tot Fwd Pkts	0.0105	0.0038	2.6127
Tot Bwd Pkts	0.0034	0.0025	0.4981
Pkt Len Min	0.0002	0.008	0.0054
Fwd Seg Size Min	0.4912	0.0182	0.0494
Subflow Bwd Byts	0.718	0.7251	31.9957

Table 6.1: Mean Absolute Error

6.4.2 Root Mean Square Error Absolute Error (RMSE)

The Root-Mean-Square Error (RMSE) measures the accuracy of predictions obtained by a model by measuring the differences between actual and predicted values. In the RMSE, the first step is to calculate the difference between the numbers and then square them. This is followed by finding the mean of these numbers and, finally, the square root of the mean is calculated as shown in equation 6.2.

Target Feature	Linear Regression	SMOreg	LSTM
Tot Fwd Pkts	0.0129	0.0045	2.8946
Tot Bwd Pkts	0.0039	0.003	0.5842
Pkt Len Min	0.0002	0.0096	0.0063
Fwd Seg Size Min	0.0198	0.8776	0.0497
Subflow Bwd Byts	0.8163	1.2969	35.0395

Table 6.2: Root Mean Square Error

$$RMSE = \sqrt{(\frac{1}{n})\sum_{i=1}^{n}(y_i - x_i)^2} \tag{6.2}$$

6.4.3 Long short-term memory (LSTM)

Recent advances and availability of large data have resulted in the popularity and application of deep learning-based algorithms such as Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) to forecast future trends in sectors such as finance. For example, LSTM models have been deployed to process events sequence and are widely adopted in time

series forecasting. LSTM is a variant of Recurrent Neural Network(RNN) and can predict future events based on previous data. It performs better than traditional forecasting methods such as Auto-Regressive Moving Average (ARIMA), which is one well known classical forecasting method [231]. Although ARIMA is also widely adopted for time series based forecasting, it has its own limitation, such as the inability to model nonlinear relationships between variables [232]. In this experiment, LSTM was used to forecast the subsequent 200 events. The metrics Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) were then used to evaluate the model's performance. Figures 6.1 and 6.2 show the performance results of the two metrics. The results show that LSTM forecasted events had the least accuracy compared to those obtained through linear regression and SMOreg.

6.5 Analysis of the results

In this section, an analysis of the results of this experiment is provided. The results show that the events forecasted through SMOreg performed better than those from linear regression and LSTM when the metric MAE is used for evaluation. SMOreg forecasted event produced the best predictions in three out of the 5 top features selected. These features were Tot Fwd Pkts, Tot Bwd Pkts and Fwd Seg Size Min, as shown in table 6.1. The lower the MAE score, the better the performance. Linear regression forecasted events produced the second-highest predictions, with two out of the 5 top features performing better than the other two base learners. When the RMSE metric was used to evaluate the model, the linear regression forecasted events

produced the highest predictions accuracy in three out of the top five features selected. In contrast, SMOreg forecasted events produced the second-highest predictions in two out of the 5 top features selected, as shown in table 6.2. LSTM produced the lowest accuracy for MAE and RMSE evaluation metrics compared to SMOreg, and linear regression forecasted events.

Security teams can use these metrics to determine the model's accuracy and forecast to anticipate cyberattacks and implement corrective measures before the actual attacks are executed.

In the next part, the performance metrics and evaluate results will be covered.

6.6 Summary

In this chapter, our cyber events forecasting and prediction model was discussed. The experiment used a recently released IDS dataset captured from realistic network settings, and the dataset contained seven different attack types and benign traffic. The research began by preparing the dataset, performing data re-sampling and feature selection. Cyber events forecasting was then performed using the base learners linear regression, SMOreg and LSTM to forecast the subsequent 200 events for each technique. The metrics Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) were then used to evaluate the performance of the forecasted events. The results showed that the forecasted events from SMOreg performed better by producing the lowest MAE compared to others in three out of the five top selected features. At the same time, linear regression forecasted events performed

better in three of the selected top 5 features when the RMSE metric was used for the evaluation.

Given the limited research on cyberattack prediction using time series forecasting, we believe our work will contribute to the accurate detection of cyberattack detection and ultimately prevent data breaches by allowing security professionals and decision-makers to anticipate attacks and take proactive measures to prevent potential attacks before they occur. Our forecasting was limited to specific time frames in this work, which was in hours due to the dataset's constraints. However, we believe this is a reasonable time to take corrective measures given the evolving threat landscape. We also plan to expand on this work in our future work and increase the forecasting window to between 1 to 7 days.

Chapter 7

Threat detection using Indicators of Compromise (IoC)

Indicator of compromise are artefacts left behind following the action of malicious actors. These artefacts can be found on a network or computer systems, giving the investigator some degree of confidence that a cyber-attack or intrusion has taken place. Threat hunting is a proactive approach to searching for cyber threats that were not detected and hidden in the network. These artefacts can be found in networks, endpoints or previously collected datasets. The investigations performed by the threat hunters is influenced by factors such as the threats landscape and known IoCs or indicators of attack. There are triggers such as malicious activities on computer systems or networks that can also result in investigations to be carried out by the threat hunters. Our focus is on data-driven investigation and utilises Machine learning approaches to detect IoCs. We look at system logs to search for hidden anomalies to build the pattern of the attack.

In this chapter, we present our work on cyber threat detection using Indicators of Compromise. Some of the contents of this chapter are derived from the author's work, which was recently submitted to a peer-reviewed book chapter [149]. We designed an experiment and simulated attacks on Windows-based systems using Kali Linux as the attack machine. We collected data from logs and alert systems to create a dataset for machine learning classification. The experiment was in two stages, and the first part involved 3 Windows clients and a Kali machine. Although this experiment produced some good results with the best accuracy of 96.7%, there was the challenge of over-fitting, which we had to deal with due to the limitation and size of the dataset. To expand on this work, we performed further experiments and increased the datasets to 42 features and 215 observations from the original 29 features and 87 observations. We also added two new attack labels, which were reconnaissance and infiltration attacks, to the dataset.

Although several predecessors have researched indicators of compromise and its role in intrusion detection, there have been some shortcomings and challenges in their broader implementation. These challenges include: (i) limited availability of publicly available datasets on IoCs (ii) limited work on host-based IoC mainly due to restrictions on information sharing for fear of litigation (iii) limited experimental work to validate the effectiveness of the proposed solutions. In this work, we will target (ii) and (iii) by creating a dataset collected from windows hosts and performing experimental validation of the work. There are very few existing host-based IOC datasets, but these are heavily anonymised, limiting their capability. Some of these datasets are also old. Anonymisation of the data is the main reason why the IoC dataset

was created in our experiment. Next, we are going to start with the first IoC experiment.

7.1 Experiment I

In the first experiment, we created a small network, as shown in Fig. 7.1 and simulated various attacks. We then collected data relating to both benign and attacks events which constituted our dataset. The experiment setup consists of a Kali Linux attack machine and three Windows clients. The Windows machines were Windows 10, Window 8.1 and Window 7 clients. In this experiment, we used the stages of the Cyber Kill Chain (CKC) when simulating the attacks. The CKC was developed by Lockheed Martin and consists of seven stages which are: (i) Reconnaissance (ii) Weaponisation (iii) Delivery (iv) Exploitation (v) Installations (vi) Command and Control(C2) (vii) Action on Objectives. Other frameworks used for attack stage detection include the MITRE ATT&CK and Unified Kill Chain, which combines elements from the CKC and MITRE ATT&CK.

The Kali Linux machine was used as the attacking platform, and several attacks were executed against the Windows victim machines. The attack labels can be found in Fig. 7.5. The distribution of the attack labels is depicted in Fig. 7.6. From the diagram, it is evident majority of the attacks was directory scanning and privilege escalations followed by Brute force and SMB attacks. The target machines were attacked over a 24 hour period at random times. Kali Linux tools such as Dirb and NMAP were used for scanning attack, and Metasploit was used for the SMB attack, while hydra

was used for the Brute force attack. The Dirb tool was used for directory scanning and useful to detect hidden directories, while NMAP was used to scan for open ports and services. Metasploit was used to search and execute the exploits, while hydra was used for password cracking. Security onion was used as the network security monitoring and log management tool. Security onion contains a collection of security tools such as Snort, Zeeks and Suricata IDS systems [233]. It also contains Sguil, Squirt, Elastisearch, Logstash and Kibana, which provides the front-end visualisations. Security onion is a free open source tool that provides comprehensive solutions and can be used for threat hunting.

The dataset we created consists of 29 features, 87 observation and 5 labels. Fig. 7.3 shows the list of features and their descriptions, while Fig. 7.5 shows the attack labels. Included in the features are several Windows event IDs, as shown in Fig. 7.4. These events IDs can be used to piece together the activities under-taken by malicious actors and aid with the accurate detection of IoCs. Building the digital trail of the attackers can help incident response teams execute corrective measures to contain incidents and implement proactive approaches to prevent similar attacks occurring in the future by deploying extra security measures or fine-tuning the existing tools. Detecting intrusion in the earlier stages can help limit the damage caused by cyber attackers. IoC detection can play an essential role in preventing such attacks from moving through the network undetected and detecting any lateral movements.

Attackers do not often restrict themselves to a single compromised host, and they try to find other vulnerable systems in the network. Cybercriminals

are also known to search for online backups, especially in attacks involving ransomware. If successful, they tend to encrypt the backup first before encrypting the production systems. Therefore it is crucial to quickly detect IoCs accurately and prevent or contain the attack if it is still in progress. Logs files play an essential role in intrusion detection, and they are considered a rich source of information when looking for signs of compromise on target hosts. These logs provide information such as login attempt, privilege escalation and other suspicious activities that form broader patterns that point to an attack.

In the next part, we will explain how we prepared the dataset.

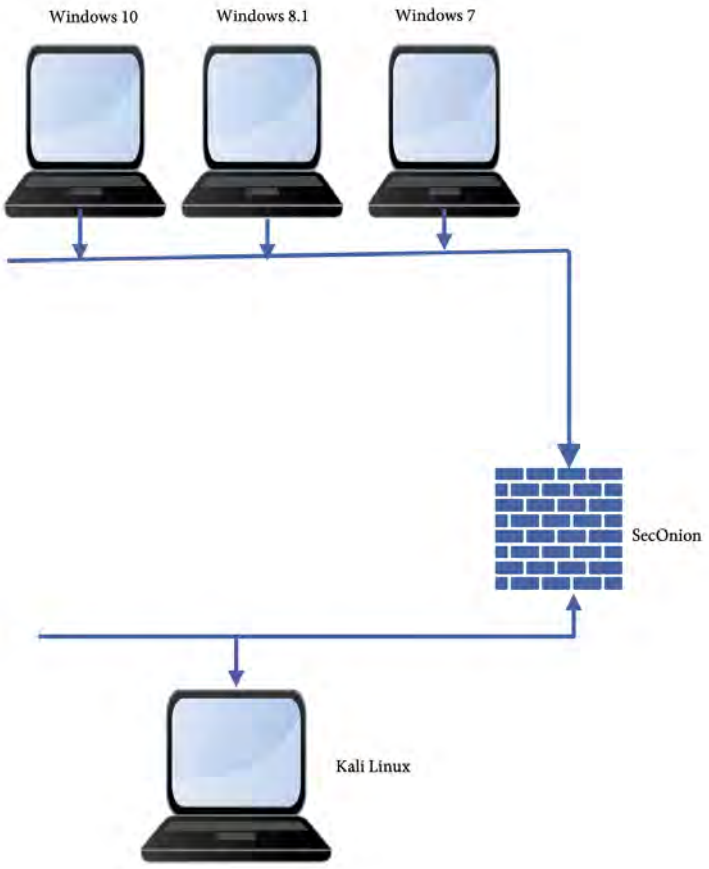

Figure 7.1: IoC Network Architecture Experiment I

7.2 Dataset Preparation

We constructed the dataset, which consisted of 29 features and 87 observations. We then explored feature selection methods to remove features that are redundant to our model. Fig. 7.2, shows the stages involved in the dataset preparation. Once we created the dataset, we performed feature selection to choose the best features and then proceeded with the machine learning classification.

Fig. 7.3, represents the list of the features contained in the dataset while Fig. 7.5, shows the labels. Fig. 7.4, describes of some of the Windows event that forms part of the features in the dataset. Windows events provide a snapshot of the activities taking place on these systems. During the preprocessing, the attributes relating to the Windows events were converted to binary form. The presence of an event was denoted by 1 and the absence of event by 0. We used Weka machine learning tool in these experiments.

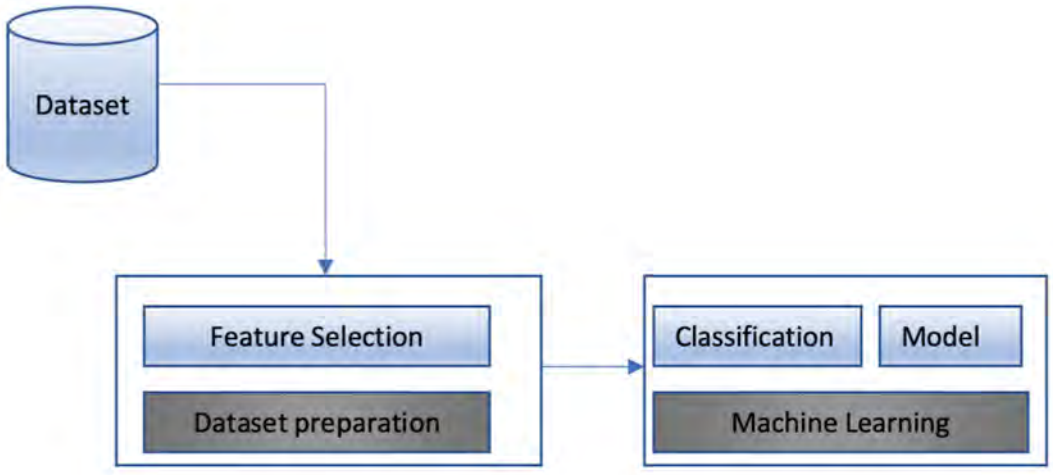

Figure 7.2: IoC Data Preparation and Classification Experiment I

In the next part, we will cover how the feature selection was performed and the chosen selection methods.

7.2.1 Feature Selection

Feature selection is the process of choosing features that are most relevant to the model. They are used to reduce noise and increase the efficiency of the model. There are several feature selection methods such as Information Gain (IG), Gain Ratio (GR), OneR, Principle Components (PC), Correlation, Wrapper and Support Vector Machine (SVM). Feature extraction can also be performed before the feature selection in order to increase the number of features. Some of the most common feature extraction methods include Principal Component Analysis (PCA) and Multiple Component Analysis (MCA). Other feature extraction methods include deep learning approaches such as auto-encoders as demonstrated in [216]. In this work, we selected Information Gain and Gain Ratio feature selection methods. The reason for selecting these methods is that they rank features according to their importance.

Feature	Description	Feature	Description
Feat1	Source IP	Feat2	Destination IP
Feat3	Source port	Feat4	destination port
Feat5	Total login attempts	Feat6	Successful login
Feat7	logon type	Feat8	packet length
Feat9	Flags	Feat10	Protocol
Feat11	Operating System	Feat12	Hostname
Feat13	Event ID 4624	Feat14	Event ID 4672
Feat15	Event ID 4634	Feat16	Event ID 4720
Feat17	Event ID 4728	Feat18	Event ID 4688
Feat19	Event ID 4905	Feat20	Event ID 4722
Feat21	Event ID 1102	Feat22	Event ID 5033
Feat23	Event ID 4902	Feat24	Event ID 4907
Feat27	Event ID 4724	Feat28	Event ID 4732
Feat28	Event ID 4625	Feat29	Event ID 5024
Labe	Attack types		

Figure 7.3: IoC Feature Description Experiment I

Windows Event ID	Event Description
Event ID 4624	An account was successfully logged on
Event ID 4672	A special privilege was assigned to new logon
Event ID 4634	An account was logged off
Event ID 4720	An account was created
Event ID 4728	A member was added to a security enabled global group
Event ID 4688	A new process has been created
Event ID 4905	An event was made to unregister a security event source
Event ID 4722	A user account was enabled
Event ID 1102	The audit log was cleared
Event ID 5033	The Windows firewall driver has started successfully
Event ID 4902	The per-user audit policy table was created
Event ID 4907	Auditing Settings was on object were changed
Event ID 5061	Cryptographic operations
Event ID 4738	A user account was changed
Event ID 4724	An attempt was made to reset an accounts password
Event ID 4732	A member was added to a security-enabled local group
Event ID 4625	An account failed to log on
Event ID 5024	Windows firewall services has started successfully

Figure 7.4: Event Logs [234]

Label ID	Description
1	Normal
2	Directory Scanning Attack
3	SMB Attack
4	Privilege Escalation
5	Brute Force

Figure 7.5: Attack label Experiment I

7.3 Attack Classification

Many classification algorithms are widely used in machine learning. In this experiment, we used BayeNet, Naive Bayes, Support Vector Machine (SVM), K Nearest Neighbour(KNN) and Random Forest. We carried classification using the best 7 features from Information Gain (IG) and Gain Ration (IG), as shown in Table 7.1. We then performed classification on the 7 top features from Information Gain using the classifiers BayeNet, Naive Bayes, Support Vector Machine (SVM), K Nearest Neighbour(KNN) and Random Forest. Table 7.2 shows a breakdown of the classifier performance. The classifier performance on the Information Gains features can also be seen in Fig. 7.7. Similarly, we performed the same steps on the 7 top features obtained through the Gain Ratio feature selection method, and the performance of the classifier is shown in Fig. 7.8. Similarly, the classification performance of the Gain ratio and Info gain features using the Naive Bayes classifier can be seen in Table 7.3.

Table 7.1: Selected features

Feature Selection Methods	**Selected top 7 Features**
InfoGain	3,16,17,20,21,27,28
GainRation	3,5,8,22,23,24,25

Table 7.2: Classifier results

Feature selection	**BN**	**NB**	**SVM**	**KNN**	**RF**
Gain Ratio	93.3%	96.7%	90.0%	90.0%	94.3%
InfoGain	94.28%	94.28%	83.33%	93.3%	96.7%

Herein BN and NB refer to Bayes Net and Naive Bayes, respectively, while SVM, KNN and RF denote Support Vector Machine, k-Nearest Neighbour, and Random Forest.

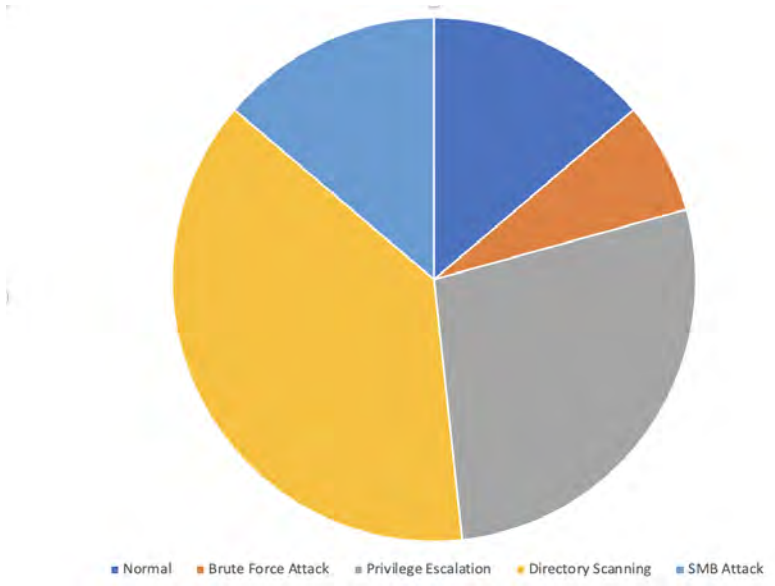

Figure 7.6: IoC Attack Label Distribution Experiment I

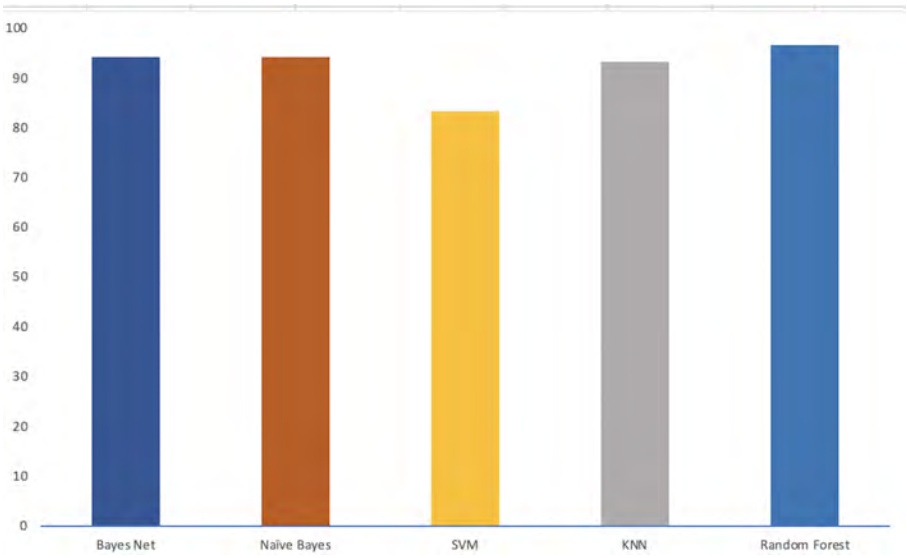

Figure 7.7: Info-Gain Top 7 Features

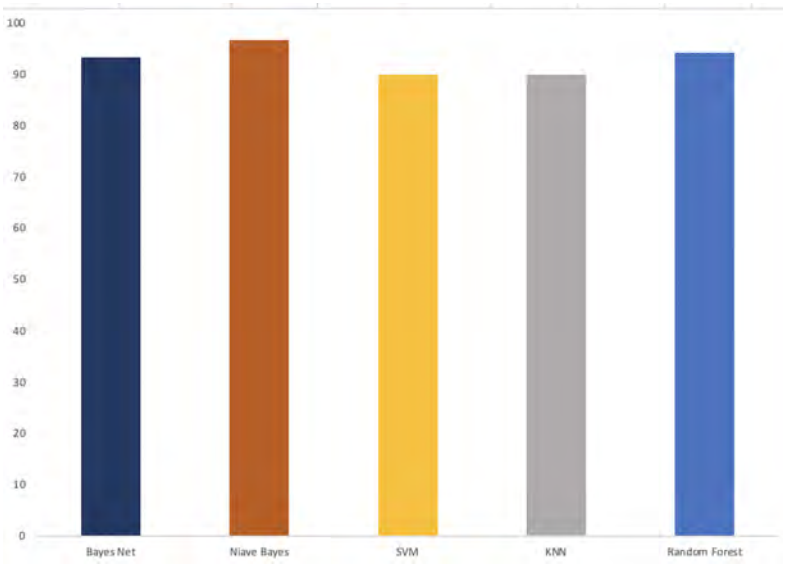

Figure 7.8: Gain-Ratio Top 7 Features

7.3.1 Evaluation Metrics

In this experiment, we used performance metrics to evaluate our model. These metrics include accuracy (Acc), detection rate (DR), Precision, F-measure (F1), Matthew correlation efficiency (MCC), False negative rate (FNR) and false alarm rate (FAR). The measurement of these metrics is defined in section 5.2.3.

Table 7.3: Performance results using Naive Bayes Classifier

Feature selection	**Acc**	**Precision**	**FAR**	**F_1**	**Mcc**
Gain Ratio	96.7%	97%	0.7%	96.6%	95.5%
InfoGain	94.28%	95.4%	0.7%	94.3%	92.8%

7.4 Analysis and Discussions

In the first part of the chapter, we explored the role of indicators of compromise (IoCs) in threat hunting and utilised machine learning approaches to improve detection accuracy. IoCs play an essential role in detecting cyber breaches and adopting proactive measures to prevent similar attacks from occurring in the future. In this work, we started with creating the experiments and collecting data that captures both attacks and benign traffic. The dataset we created was then used for machine learning classification using algorithms such as BayesNet, Naive Bayes, SVM, KNN and Random For-

est. We performed feature selection during the data preparation stage and chose the best 7 features from Information Gain (IG) and Gain Ratio (GR). We obtained a high classification accuracy of 96.7% with Naive Bayes and Random Forest.

We then used performance metrics to test the performance of the model. The metrics we used were: (i) Accuracy (Acc), (ii) Precision, (iii) False Alarm rate (FAR), F-Measure (F_1) and Matthews Correlation Coefficient (MCC). We used the top features from IG and GR feature selection methods and used the Naive Bayes classifier. The result shows features from Gain Ratio performed better in terms of the accuracy score, precision, F_1 and MCC, while the False alarm rate was 0.7% for both sets of features as shown in Table 7.3. The performance of the top 7 features for IG and GR can be also be seen in Fig. 7.7 and 7.8.

We carried further experiments to improve the work performed in the first part of the chapter to address some of the data limitation found in the first experiments. The primary challenge was overfitting, which we overcame by performing feature selection to remove features that were not contributing to the model's efficiency. In the next part, we will discuss how we performed the second experiment, in which we increased the dataset to 42 features and 125 observations.

7.5 Experiment II

In this experiment, we expanded on the work performed in Experiment I to deal with some of the challenges encountered due to the dataset limitations.

In the second experiment, we increased both the features and observations to 42 and 215, respectively, from the original 29 features and 87 observations. We also added a further two attack labels to the dataset. The experiment setup involved a Kali Linux attack machine and four Windows hosts, which were the victim machines, as shown in Fig. 7.9. The Windows machines were Win XP, Win7, Win 8 and Win 10. Similar to the first setup, we used Security Onion for network monitoring and log management. In this experiment, we used the stages of the cyber kill chain described in section 7.1.

7.5.1 Attack Simulations

Kali Linux was used as the attacking machine, and several attacks were executed against the Windows victim machines. The attack labels can be found in Table 7.6. The distribution of the attack labels for this second phase of the experiment is depicted in Fig. 7.11. From the diagram, this time, it is evident majority of the attacks were Reconnaissance and Infiltration followed by directory scanning, privilege escalations, Brute force and SMB attacks. The target machines were attacked over 24 hours at random times as per the first experiment. The attack types, tools and systems involved can be seen in Table 7.7. Security Onion was again used as the network monitoring and log management tool. The dataset preparation stage followed similar steps as described in Section 7.2. The attacks involved directory scanning attacks using tools available in Kali Linux, and these tools include Dirb and Gobuster, which can be used to discover hidden directories. In the SMB attacks, we used Metasploit to deliver working exploits to gain access to

these Windows systems. We utilised eternal blue, which is a well-known exploit for Windows SMB vulnerability.

It is common for skilled cyber attackers to perform privilege escalations once they compromise a system to perform further activities that may require elevated privileges. In this experiment, we demonstrated privilege escalation using various tools. One of the primary tools used in this experiment was Metasploit which we used to search and deploy exploit to compromise these systems and elevate the privileges. We added new user accounts to the compromised machines and then added them to the administrative group. Other tools we used for the privilege escalation include Getsystem, which gives system-level access; Mimikatz is also a useful tool for obtaining passwords and other information. In this experiment, we used Mimikatz to obtain credentials and elevate privilege. We also used it to perform pass the hash attacks, which allowed us to authenticate without the need for the actual password by using the hash. Mimikatz is a very good tool for privilege escalations. We also used Armitage, which provided a graphical interface for Metasploit and used it to deploy exploits to access these systems. For the Brute force attack, we used Hydra, which is a tool available in Kali Linux.

Reconnaissance is one of the most important steps that precede the actual active attacks. To perform reconnaissance, we used several tools against our target Windows machines. The tool we used in this experiment includes NMAP, Netcat and Enum4Linux. NMAP is a popular open-source tool for reconnaissance. In this experiment, we used it to scan ports and services that ran on them and OS fingerprinting to determine the version of the operating systems; this will help deploy the correct exploit that works against

the operating system. Netcat was also used for port scanning during the experiment, and it can also be used during the lateral movement stage to establish a listener on a particular port. In the infiltration, we gained access to these systems using Metasploit and deploying exploits to access the Windows Systems. From there, we created accounts with admin privileges and performed actions such as internal reconnaissance for lateral movement. We also performed remote interactions with the target machines, which were compromised and copied infected files across to the systems and run them remotely to demonstrate the action of actual attackers. Once we got full control of these systems and performed the actions needed to achieve our goal, we proceeded to stop all the running services relating to the security monitoring tools and delete all the audit logs and system logs to hide our audit trail.

Before deleting the logs, we exported the Windows logs to capture all the triggered events during our experiments, especially during the active attack stages. We also collected alerts and other security information from our security Onion network monitoring tool. We obtained the alerts from other sources such as Squert and Kibana. Squert is a web-based portal used to query events stored in the Squil IDS database. Kibana provides visualisation for events from the logs. We used information from these sources and Windows Events to build and prepare our machine learning dataset.

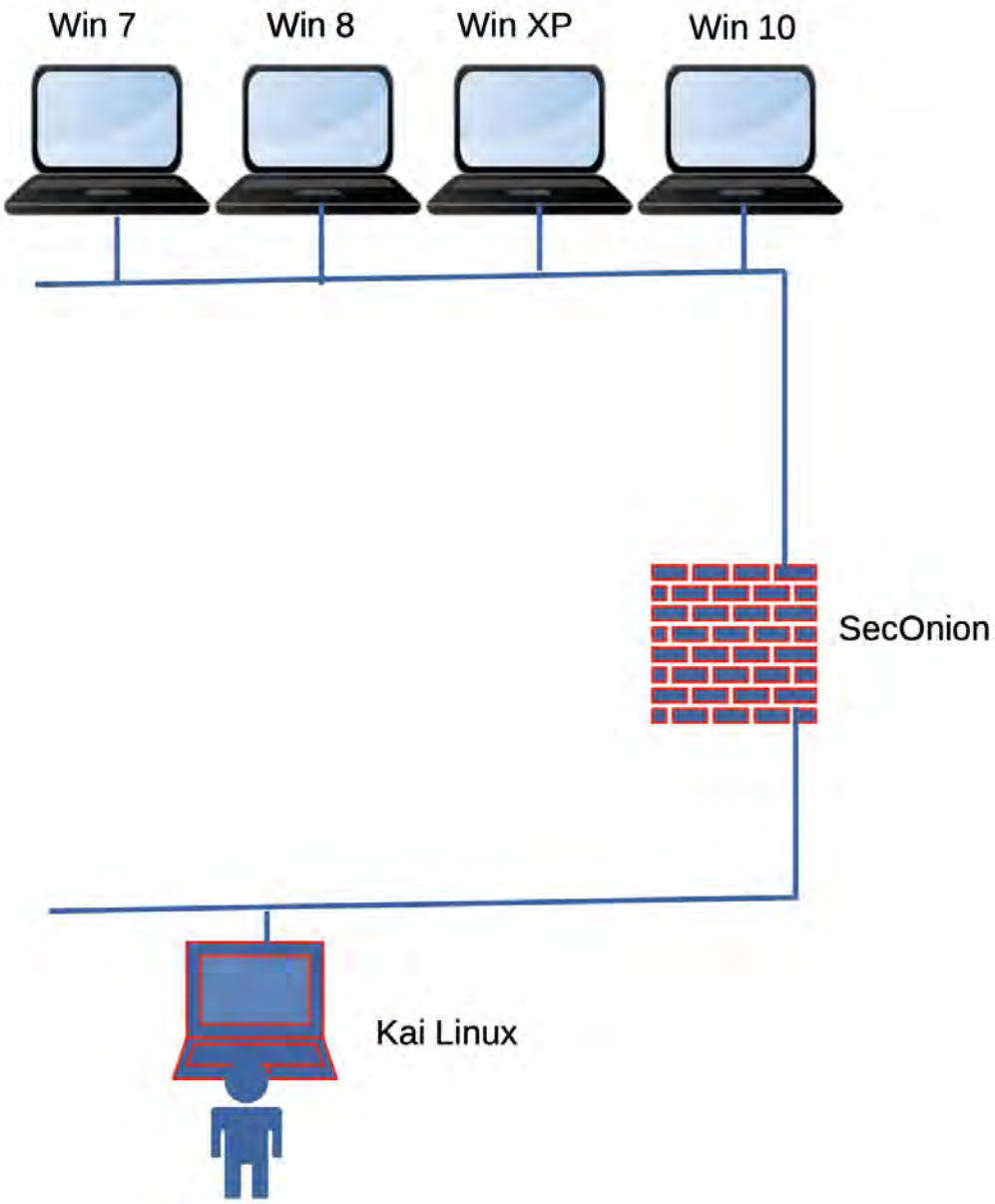

Figure 7.9: IoC Network Architecture Experiment II

7.6 Feature Selection

In the feature selection process, we followed same feature selection methods which were described in the section 7.2.1. We used InfoGain and GainRatio methods to select the most suitable features for our model until we were left with final seven features. Table 7.4 show the top seven selected feature for the two methods. Fig 7.12 shows the list of features used in the 2nd IoC experiment.

Table 7.4: Selected features from experiment II

Feature Selection Methods	**Selected top 7 Features**
InfoGain	9,12,13,15,17,33,39
GainRatio	8,10,15,22,30,32,33

7.7 Attack Classification

In the attack classification, we used the top features chosen during our feature selection process. We then used machine learning classifiers such as BayesNet, Naive Bayes, SVM, KNN and Random Forest. Table 7.5 shows the performance of the classifiers, with features from InfoGain performing better in the classification. We used performance metrics such as accuracy, precision, recall, false alarm rate, f-measure and Matthews correlation coefficient (MCC) to evaluate the performance of our model. Fig. 7.10 shows the

classifier's performance based on the Infogain selected features side by side.

Table 7.5: Performance results with Naive Bayes Classifier - Experiment II

Feature selection	**Acc**	**Precision**	**Recall**	**FAR**	**F_1**	**Mcc**
InfoGain	98.8%	98.9%	98.8%	0.2%	98.8%	98.7%
Gain Ratio	97.7%	97.8%	97.7%	0.5%	97.1%	96.9%

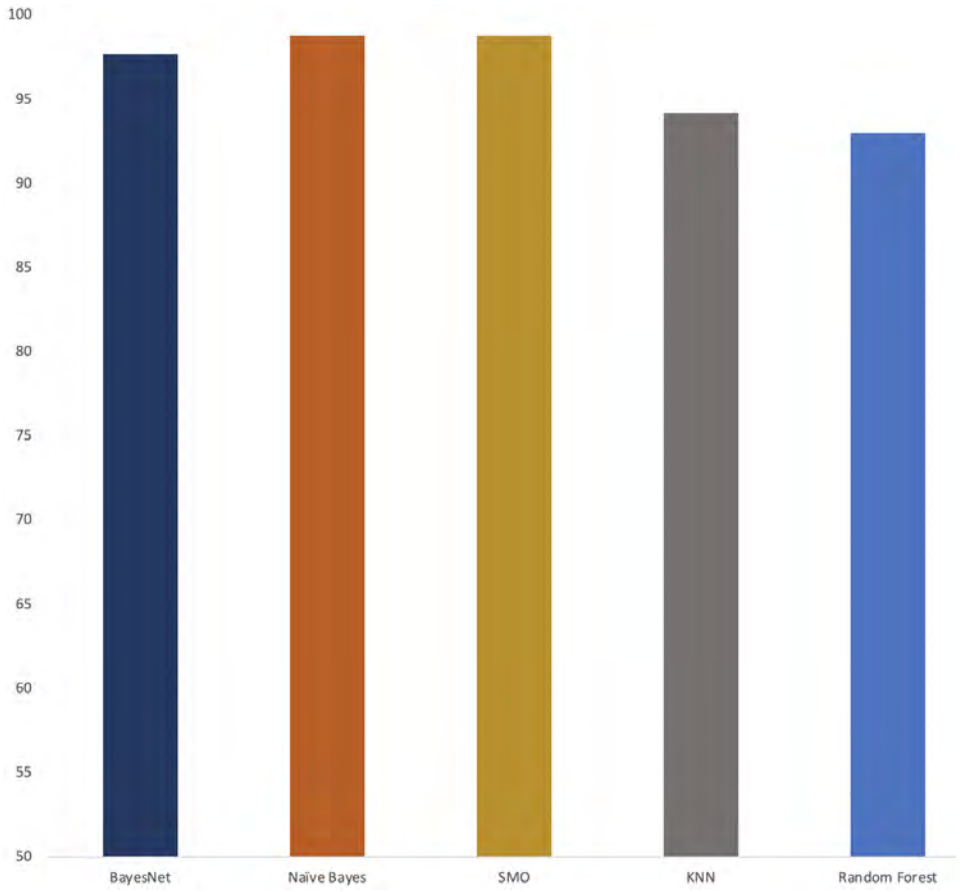

Figure 7.10: Info-Gain Top 7 Features-Experiment II

Label ID	Description
1	Normal
2	Directory Scanning Attack
3	SMB Attack
4	Privilege Escalation
5	Brute Force
6	Reconnaissance
7	Infiltration

Table 7.6: Expanded Attack Labels Experiment II

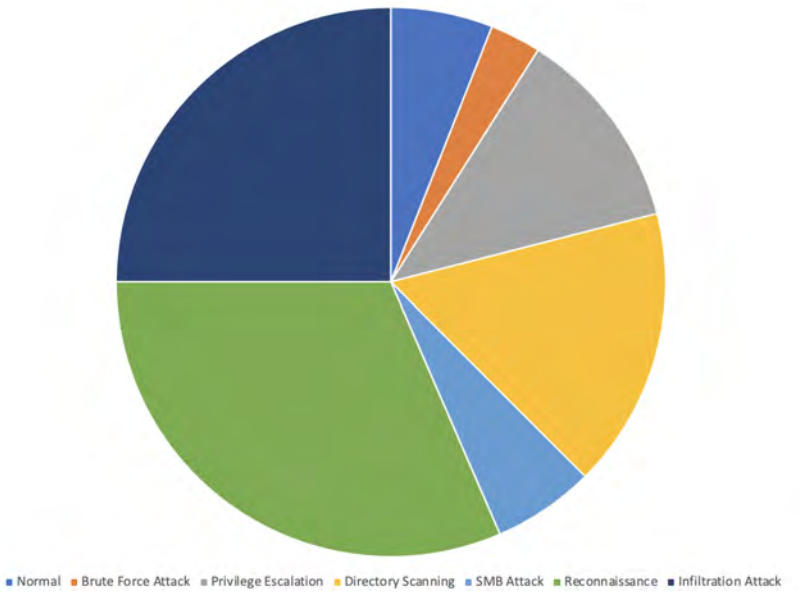

Figure 7.11: IoC Attack Label Distribution Experiment II

Attack	Tools	Attacker	Victim
Directory Scanning Attack	Dirb, Gobuster	Kali Linux	Win 7, Win 8
SMB Attack	Metasploit and Eternal Blue	Kali Linux	Win XP, 7, 8 and 10
Privilege Escalation	Metasploit, Eternal Blue, GetSystem , Mimikatz and Armitage	Kali Linux	Win 7, 8 and 10
Brute Force	Hydra	Kali Linux	Win 7, Win 8 and Win 10
Reconnaissance	NMAP, Netcat, Enum4Linux	Kali Linux	Win XP, 7, Win 8 and Win 10
Infiltration	Metasploit, Metsvc	Kali Linux	Win XP, 7, Win 8 and Win 10

Table 7.7: Attack Labels and Tools

Feature	Description	Feature	Description
Feat1	Source IP	Feat2	Destination IP
Feat3	Source Port	Feat4	Destination Port
Feat5	Total Login Attempts	Feat6	Successful Login
Feat7	Logon Type	Feat8	Byte Length
Feat9	Flags	Feat10	Protocol
Feat11	Operating System	Feat12	Hostname
Feat13	Event ID 4624	Feat14	Event ID 4672
Feat15	Event ID 4634	Feat16	Event ID 4720
Feat17	Event ID 4728	Feat18	Event ID 4688
Feat19	Event ID 4905	Feat20	Event ID 4722
Feat21	Event ID 1102	Feat22	Event ID 5033
Feat23	Event ID 4902	Feat24	Event ID 4907
Feat25	Event ID 5061	Feat26	Event ID 4738
Feat27	Event ID 4724	Feat28	Event ID 4732
Feat 29	Event ID 4625	Feat30	Event ID 5025
Feat31	Event ID 5058	Feat32	Event ID 517
Feat33	Event ID 4704	Feat34	Event ID 4678
Feat35	Event ID 4698	Feat36	Event ID 4700
Feat37	Event ID 4702	Feat38	Event ID 4670
Feat39	Event ID 4648	Feat40	Event ID 4740
Feat41	Event ID 4825	Feat42	Event ID 602
Label	Attack Types		

Figure 7.12: IoC Feature Description - Experiment II

7.8 Threat hunting example with Windows Events

Windows event ids play a vital role in threat hunting and help threat hunters to link together events that happened to determine whether a cyberattack

has taken place on their systems. This section discusses how Windows events IoCs could be collected from the target machines and demonstrate how these could lead us to the attack that may have remained undetected by other network security monitoring tools. In this example, we used a simple SMB attack to demonstrate the concept.

In the example, we assume the system has been exploited through SMB vulnerabilities on the target Windows systems. Attackers perform further activities once the systems are compromised, and these include privileges escalation and lateral movement. Here we explain possible Windows event ids that are triggered when such actions are executed and group them according to the actions performed by the attackers. We also perform some typical post exploitation steps performed on target machines and explain the various event ids that are triggered and link them to build possible attacks that may have taken place.

1. Target machines exploited through SMB vulnerabilities.
2. Gain access to the remote target systems
3. Perform privilege escalations
4. Perform interactive remote desktop connection to the target machines
5. Stop the firewall services
6. Re-enable and reset the password for an existing user
7. Perform lateral movements
8. Create scheduled tasks to install backdoor malware

9. Move files from the target machine (data filtration)

10. Clear security audit logs

11. Logs out from the session

In this example, we look at the Windows events IDs relating to this kind of attack and subsequent steps directed at the target systems. Fig. 7.13 shows the steps involved, and the relevant Windows IDs generated to piece the attack together. The process starts with an SMB attack that targets SMB vulnerabilities on Windows systems. SMB is a trusted protocol that enables computers to communicate, making it a popular target for cyber attackers. Once the target system is compromised, the attackers perform various actions that led to the remote host's Windows Event IDs. In this scenario, a user account was added to the remote hosts that generated Windows event IDs 4704, 4672, and 4722. On their own, each of the individual events might be harmless. However, when you are investigating incidents and notice an account that was just created and then assigned a special privilege as per event ID 4672, further checks such as the login id and details of the remote system used to create should be performed. Cyber attackers will often do this step in order to access restricted sources. Next, an existing account on the target system was re-enabled, and the password reset. These actions generate Windows event Ids 4722, 4738 and 4724. These actions demonstrated that an account was enabled and changes made to it. Cyber attackers usually prefer to use existing accounts instead of creating a new one which can generate some noise on the monitoring tools or picked through audits and new account request management process. In this case, it will also be good to link all these

event ids IoCs and see whether there is a possibility of a breach.

In the next step, a privilege escalation was performed to control these systems and perform lateral movement during the later stages. Commands such as getsystem can be used to gain system-level access during the privilege escalation, and many other tools are available for privilege escalations. Further actions such as creating accounts, giving them administrative access, and then logging with these credentials can be performed with privileged access. Regular user accounts have limitations on what the cyber attacker can do, hence why privilege escalation is necessary. In this scenario, the attackers then stopped the firewall service, which is not an action usually performed by standard users, and this action triggers Windows event id 5025. The person who stopped the firewall services must have administrative access, which demonstrates the attacker has already gained elevated privileges, and it is worth investigating it further. Although it is rare for attackers to perform a remote desktop connection to the victim machines, it is a possibility that was considered in this scenario. Initially, they might get a connection error that triggered Widow event id 4825, followed by re-enabling the remote desktop service and successfully log in, which generated event Id 4625. Again, most organisations will disable remote desktop by default due to the various vulnerabilities associated with it, and event id 4825 combined with a successful login (event id 4625) should trigger some investigations. A lateral movement can then be performed using tools such as Mimikatz and then run some programs which can trigger Window event ID 4688 and 4656. The event 4688 is generated when a new process is started and shows who started, the process path and what actions were taken. A scheduled task was then created to run

a program on the target machines, which triggers Windows event ids 4698, 602 and 4688. Attackers can use scheduled tasks to deployed malware on the target machine, and scheduled tasks need to be monitored. Finally, an attempt was made to delete the audit logs, which triggers event id 1102 and 517. This an anti-forensic action performed by skilled cybercriminals to hide their tracks.

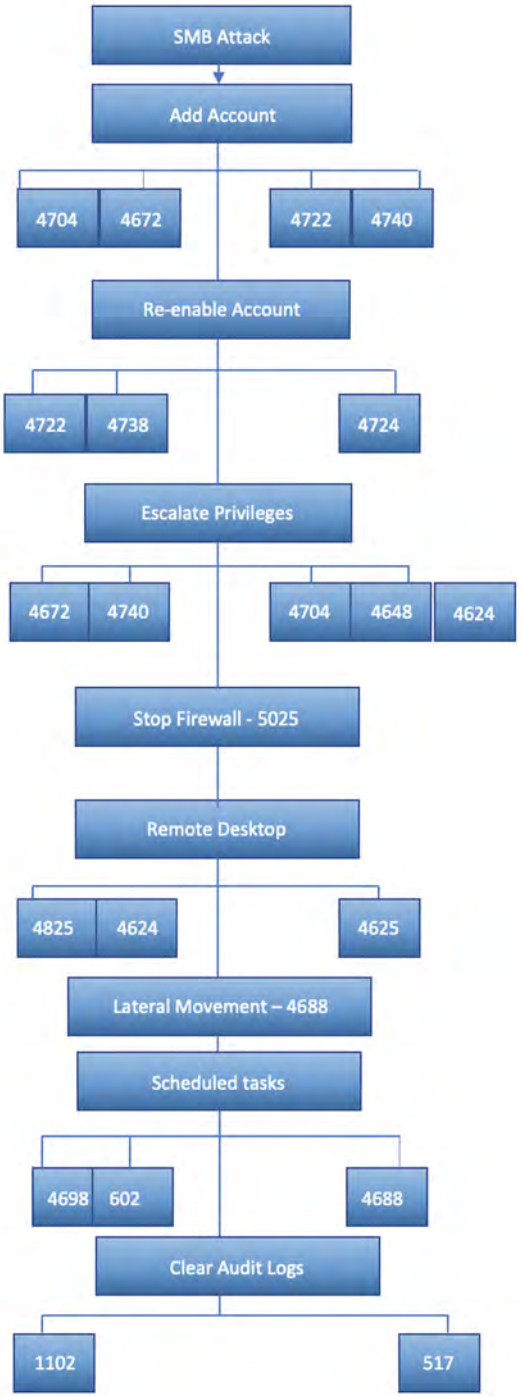

Figure 7.13: Windows Event-ID Based Scenario

7.8.1 Scenario Summary

This section demonstrates the concept of using Windows event ids IoCs to build the attacker's action to compromise systems and determine the various steps involved. There are many Windows event logs, and piecing this together is not an easy task. Most of the event ids are harmless on their own but can be used to build a picture of the attack once the various events are combined. There are always some chances of false positives, and it is always a good idea to crosscheck the Windows events with alerts from other monitoring tools if they exist to ensure the events are malicious.

7.9 Analysis and Discussion - Experiment II

In this second experiment, we expanded on the first experiment to improve our work. We added two other attack labels to the dataset, namely reconnaissance and infiltration, making seven labels as shown in Fig. 7.6. The distribution of the attack is in Fig. 7.11.

We increased the features and observations to make the data more representative. We used four Windows machines and a Kali Linux attack system in our experiment and then collected the data, including alerts, to prepare the new machine learning dataset. In the dataset, we performed feature selection to choose the best features for our model. The list of features selected through InfoGain and GainRatio are in Table 7.4.

We then performed machine learning classification using Naive Bayes, BayesNet, KNN, SVM and Random Forest. The result of the classification using the Naive Bayes classifier is shown in Table 7.5. Looking at the results,

we can see that the features from InfoGain performed better than those from GainRatio using the Naive Bayes classifier. We used performance metrics such as accuracy, precision, recall, f-measure and Mcc to evaluate our model's effectiveness. We obtained the best performance accuracy of 98.8%, which is an improvement from the first experiment where we obtained an overall score of 96.7% with the feature from GainRatio. Overall the features from InfoGain performed better on all six metrics. Fig. 7.10 shows the top seven features from InfoGain using the Naive Bayes classifier performed in experiment two. We also demonstrated a scenario on threat hunting based on Windows event ids and utilising SMB attacks, as shown in Section 7.8

7.10 Summary

This chapter demonstrated IoC's role in threat hunting and how various events can be used to reconstruct the attacker's digital trail. In the first experiment, we constructed a dataset that contained 29 features and 87 observations. Although we achieved good results from this work and achieved an overall accuracy of 96.7% for our model, we decided to improve this work. We created a new testbed and expanded on the machine learning dataset features to 42 features and 215 observation, which removed the limitations of the original dataset. We compared our model's performance based on the two experiments. The newer model from Experiment II produced a better classification accuracy of 98.8% with features from InfoGain. We believe our work, including the rigorous analysis of the results, will contribute to the effective detection of IoCs and reduce data breaches by allowing system ad-

ministrators to implement proactive approaches to deal with IoC that may be hidden or remained undetected in the system or network. In our future work, we intend to expand on this work and create a large dataset consisting of both indicator of compromise (IOC) and indicator of attacks (IOA) to help improve detection of cyber breaches or prevent such attacks through proactive approaches.

Chapter 8

Conclusions and Future Work

8.1 Conclusion

Cybercriminals are using advanced techniques to bypass the security defences and compromise secure networks. The availability of highly sophisticated tool has enabled cybercriminals to execute attacks with ease. The motivation of these attackers vary and range from financial gains, intellectual property theft to espionage. We discussed several APT groups in section two that were attributed to nation-state actors. However, there are also other groups driven by criminal activities serving their interests. These criminal entities include those driven by financial gains who sell and share stolen data on the dark web and damage the reputations of the affected organisations. That is one reason why there are substantial security investments to protect critical systems and the data hold. There are also data protection regulations that mandate the protection of confidential personal data, such as the EU General Data Protection Regulations and the UK Data Protection Act 2018. These

data breaches attract the regulators' attention.

Despite all these efforts, cybercriminals are finding ways to go beyond the security defences. Most of the current solutions are reactive and send alerts to system administrators when anomalies are detected, but these alerts can be overwhelming for the analyst and require triaging. There are also many false positives and negatives generated by these systems, which can result in important alerts being missed. Although some of these tools, such as antivirus, intrusion detection systems and firewalls, detect cyber threats and prevent cyber attacks, their performance against complex attacks such as advanced persistent threats has not been consistent, given APT attacks take on average 56 days to be detected [80]. It is in this backdrop that we are aiming to propose a framework for security monitoring of network systems that demonstrate the capability to detect cyber threats with minimal false positives.

Our proposed framework consists of three key attacks detection components, and systems administrators can create a module for each to detect cyber threats. This framework is data-driven, and the focus is mainly on the machine learning aspect. The first components is a machine learning model for detecting sophisticated attacks such as APTs. In this model, we built on the work by the APT dataset provider [5] and improve on the machine learning aspect of their work. We used some of their APT stage detection modules and proposed our modules as shown in Fig. 5.3. We reconstructed the dataset, performed feature extraction and selection to improve the model efficiency. We leveraged the Cyber Kill Chain (CKC) approach and mapped the detection modules to the CKC stages. The CKC is a well-known indus-

try informed model that describes the various phases of a cyber attack. We set the threshold of a satisfactory outcome to be 84.9% for the prediction accuracy based on the original APT dataset provider's work. We achieved 91.1% in our work which was more than the threshold. This work was published in a peer-reviewed journal [32]. To expand on the feature selection techniques, I contributed to another paper that was also published in a peer-reviewed Journal [216]. This paper was on "Effective combining of feature selection techniques for machine learning-enabled IoT intrusion detection", and most of my individual contribution was on the feature selection methods. We utilised feature extraction using deep learning techniques and then performed feature selection using the wrapper method using three different classifiers. We then combined the features, selected the top 20, and obtained an overall detection accuracy of 99.5%. We hope our proposed model and the Cyber Kill approach that was utilised will help detect APT attacks and reduce the cost of data breaches. Furthermore, the feature selection and extraction concept explained in the second paper can also be applied to APT stage detection in the context of the Internet of Things (IoT).

The second component of the framework is cyber attack prediction using machine learning-enabled feature forecasting. Given the sophistication of the cyber attacks and the limitations of the reactive approaches, we proposed the cyber attack prediction model, which can be used to detect cyber attacks and provide a time window which system administrators can use to implement proactive approaches to mitigate the predicted risks before the cybercriminal exploit these vulnerabilities. Although time series forecasting has been used widely in weather and stock prediction, it has not been widely explored to

predict cyber events. Our prediction model will look at patterns of cyber events to predict the cyber attacks. We utilised an open dataset from a realistic secure network with various departments consisting of 450 machines victim machines, including servers and 50 attack machines. The dataset contained seven different attack types and benign traffic. We reconstructed the dataset and created equal portions using time intervals. To evaluate this model, we used performance metrics and obtained a top accuracy of 90.4%. We also used the metric Mean Absolute Error (MAE) to evaluate the time series data's performance. The results showed that the forecasted features from linear regression produced the best results with a low MAE compared to the SMOreg features. We believe this prediction model will reduce cyber breaches by predicting the attacks and allowing systems administrators time to implement proactive approaches.

This framework's third machine learning component is threat detection using Indicators of Compromise(IoC). In this component, we created our experiments and collected data to create the dataset. IoCs are artefacts left behind following an attack, which can be used to build the attacker picture. Several high profile attacks have been detected through IoC. The security community are sharing IoC to help the mitigate risk associated with these threats. There is limited dataset availability for host IoCs due to fear of litigation which necessitated us to create our dataset. We created two experiments related to IoC detection in hosts. In the first experiment, we achieved an accuracy of 96.7%, and the work was submitted to a peer-reviewed book chapter [149]. We followed this up with another experiment for IoC detection to deal with the limitations of the first experiment. We

expanded on the features, observation and attack labels and performed a rigorous analysis of the results. In the second experiment, we achieved an accuracy result of 98.8%.

Finally, we have the visualisation charts for decision-making. The charts displays the results from the machine learning models, such as accuracy and false alarm rates, which can assist with determining how well the security controls are performing. The visualisation can be performed using third-party tools such as Excel. Systems administrators can build detection modules for each of these models and feed the results to the charts.

In the framework, we choose these three machine learning models, which are (i) cyber attack detection, (ii) prediction using time series enabled feature forecasting, (iii) IoC detections. There is a need for a solution that can detect attacks, predict cyber events, and capture IoCs using the attackers' artefacts and trails. We believe the framework we are proposing will capture these three elements and reduce cyber breaches. Systems administrators and decision-makers can use this framework to determine how well their security controls are performing by looking at their detection accuracy based on the models. Although time series based cyber event forecasting received little attention from researchers, we believe our work will contribute to the advancement of this domain in terms of cyberattack prediction. The IoC element can be used to detect attacks missed by the security monitoring tools. The IoCs that were detected can then be fed to the security monitoring tool to detect similar attacks in the futures. In this work, we achieved high accuracy results with minimal false alarm rate.

8.2 Limitation of this research

In this research, four datasets were used, of which two were publicly available open datasets and the others derived from our own experiments. The two open datasets were selected based on their quality and alignment to the research performed in this thesis and to help answer the research questions and meet the objectives. Although these datasets were enough to fulfil our requirement, there were some limitations. In the APT dataset, there were only 8 features, 3676 observation and a label which meant the features were limited. We overcame this challenge by performing feature extraction using Multiple Factor Analysis (MFA). The CSE-CIC2018 IDS dataset was more comprehensive and contained 79 features and large observations, but the time window was very short, which meant that when we performed resampling based on a 30-second interval, the number of observations was substantially reduced, which resulted in our attack prediction window to be in hours due to the dataset's constraints. However, we believe this is a reasonable time to take corrective measures given the evolving threat landscape. We plan to expand on this work in our future work and increase the forecasting window to between 1 to 7 days.

8.3 Future Work

In this research, we proposed a framework for the security monitoring of networked systems. We leveraged machine learning to create our models, which we believe will reduce cyber breaches through accurate detection and prediction of cyber attacks. Through this research, we identified several areas

of future research and these include:

- Applying our proposed APT detection in the context of the Internet of Things (IoT.) Our proposed framework was more generic and could be applied in various setups but these could be extended and applied in the context of IoT. The proposed detection modules could particularly be applied in the IoT environment to detect sophisticated threats targetting IoT ecosystems. The threat from IoT devices will continue to increase due to the sheer number of devices deployed across organisations and their associated vulnerabilities.

- Correlating Indicator of compromise and indicators of attacks. Indicators of compromise are forensic artefacts left behind by attackers, while the indicators of attacks are malicious behaviours observed in live production environments that could indicate an attack in progress. Correlating these two indicators and how they can influence each other is a possible area of future research.

- Expanding on the prediction windows to days and weeks. In this research, our forecasted cyber events were in hours due to the dataset constraints and expanding these forecasted cyber events to days and weeks could be an option. Comparing the results and performance from these time window could be an area of further research. Given the rapidly changing threat landscape, we believe prediction in hours is the best option, especially if the organisation in question has enough resources. However, exploring the forecasted features in days and weeks might also be feasible for other organisations, hence why we will follow

this in our feature research. The challenge with longer prediction times is the complexity of the data needed to make an informed decision and the chances of the predicted attack materialising before corrective actions can occur. This is due to the dynamic nature of the threat landscape, making such predictions redundant, especially if they are based on weeks and over.

- Build more comprehensive detection modules, including those from other challenging attack types such as fileless malware. In this research, we built on some existing detection modules and added our own proposed modules to detect APT attacks. We believe expanding these attack modules to include other sophisticated attacks such as fileless malware and complex ransomware attacks will contribute to advances in detecting the highly complex attacks that are difficult to detect. For example, fileless malware is executed in memory and often embedded in trusted applications such as PowerShell, making it difficult to be detected by the security monitoring systems. This is an area we are planning to explore in our future work.

- Expanding the IoC dataset. Currently, there are limited IoC datasets relating to host systems, and organisations are not sharing these data for fear of litigations. Creating a large dataset that brings together both host and network IoCs and determining how they inform each other could be an area of future research. We plan to explore this further in our future work to build large networks to collect such data from the hosts and network devices.

Bibliography

[1] Harjinder Singh Lallie, Lynsay A Shepherd, Jason RC Nurse, Arnau Erola, Gregory Epiphaniou, Carsten Maple, and Xavier Bellekens. Cyber security in the age of covid-19: A timeline and analysis of cyber-crime and cyber-attacks during the pandemic. *arXiv preprint arXiv:2006.11929*, 2020.

[2] Chanchala Joshi and Umesh Kumar Singh. Information security risks management framework–a step towards mitigating security risks in university network. *Journal of Information Security and Applications*, 35:128–137, 2017.

[3] Ipsos MORI. Cyber security breaches survey 2020. Survey report, The Department for Digital, Culture, Media and Sport, March 2020.

[4] I. Ponemon. Cost of a data breach report," ibm technical report, 2020. Report 17, IBM, IBM Corporation New Orchard Road Armonk, NY 10504, July 2021.

[5] Ibrahim Ghafir, Mohammad Hammoudeh, Vaclav Prenosil, Liangxiu Han, Robert Hegarty, Khaled Rabie, and Francisco J. Aparicio-Navarro. Detection of advanced persistent threat using machine-learning correlation analysis. *Future Generation Computer Systems*, 89:349–359, 2018.

[6] Kelly Fiveash. Playstation clambers back online days after ddos attack paralysed network, December 2014.

[7] ENISA. Enisa threat landscape 15 top threats in 2020. Report, ENISA, October 2020.

[8] Kang Leng Chiew, Kelvin Sheng Chek Yong, and Choon Lin Tan. A survey of phishing attacks: Their types, vectors and technical approaches. *Expert Systems with Applications*, 106:1–20, 2018.

[9] Issa Qabajeh, Fadi Thabtah, and Francisco Chiclana. A recent review of conventional vs. automated cybersecurity anti-phishing techniques. *Computer Science Review*, 29:44–55, 2018.

[10] Ankit Kumar Jain and Brij B Gupta. A machine learning based approach for phishing detection using hyperlinks information. *Journal of Ambient Intelligence and Humanized Computing*, 10(5):2015–2028, 2019.

[11] Ozgur Koray Sahingoz, Ebubekir Buber, Onder Demir, and Banu Diri. Machine learning based phishing detection from urls. *Expert Systems with Applications*, 117:345–357, 2019.

[12] Kang Leng Chiew, Choon Lin Tan, KokSheik Wong, Kelvin SC Yong, and Wei King Tiong. A new hybrid ensemble feature selection framework for machine learning-based phishing detection system. *Information Sciences*, 484:153–166, 2019.

[13] Centrify. Remote working has increased the risk of a cyber breach, says three quaters of uk businesses, April 2020.

[14] Chen Han and Rituja Dongre. Q&a. what motivates cyber-attackers? *Technology innovation management review*, 4(10), 2014.

[15] Nidal Hassan Hussein and Ahmed Khalid. A survey of cloud computing security challenges and solutions. *International Journal of Computer Science and Information Security*, 14(1):52, 2016.

[16] P Ravi Kumar, P Herbert Raj, and P Jelciana. Exploring data security issues and solutions in cloud computing. *Procedia Computer Science*, 125:691–697, 2018.

[17] Suhail Qadir and SMK Quadri. Information availability: An insight into the most important attribute of information security. *Journal of Information Security*, 7(3):185–194, 2016.

[18] Ahmad Javaid, Quamar Niyaz, Weiqing Sun, and Mansoor Alam. A deep learning approach for network intrusion detection system. In *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)*, pages 21–26, 2016.

[19] Rafath Samrin and D Vasumathi. Review on anomaly based network intrusion detection system. In *2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT)*, pages 141–147. IEEE, 2017.

[20] Chirag Modi, Dhiren Patel, Bhavesh Borisaniya, Hiren Patel, Avi Patel, and Muttukrishnan Rajarajan. A survey of intrusion detection techniques in cloud. *Journal of network and computer applications*, 36(1):42–57, 2013.

[21] Ansam Khraisat, Iqbal Gondal, Peter Vamplew, and Joarder Kamruzzaman. Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity*, 2(1):1–22, 2019.

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

[23] Deris Stiawan, Abdul Hanan Abdullah, and Mohd Yazid Idris. The trends of intrusion prevention system network. In *2010 2nd International Conference on Education Technology and Computer*, volume 4, pages V4–217. IEEE, 2010.

[24] W. Seo and W. Pak. Real-time network intrusion prevention system based on hybrid machine learning. *IEEE Access*, 9:46386–46397, 2021.

[25] A. Krishna, A. Lal M.A., A. J. Mathewkutty, D. S. Jacob, and M. Hari. Intrusion detection and prevention system using deep learning. In *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*, pages 273–278, 2020.

[26] Jin Kim, Nara Shin, Seung Yeon Jo, and Sang Hyun Kim. Method of intrusion detection using deep neural network. In *2017 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 313–316. IEEE, 2017.

[27] Radwan Tahboub and Yousef Saleh. Data leakage/loss prevention systems (dlp). In *2014 World Congress on Computer Applications and Information Systems (WCCAIS)*, pages 1–6. IEEE, 2014.

[28] Shumeet Baluja. Hiding images in plain sight: Deep steganography. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 2066–2076, 2017.

[29] John Stevenson. Attack using steganography bypasses dlp systems, November 2018.

[30] Mamta Jain and Saroj Kumar Lenka. A review on data leakage prevention using image steganography. *International Journal of Computer Science Engineering*, 5(2):56–59, 2016.

[31] Jinhyung Kim, Choonsik Park, Jun Hwang, and Hyung-Jong Kim. Privacy level indicating data leakage prevention system. *KSII Transactions on Internet and Information Systems (TIIS)*, 7(3):558–575, 2013.

[32] Yussuf Ahmed, A.Taufiq Asyhari, and Md Arafatur Rahman. A cyber kill chain approach for detecting advanced persistent threats. *Computers, Materials & Continua*, 67(2):2497–2513, 2021.

[33] Nutan Farah Haq, Abdur Rahman Onik, Md Avishek Khan Hridoy, Musharrat Rafni, Faisal Muhammad Shah, and Dewan Md Farid. Application of machine learning approaches in intrusion detection system: a survey. *IJARAI-International Journal of Advanced Research in Artificial Intelligence*, 4(3):9–18, 2015.

[34] CSE & CIC. A realistic cyber defense dataset (cse-cic-ids2018), 2018.

[35] 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.

[36] Majjed Al-Qatf, Yu Lasheng, Mohammed Al-Habib, and Kamal Al-Sabahi. Deep learning approach combining sparse autoencoder with svm for network intrusion detection. *IEEE Access*, 6:52843–52856, 2018.

[37] Ravi Vinayakumar, Mamoun Alazab, KP Soman, Prabaharan Poornachandran, Ameer Al-Nemrat, and Sitalakshmi Venkatraman. Deep learning approach for intelligent intrusion detection system. *IEEE Access*, 7:41525–41550, 2019.

[38] Hongyu Liu and Bo Lang. Machine learning and deep learning methods for intrusion detection systems: A survey. *applied sciences*, 9(20):4396, 2019.

[39] Kelton AP da Costa, João P Papa, Celso O Lisboa, Roberto Munoz, and Victor Hugo C de Albuquerque. Internet of things: A survey on machine learning-based intrusion detection approaches. *Computer Networks*, 151:147–157, 2019.

[40] Preeti Mishra, Vijay Varadharajan, Uday Tupakula, and Emmanuel S Pilli. A detailed investigation and analysis of using machine learning techniques for intrusion detection. *IEEE Communications Surveys & Tutorials*, 21(1):686–728, 2018.

[41] Chris Sinclair, Lyn Pierce, and Sara Matzner. An application of machine learning to network intrusion detection. In *Proceedings 15th Annual Computer Security Applications Conference (ACSAC'99)*, pages 371–377. IEEE, 1999.

[42] Alice Zheng and Amanda Casari. *Feature engineering for machine learning: principles and techniques for data scientists.* " O'Reilly Media, Inc.", 2018.

[43] Fatemeh Nargesian, Horst Samulowitz, Udayan Khurana, Elias B Khalil, and Deepak S Turaga. Learning feature engineering for classification. In *Ijcai*, pages 2529–2535, 2017.

[44] Sydney Mambwe Kasongo and Yanxia Sun. A deep learning method with filter based feature engineering for wireless intrusion detection system. *IEEE Access*, 7:38597–38607, 2019.

[45] Zhihua Ma and Guanghui Chen. Bayesian methods for dealing with missing data problems. *Journal of the Korean Statistical Society*, 47:297–313, 2018.

[46] Chih-Yu Hsu, Shuai Wang, and Yu Qiao. Intrusion detection by machine learning for multimedia platform. *Multimedia Tools and Applications*, 80(19):29643–29656, 2021.

[47] Fabio Caraffini. The naive bayes learning algorithm. 2019.

[48] Himani Bhavsar and Mahesh H Panchal. A review on support vector machine for data classification. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, 1(10):185–189, 2012.

[49] Haneen Arafat Abu Alfeilat, Ahmad BA Hassanat, Omar Lasassmeh, Ahmad S Tarawneh, Mahmoud Bashir Alhasanat, Hamzeh S Eyal Salman, and VB Surya Prasath. Effects of distance measure choice on k-nearest neighbor classifier performance: a review. *Big data*, 7(4):221–248, 2019.

[50] Jehad Ali, Rehanullah Khan, Nasir Ahmad, and Imran Maqsood. Random forests and decision trees. *International Journal of Computer Science Issues (IJCSI)*, 9(5):272, 2012.

[51] Musab Coşkun, Özal YILDIRIM, UÇAR Ayşegül, and Yakup Demir. An overview of popular deep learning methods. *European Journal of Technique (EJT)*, 7(2):165–176, 2017.

[52] Shih-Chung B Lo, Heang-Ping Chan, Jyh-Shyan Lin, Huai Li, Matthew T Freedman, and Seong K Mun. Artificial convolution neural network for medical image pattern recognition. *Neural networks*, 8(7-8):1201–1214, 1995.

[53] Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Geert Litjens, Paul Gerke, Colin Jacobs, Sarah J Van Riel, Mathilde Marie Winkler Wille, Matiullah Naqibullah, Clara I Sánchez, and Bram Van Ginneken. Pulmonary nodule detection in ct images: false positive reduction using multi-view convolutional networks. *IEEE transactions on medical imaging*, 35(5):1160–1169, 2016.

[54] Subhashree Mohapatra, Tripti Swarnkar, and Jayashankar Das. Deep convolutional neural network in medical image processing. In *Handbook of Deep Learning in Biomedical Engineering*, pages 25–60. Elsevier, 2021.

[55] Kenji Suzuki. Overview of deep learning in medical imaging. *Radiological physics and technology*, 10(3):257–273, 2017.

[56] S Deepthi, K Sandeep, and L Suresh. Detection and classification of objects in satellite images using custom cnn. *International Journal of Engineering Research & Technology*, 10(6):629–635, 2021.

[57] Arshitha Femin and KS Biju. Accurate detection of buildings from satellite images using cnn. In *2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*, pages 1–5. IEEE, 2020.

[58] Muhammad Jaleed Khan, Adeel Yousaf, Nizwa Javed, Shifa Nadeem, and Khurram Khurshid. Automatic target detection in satellite images using deep learning. *Journal of Space Technology*, 7(1):44–49, 2017.

[59] Xuan Zhang, Xun Liang, Aakas Zhiyuli, Shusen Zhang, Rui Xu, and Bo Wu. At-lstm: An attention-based lstm model for financial time series prediction. In *IOP Conference Series: Materials Science and Engineering*, volume 569, page 052037. IOP Publishing, 2019.

[60] Refat Khan Pathan, Munmun Biswas, and Mayeen Uddin Khandaker. Time series prediction of covid-19 by mutation rate analysis using recurrent neural network-based lstm model. *Chaos, Solitons & Fractals*, 138:110018, 2020.

[61] Peipei Wang, Xinqi Zheng, Gang Ai, Dongya Liu, and Bangren Zhu. Time series prediction for the epidemic trends of covid-19 using the improved lstm deep learning method: Case studies in russia, peru and iran. *Chaos, Solitons & Fractals*, 140:110214, 2020.

[62] Chen Lin, Yuan Zhang, Julie Ivy, Muge Capan, Ryan Arnold, Jeanne M Huddleston, and Min Chi. Early diagnosis and prediction of sepsis shock by combining static and dynamic information using convolutional-lstm. In *2018 IEEE International Conference on Healthcare Informatics (ICHI)*, pages 219–228. IEEE, 2018.

[63] Zahra Karevan and Johan AK Suykens. Transductive lstm for time-series prediction: An application to weather forecasting. *Neural Networks*, 125:1–9, 2020.

[64] Idil Sülo, Seref Recep Keskin, Gülüstan Dogan, and Theodore Brown. Energy efficient smart buildings: Lstm neural networks for time series prediction. In *2019 International Conference on Deep Learning and Machine Learning in Emerging Applications (Deep-ML)*, pages 18–22, 2019.

[65] Ioannis E Livieris, Emmanuel Pintelas, and Panagiotis Pintelas. A cnn–lstm model for gold price time-series forecasting. *Neural computing and applications*, 32(23):17351–17360, 2020.

[66] Sara A Althubiti, Eric Marcell Jones, and Kaushik Roy. Lstm for anomaly-based network intrusion detection. In *2018 28th International telecommunication networks and applications conference (ITNAC)*, pages 1–3. IEEE, 2018.

[67] Abhishek Verma and Virender Ranga. Statistical analysis of cidds-001 dataset for network intrusion detection systems using distance-based machine learning. *Procedia Computer Science*, 125:709–716, 2018.

[68] Pengfei Sun, Pengju Liu, Qi Li, Chenxi Liu, Xiangling Lu, Ruochen Hao, and Jinpeng Chen. Dl-ids: extracting features using cnn-lstm hybrid network for intrusion detection system. *Security and Communication Networks*, 2020, 2020.

[69] Yakubu Imrana, Yanping Xiang, Liaqat Ali, and Zaharawu Abdul-Rauf. A bidirectional lstm deep learning approach for intrusion detection. *Expert Systems with Applications*, 185:115524, 2021.

[70] Yan Li and Yifei Lu. Lstm-ba: Ddos detection approach combining lstm and bayes. In *2019 Seventh International Conference on Advanced Cloud and Big Data (CBD)*, pages 180–185. IEEE, 2019.

[71] Arunavo Dey. Deep ids: A deep learning approach for intrusion detection based on ids 2018. In *2020 2nd International Conference on Sustainable Technologies for Industry 4.0 (STI)*, pages 1–5. IEEE, 2020.

[72] Rawaa Ismael Farhan, TM Abeer, and FH Nidaa. Performance analysis of flow-based attacks detection on cse-cic-ids2018 dataset using deep learning. *Indones. J. Electr. Eng. Comput. Sci*, 20:16–27, 2020.

[73] Mohamed Amine Ferrag, Leandros Maglaras, Helge Janicke, and Richard Smith. Deep learning techniques for cyber security intrusion detection: A detailed analysis. In *6th International Symposium for ICS & SCADA Cyber Security Research 2019 6*, pages 126–136, 2019.

[74] Fahimeh Farahnakian and Jukka Heikkonen. A deep auto-encoder based approach for intrusion detection system. In *2018 20th International Conference on Advanced Communication Technology (ICACT)*, pages 178–183. IEEE, 2018.

[75] Yisroel Mirsky, Tomer Doitshman, Yuval Elovici, and Asaf Shabtai. Kitsune: an ensemble of autoencoders for online network intrusion detection. *arXiv preprint arXiv:1802.09089*, 2018.

[76] Giuseppina Andresini, Annalisa Appice, and Donato Malerba. Autoencoder-based deep metric learning for network intrusion detection. *Information Sciences*, 569:706–727, 2021.

[77] Yesi Novaria Kunang, Siti Nurmaini, Deris Stiawan, Ahmad Zarkasi, et al. Automatic features extraction using autoencoder in intrusion detection system. In *2018 International Conference on Electrical Engineering and Computer Science (ICECOS)*, pages 219–224. IEEE, 2018.

[78] Hyunseung Choi, Mintae Kim, Gyubok Lee, and Wooju Kim. Unsupervised learning approach for network intrusion detection system using autoencoders. *The Journal of Supercomputing*, 75(9):5597–5621, 2019.

[79] Atif Ahmad, Jeb Webb, Kevin C Desouza, and James Boorman. Strategically-motivated advanced persistent threat: Definition, process, tactics and a disinformation model of counterattack. *Computers & Security*, 86:402–418, 2019.

[80] Jurgen Kutscher, September 2020.

[81] Kaspersky. Advanced persistent threats in 2021: new threat angles and attack strategy changes are coming. Report 1, Kaspersky, November 2020.

[82] Verizon. Data breach investigation report. Report 1, Verizon, September 2020.

[83] Malwarebytes. Apts and covid-19: How advanced persistent threats use the coronavirus as a lure. Report 1, Malwarebytes, April 2020.

[84] Global Research and Analysis Team (GReAT). Apt trends report q1 2020 apt trend report 2020. Report 1, Kaspersky, April 2020.

[85] Sergio Caltagirone, Andrew Pendergast, and Christopher Betz. The diamond model of intrusion analysis. Technical report, Center For Cyber Intelligence Analysis and Threat Research Hanover Md, 2013.

[86] Dimitar Kostadinov. The cyber exploitation life cycle, March 2013.

[87] MITRE. Mitre att&ck, 2020.

[88] The search engine of internet-connected devices, April 2021.

[89] Olivier Thonnard, Leyla Bilge, Gavin O'Gorman, Seán Kiernan, and Martin Lee. Industrial espionage and targeted attacks: Understanding the characteristics of an escalating threat. In *International workshop on recent advances in intrusion detection*, pages 64–85. Springer, 2012.

[90] FireEye, March 2021.

[91] Kaspersky, 2019.

[92] Kaspersky, August 2019.

[93] Alex Tarter and Gurbir Singh, 09 2019.

[94] Ionut Ilascu. China apt hackers move to ransomware attacks, January 2021.

[95] FireEuye. Apt28: A window into russia's cyber espionage operations?, October 2014.

[96] ENISA. Main incidents in the eu and worldwide. 1, ENISA, April 2020.

[97] Gareth Corfield. From the crew behind the sony pictures hack comes operation interception: An aerospace cyber-attack thriller, June 2020.

[98] US Department of Treasury. Treasury sanctions north korean state-sponsored malicious cyber groups, September 2019.

[99] Stephan Haggard and Jon R Lindsay. North korea and the sony hack: Exporting instability through cyberspace. 2015.

[100] Benjamin Jensen, Brandon Valeriano, and Ryan Maness. Fancy bears and digital trolls: Cyber strategy with a russian twist. *Journal of Strategic Studies*, 42(2):212–234, 2019.

[101] Sergiu Gatlan. Nation-backed hackers spread crimson rat via coronavirus phishing, May 2020.

[102] Adam Meyers, February 2019.

[103] Adel Alshamrani, Sowmya Myneni, Ankur Chowdhary, and Dijiang Huang. A survey on advanced persistent threats: Techniques, solutions, challenges,

and research opportunities. *IEEE Communications Surveys & Tutorials*, 21(2):1851–1877, 2019.

[104] Antoine Lemay, Joan Calvet, François Menet, and José M Fernandez. Survey of publicly available reports on advanced persistent threat actors. *Computers & Security*, 72:26–59, 2018.

[105] Meicong Li, Wei Huang, Yongbin Wang, Wenqing Fan, and Jianfang Li. The study of apt attack stage model. In *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*, pages 1–5. IEEE, 2016.

[106] Tarun Yadav and Arvind Mallari Rao. Technical aspects of cyber kill chain. In *International Symposium on Security in Computing and Communication*, pages 438–452. Springer, 2015.

[107] Dennis Kiwia, Ali Dehghantanha, Kim-Kwang Raymond Choo, and Jim Slaughter. A cyber kill chain based taxonomy of banking trojans for evolutionary computational intelligence. *Journal of computational science*, 27:394–409, 2018.

[108] Pooneh Nikkhah Bahrami, Ali Dehghantanha, Tooska Dargahi, Reza M Parizi, Kim-Kwang Raymond Choo, and Hamid HS Javadi. Cyber kill chain-based taxonomy of advanced persistent threat actors: analogy of tactics, techniques, and procedures. *Journal of Information Processing Systems*, 15(4):865–889, 2019.

[109] Sana Siddiqui, Muhammad Salman Khan, Ken Ferens, and Witold Kinsner. Detecting advanced persistent threats using fractal dimension based machine learning classification. In *Proceedings of the 2016 ACM on international workshop on security and privacy analytics*, pages 64–69, 2016.

[110] Wen-Lin Chu, Chih-Jer Lin, and Ke-Neng Chang. Detection and classification of advanced persistent threats and attacks using the support vector machine. *Applied Sciences*, 9(21):4579, 2019.

[111] Ross Brewer. Advanced persistent threats: minimising the damage. *Network security*, 2014(4):5–9, 2014.

[112] Paul Giura and Wei Wang. A context-based detection framework for advanced persistent threats. In *2012 International Conference on Cyber Security*, pages 69–74. IEEE, 2012.

[113] Ibrahim Ghafir, Konstantinos G Kyriakopoulos, Sangarapillai Lambotharan, Francisco J Aparicio-Navarro, Basil AsSadhan, Hamad BinSalleeh, and Diab M Diab. Hidden markov models and alert correlations for the prediction of advanced persistent threats. *IEEE Access*, 7:99508–99520, 2019.

[114] Tero Bodström and Timo Hämäläinen. A novel deep learning stack for apt detection. *Applied Sciences*, 9(6):1055, 2019.

[115] Cho Do Xuan and Mai Hoang Dao. A novel approach for apt attack detection based on combined deep learning model. *Neural Computing and Applications*, pages 1–14, 2021.

[116] 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.

[117] Fargana J Abdullayeva. Advanced persistent threat attack detection method in cloud computing based on autoencoder and softmax regression algorithm. *Array*, 10:100067, 2021.

[118] Ronald Holt, Spencer Aubrey, Auston DeVille, William Haight, Todd Gary, and Qingguo Wang. Deep autoencoder neural networks for detecting lateral movement in computer networks. In *Proceedings on the International Conference on Artificial Intelligence (ICAI)*, pages 277–283. The Steering Committee of The World Congress in Computer Science, Computer ..., 2019.

[119] Yussuf Ahmed, Taufiq Asyhari, and Adel Aneiba. Feature forecasting for cyber attack prediction. *SN Computer Science (Submitted)*, 2021.

[120] L Constantin. Solarwinds attack explained: And why it was so hard to detect, December 2020.

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

[122] Martin Husák, Jana Komárková, Elias Bou-Harb, and Pavel Čeleda. Survey of attack projection, prediction, and forecasting in cyber security. *IEEE Communications Surveys & Tutorials*, 21(1):640–660, 2018.

[123] Nan Sun, Jun Zhang, Paul Rimba, Shang Gao, Leo Yu Zhang, and Yang Xiang. Data-driven cybersecurity incident prediction: A survey. *IEEE communications surveys & tutorials*, 21(2):1744–1772, 2018.

[124] Philippe Esling and Carlos Agon. Time-series data mining. *ACM Computing Surveys (CSUR)*, 45(1):1–34, 2012.

[125] Jonathan Z Bakdash, Steve Hutchinson, Erin G Zaroukian, Laura R Marusich, Saravanan Thirumuruganathan, Charmaine Sample, Blaine Hoffman,

and Gautam Das. Malware in the future? forecasting of analyst detection of cyber events. *Journal of Cybersecurity*, 4(1):tyy007, 2018.

[126] Zhongqing Wang and Yue Zhang. Ddos event forecasting using twitter data. In *IJCAI*, pages 4151–4157, 2017.

[127] Ashok Deb, Kristina Lerman, and Emilio Ferrara. Predicting cyber-events by leveraging hacker sentiment. *Information*, 9(11):280, 2018.

[128] Kai Shu, Amy Sliva, Justin Sampson, and Huan Liu. Understanding cyber attack behaviors with sentiment information on social media. In *International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation*, pages 377–388. Springer, 2018.

[129] Palash Goyal, KSM Hossain, Ashok Deb, Nazgol Tavabi, Nathan Bartley, Andr'es Abeliuk, Emilio Ferrara, and Kristina Lerman. Discovering signals from web sources to predict cyber attacks. *arXiv preprint arXiv:1806.03342*, 2018.

[130] S Saigal and D Mehrotra. Performance comparison of time series data using predictive data mining techniques. *Advances in Information Mining*, 4(1):57–66, 2012.

[131] Gordon Werner, Shanchieh Yang, and Katie McConky. Time series forecasting of cyber attack intensity. In *Proceedings of the 12th Annual Conference on cyber and information security research*, pages 1–3, 2017.

[132] Yaman Roumani, Joseph K Nwankpa, and Yazan F Roumani. Time series modeling of vulnerabilities. *Computers & Security*, 51:32–40, 2015.

[133] Maochao Xu, Kristin M Schweitzer, Raymond M Bateman, and Shouhuai Xu. Modeling and predicting cyber hacking breaches. *IEEE Transactions on Information Forensics and Security*, 13(11):2856–2871, 2018.

[134] Baohua Wang, Hejiao Huang, and Xiaolong Wang. A novel text mining approach to financial time series forecasting. *Neurocomputing*, 83:136–145, 2012.

[135] Cristina Nichiforov, Iulia Stamatescu, Ioana Făgărăşan, and Grigore Stamatescu. Energy consumption forecasting using arima and neural network models. In *2017 5th International Symposium on Electrical and Electronics Engineering (ISEEE)*, pages 1–4. IEEE, 2017.

[136] Mahmudur Rahman, AHM Saiful Islam, Shah Yaser Maqnoon Nadvi, and Rashedur M Rahman. Comparative study of anfis and arima model for weather forecasting in dhaka. In *2013 International Conference on Informatics, Electronics and Vision (ICIEV)*, pages 1–6. IEEE, 2013.

[137] Garima Jain and Bhawna Mallick. A study of time series models arima and ets. *Available at SSRN 2898968*, 2017.

[138] Sima Siami-Namini and Akbar Siami Namin. Forecasting economics and financial time series: Arima vs. lstm. *arXiv preprint arXiv:1803.06386*, 2018.

[139] Ümit Çavuş Büyükşahin and Şeyda Ertekin. Improving forecasting accuracy of time series data using a new arima-ann hybrid method and empirical mode decomposition. *Neurocomputing*, 361:151–163, 2019.

[140] Mohammad Valipour. Long-term runoff study using sarima and arima models in the united states. *Meteorological Applications*, 22(3):592–598, 2015.

[141] Khushbu Kumari, Suniti Yadav, et al. Linear regression analysis study. *Journal of the practice of Cardiovascular Sciences*, 4(1):33, 2018.

[142] Lubna A Gabralla and Ajith Abraham. Prediction of oil prices using bagging and random subspace. In *Proceedings of the Fifth International Conference on Innovations in Bio-Inspired Computing and Applications IBICA 2014*, pages 343–354. Springer, 2014.

[143] Andrew Gordon Wilson, David A Knowles, and Zoubin Ghahramani. Gaussian process regression networks. *arXiv preprint arXiv:1110.4411*, 2011.

[144] H Taud and JF Mas. Multilayer perceptron (mlp). In *Geomatic Approaches for Modeling Land Change Scenarios*, pages 451–455. Springer, 2018.

[145] Gordon Werner, Ahmet Okutan, Shanchieh Yang, and Katie McConky. Forecasting cyberattacks as time series with different aggregation granularity. In *2018 IEEE International Symposium on Technologies for Homeland Security (HST)*, pages 1–7. IEEE, 2018.

[146] Hicham Hammouchi, Ghita Mezzour, Mounir Ghogho, and Mohammed El Koutbi. Predicting probing rate severity by leveraging twitter sentiments. In *2019 15th international wireless communications & mobile computing conference (IWCMC)*, pages 883–888. IEEE, 2019.

[147] Rupinder Paul Khandpur, Taoran Ji, Steve Jan, Gang Wang, Chang-Tien Lu, and Naren Ramakrishnan. Crowdsourcing cybersecurity: Cyber attack detection using social media. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1049–1057, 2017.

[148] MingJian Tang, Mamoun Alazab, Yuxiu Luo, and Matthew Donlon. Disclosure of cyber security vulnerabilities: time series modelling. *International Journal of Electronic Security and Digital Forensics*, 10(3):255–275, 2018.

[149] Yussuf Ahmed and Taufiq Asyhari. Cooperative machine learning for internet of agricultural things. 2021.

[150] Yussuf Ahmed, Syed Naqvi, and Mark Josephs. Cybersecurity metrics for enhanced protection of healthcare it systems. In *2019 13th International Symposium on Medical Information and Communication Technology (ISMICT)*, pages 1–9. IEEE, 2019.

[151] J Ray. Understanding the threat landscape: Indicators of compromise (iocs), 2015.

[152] Openioc 1.1 editor, 2021.

[153] IoC Bucket. Ioc bucket community supported threat intelligence.

[154] William Ballenthin, Christopher Glyer, Josh Madeley, Nick Carr, Matt Bromiley, and Charles Carmakal. Fireeye and citrix tool scans for indicators of compromise related to cve-2019-19781, January 2020.

[155] Mohammed Lutf. Threat intelligence sharing: A survey. *Journal of Applied Science and Computations*, 8(11):1811–1815, 2018.

[156] Wiem Tounsi and Helmi Rais. A survey on technical threat intelligence in the age of sophisticated cyber attacks. *Computers & security*, 72:212–233, 2018.

[157] Yuta Kazato, Yoshihide Nakagawa, and Yuichi Nakatani. Improving maliciousness estimation of indicator of compromise using graph convolutional

networks. In *2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC)*, pages 1–7. IEEE, 2020.

[158] Umara Noor, Zahid Anwar, Tehmina Amjad, and Kim-Kwang Raymond Choo. A machine learning-based fintech cyber threat attribution framework using high-level indicators of compromise. *Future Generation Computer Systems*, 96:227–242, 2019.

[159] Hyeisun Cho, Seulgi Lee, Nakhyun Kim, Byungik Kim, and Junhyung Park. Method of quantification of cyber threat based on indicator of compromise. In *2018 International Conference on Platform Technology and Service (PlatCon)*, pages 1–6. IEEE, 2018.

[160] Amirreza Niakanlahiji, Lida Safarnejad, Reginald Harper, and Bei-Tseng Chu. Iocminer: Automatic extraction of indicators of compromise from twitter. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 4747–4754. IEEE, 2019.

[161] Will Gibb & Devon Kerr. Openioc: Back to the basics, October 2013.

[162] Xiaojing Liao, Kan Yuan, XiaoFeng Wang, Zhou Li, Luyi Xing, and Raheem Beyah. Acing the ioc game: Toward automatic discovery and analysis of open-source cyber threat intelligence. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 755–766, 2016.

[163] Onur Catakoglu, Marco Balduzzi, and Davide Balzarotti. Automatic extraction of indicators of compromise for web applications. In *Proceedings of the 25th international conference on world wide web*, pages 333–343, 2016.

[164] David Bianco. The pyramid of pain, January 2014.

[165] Yussuf Ahmed, Syed Naqvi, and Mark Josephs. Aggregation of security metrics for decision making: a reference architecture. In *Proceedings of the 12th European Conference on Software Architecture: Companion Proceedings*, pages 1–7, 2018.

[166] Yi Cheng, Julia Deng, Jason Li, Scott A DeLoach, Anoop Singhal, and Xinming Ou. Metrics of security. In *Cyber defense and situational awareness*, pages 263–295. Springer, 2014.

[167] Thycotic. State of cybersecurity metrics annual report. Report, Thycotic, 2017.

[168] McAfee Labs. Mcafee labs 2017 threats predictions. November 2017.

[169] NIST. Federal cybersecurity research and development strategic plan. Report, NIST, February 2016.

[170] DM Nicol, WL Scherlis, J Katz, WL Scherlis, T Dumitras, LM Williams, and MP Singh. Science of security lablets progress on hard problems. *Science of Security and Privacy Virtual Organization: http://cps-vo. org/node/21590. Accessed*, 7:18, 2015.

[171] Marcus Pendleton, Richard Garcia-Lebron, Jin-Hee Cho, and Shouhuai Xu. A survey on systems security metrics. *ACM Computing Surveys (CSUR)*, 49(4):1–35, 2016.

[172] Alex Ramos, Marcella Lazar, Raimir Holanda Filho, and Joel JPC Rodrigues. Model-based quantitative network security metrics: A survey. *IEEE Communications Surveys & Tutorials*, 19(4):2704–2734, 2017.

[173] Subil Abraham and Suku Nair. A predictive framework for cyber security analytics using attack graphs. *arXiv preprint arXiv:1502.01240*, 2015.

[174] Jin B Hong, Simon Yusuf Enoch, Dong Seong Kim, Armstrong Nhlabatsi, Noora Fetais, and Khaled M Khan. Dynamic security metrics for measuring the effectiveness of moving target defense techniques. *Computers & Security*, 79:33–52, 2018.

[175] Jin-Hee Cho, Dilli P Sharma, Hooman Alavizadeh, Seunghyun Yoon, Noam Ben-Asher, Terrence J Moore, Dong Seong Kim, Hyuk Lim, and Frederica F Nelson. Toward proactive, adaptive defense: A survey on moving target defense. *IEEE Communications Surveys & Tutorials*, 22(1):709–745, 2020.

[176] Matheus Torquato and Marco Vieira. Moving target defense in cloud computing: A systematic mapping study. *Computers & Security*, 92:101742, 2020.

[177] Elena Doynikova and Igor Kotenko. Approach for determination of cyber-attack goals based on the ontology of security metrics. In *IOP Conference Series: Materials Science and Engineering*, volume 450, page 052006. IOP Publishing, 2018.

[178] Kai Petersen, Cigdem Gencel, Negin Asghari, and Stefanie Betz. An elicitation instrument for operationalising gqm+ strategies (gqm+ s-ei). *Empirical Software Engineering*, 20(4):968–1005, 2015.

[179] Victor Basili, Adam Trendowicz, Martin Kowalczyk, Jens Heidrich, Carolyn Seaman, Jürgen Münch, and Dieter Rombach. Gqm+ strategies in a nutshell. In *Aligning Organizations Through Measurement*, pages 9–17. Springer, 2014.

[180] Su Zhang, Xinwen Zhang, Xinming Ou, Liqun Chen, Nigel Edwards, and Jing Jin. Assessing attack surface with component-based package depen-

dency. In *International Conference on Network and System Security*, pages 405–417. Springer, 2015.

[181] First. Common vulnerability scoring system v3.1: Specification document.

[182] Perpetus Jacques Houngbo and Joël Toyigbé Hounsou. Measuring information security: understanding and selecting appropriate metrics. *International Journal of Computer Science and Security (IJCSS)*, 9(2):108, 2015.

[183] Elizabeth Chew, Marianne M Swanson, Kevin M Stine, Nadya Bartol, Anthony Brown, and Will Robinson. Performance measurement guide for information security. 2008.

[184] Hanan Hindy, David Brosset, Ethan Bayne, Amar Seeam, Christos Tachtatzis, Robert Atkinson, and Xavier Bellekens. A taxonomy and survey of intrusion detection system design techniques, network threats and datasets. 2018.

[185] Angus Naylor, James Ford, Tristan Pearce, and James Van Alstine. Conceptualizing climate vulnerability in complex adaptive systems. *One Earth*, 2(5):444–454, 2020.

[186] Mauro Conti, Tooska Dargahi, and Ali Dehghantanha. Cyber threat intelligence: challenges and opportunities. In *Cyber Threat Intelligence*, pages 1–6. Springer, 2018.

[187] NIST. Guide for conducting risk assessments. Nist special publication 800-30 revision 1, NIST, September 2012.

[188] Aileen G Bacudio, Xiaohong Yuan, Bei-Tseng Bill Chu, and Monique Jones. An overview of penetration testing. *International Journal of Network Security & Its Applications*, 3(6):19, 2011.

[189] ATTACKIQ. Security optimization platform, 2021.

[190] Steve Mansfield-Devine. The best form of defence–the benefits of red teaming. *Computer Fraud & Security*, 2018(10):8–12, 2018.

[191] Reza Arghandeh, Alexandra Von Meier, Laura Mehrmanesh, and Lamine Mili. On the definition of cyber-physical resilience in power systems. *Renewable and Sustainable Energy Reviews*, 58:1060–1069, 2016.

[192] Wayne Harrop and Ashley Matteson. Cyber resilience: A review of critical national infrastructure and cyber-security protection measures applied in the uk and usa. *Current and emerging trends in cyber operations*, pages 149–166, 2015.

[193] CIS. The 18 cis controls & resources.

[194] ENISA. Isms framework.

[195] ISO/IEC. Iso/iec 27004:2016 information technology — security techniques — information security management — monitoring, measurement, analysis and evaluation.

[196] NCSC. Cyber security for your organisation starts here.

[197] NIST. Nist cybersecurity framework.

[198] Ncsc caf guidance, September 2019.

[199] Blake Anderson and David McGrew. Os fingerprinting: New techniques and a study of information gain and obfuscation. In *2017 IEEE Conference on Communications and Network Security (CNS)*, pages 1–9. IEEE, 2017.

[200] Daniel Pauli. Robots.txt tells hackers the places you don't want them to look, May 2015.

[201] Samuel Marchal, Jérôme François, Cynthia Wagner, and Thomas Engel. Semantic exploration of dns. In *International Conference on Research in Networking*, pages 370–384. Springer, 2012.

[202] Amichai Shulman, Michael Cherny, and Sagie Dulce. Compromised insider honey pots using reverse honey tokens, July 26 2016. US Patent 9,401,927.

[203] Nir Nissim, Ran Yahalom, and Yuval Elovici. Usb-based attacks. *Computers & Security*, 70:675–688, 2017.

[204] James P Farwell and Rafal Rohozinski. Stuxnet and the future of cyber war. *Survival*, 53(1):23–40, 2011.

[205] Frank Vanhoenshoven, Gonzalo Nápoles, Rafael Falcon, Koen Vanhoof, and Mario Köppen. Detecting malicious urls using machine learning techniques. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8. IEEE, 2016.

[206] Shashank Gupta and Brij Bhooshan Gupta. Cross-site scripting (xss) attacks and defense mechanisms: classification and state-of-the-art. *International Journal of System Assurance Engineering and Management*, 8(1):512–530, 2017.

[207] Danny Hendler, Shay Kels, and Amir Rubin. Detecting malicious powershell commands using deep neural networks. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, pages 187–197, 2018.

[208] NVD. Cve-2019-1069 detail.

[209] Reza Sharifnya and Mahdi Abadi. Dfbotkiller: Domain-flux botnet detection based on the history of group activities and failures in dns traffic. *Digital Investigation*, 12:15–26, 2015.

[210] Asaf Nadler, Avi Aminov, and Asaf Shabtai. Detection of malicious and low throughput data exfiltration over the dns protocol. *Computers & Security*, 80:36–53, 2019.

[211] Md Shamim Reza and Jinwen Ma. Ica and pca integrated feature extraction for classification. In *2016 IEEE 13th International Conference on Signal Processing (ICSP)*, pages 1083–1088. IEEE, 2016.

[212] François Husson and Julie Josse. Multiple correspondence analysis. *Visualization and verbalization of data*, pages 165–184, 2014.

[213] Hervé Abdi, Lynne J Williams, and Domininique Valentin. Multiple factor analysis: principal component analysis for multitable and multiblock data sets. *Wiley Interdisciplinary reviews: computational statistics*, 5(2):149–179, 2013.

[214] Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P Trevino, Jiliang Tang, and Huan Liu. Feature selection: A data perspective. *ACM Computing Surveys (CSUR)*, 50(6):1–45, 2017.

[215] Seo Jin Lee, Paul D Yoo, A Taufiq Asyhari, Yoonchan Jhi, Lounis Chermak, Chan Yeob Yeun, and Kamal Taha. Impact: Impersonation attack detection via edge computing using deep autoencoder and feature abstraction. *IEEE Access*, 8:65520–65529, 2020.

[216] Md Arafatur Rahman, A Taufiq Asyhari, Ong Wei Wen, Husnul Ajra, Yussuf Ahmed, and Farhat Anwar. Effective combining of feature selection

techniques for machine learning-enabled iot intrusion detection. *Multimedia Tools and Applications*, pages 1–19, 2021.

[217] Radhika Chapaneri and Seema Shah. A comprehensive survey of machine learning-based network intrusion detection. *Smart Intelligent Computing and Applications*, pages 345–356, 2019.

[218] 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, 2019.

[219] Elike Hodo, Xavier Bellekens, Andrew Hamilton, Christos Tachtatzis, and Robert Atkinson. Shallow and deep networks intrusion detection system: A taxonomy and survey. *arXiv preprint arXiv:1701.02145*, 2017.

[220] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A Ghorbani. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *ICISSP*, pages 108–116, 2018.

[221] Zafar A Khan and Dilan Jayaweera. Approach for forecasting smart customer demand with significant energy demand variability. In *2018 1st International Conference on Power, Energy and Smart Grid (ICPESG)*, pages 1–5. IEEE, 2018.

[222] Xiaoou Monica Zhang, Katarina Grolinger, Miriam AM Capretz, and Luke Seewald. Forecasting residential energy consumption: Single household perspective. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 110–117. IEEE, 2018.

[223] Eugene Yu Shchetinin. Cluster-based energy consumption forecasting in smart grids. In *International Conference on Distributed Computer and Communication Networks*, pages 445–456. Springer, 2018.

[224] Surbhi Vijh, Adesh Kumar Pandey, Garima Vijh, and Sumit Kumar. Stock forecasting for time series data using convolutional neural network. In *2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pages 866–870. IEEE, 2021.

[225] R Geetha, K Ramyadevi, and M Balasubramanian. Prediction of domestic power peak demand and consumption using supervised machine learning with smart meter dataset. *Multimedia Tools and Applications*, pages 1–19, 2021.

[226] Kumar Abhishek, MP Singh, Saswata Ghosh, and Abhishek Anand. Weather forecasting model using artificial neural network. *Procedia Technology*, 4:311–318, 2012.

[227] S Santhosh Baboo and I Kadar Shereef. An efficient weather forecasting system using artificial neural network. *International journal of environmental science and development*, 1(4):321, 2010.

[228] Tanzila Saba, Amjad Rehman, and Jarallah S AlGhamdi. Weather forecasting based on hybrid neural model. *Applied Water Science*, 7(7):3869–3874, 2017.

[229] Dires Negash Fente and Dheeraj Kumar Singh. Weather forecasting using artificial neural network. In *2018 second international conference on inventive communication and computational technologies (ICICCT)*, pages 1757–1761. IEEE, 2018.

[230] Tianfeng Chai and Roland R Draxler. Root mean square error (rmse) or mean absolute error (mae). *Geoscientific Model Development Discussions*, 7(1):1525–1534, 2014.

[231] Felix A Gers, Douglas Eck, and Jürgen Schmidhuber. Applying lstm to time series predictable through time-window approaches. In *Neural Nets WIRN Vietri-01*, pages 193–200. Springer, 2002.

[232] Arul Earnest, Mark I Chen, Donald Ng, and Leo Yee Sin. Using autoregressive integrated moving average (arima) models to predict and monitor the number of beds occupied during a sars outbreak in a tertiary hospital in singapore. *BMC Health Services Research*, 5(1):1–8, 2005.

[233] November 2021.

[234] Microsoft. Events to monitor, July 2018.