ANOMALY DETECTION IN SYSTEM LOG FILES USING MACHINE LEARNING ALGORITHMS

ZAHEDEH ZAMANIAN

FACULTY OF COMPUTER SCIENCE AND
INFORMATION TECHNOLOGY
UNIVERSITY OF MALAYA
KUALA LUMPUR

2019

ANOMALY DETECTION IN SYSTEM LOG FILES USING MACHINE LEARNING ALGORITHMS

ZAHEDEH ZAMANIAN

DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF COMPUTER SCIENCE

FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY
UNIVERSITY OF MALAYA
KUALA LUMPUR

2019

UNIVERSITI MALAYA
ORIGINAL LITERARY WORK DECLARATION

Name of Candidate: **Zahedeh Zamanian**
Registration/Matric No: **WOA160035**
Name of Degree: **Master of Computer Science**
Title of Dissertation ("this Work"): **ANOMALY DETECTION IN SYSTEM LOG FILES USING MACHINE LEARNING ALGORITHMS**

Field of Study: **ANOMALY DETECTION**

I do solemnly and sincerely declare that:

(1) I am the sole author/writer of this Work;
(2) This Work is original;
(3) Any use of any work in which copyright exists was done by way of fair dealing and for permitted purposes and any excerpt or extract from, or reference to or reproduction of any copyright work has been disclosed expressly and sufficiently and the title of the Work and its authorship have been acknowledged in this Work;
(4) I do not have any actual knowledge, nor do I ought reasonably to know that the making of this work constitutes an infringement of any copyright work;
(5) I hereby assign all and every right in the copyright to this Work to the University of Malaya ("UM"), who henceforth shall be owner of the copyright in this Work and that any reproduction or use in any form or by any means whatsoever is prohibited without the written consent of UM having been first had and obtained;
(6) I am fully aware that if in the course of making this Work I have infringed any copyright whether intentionally or otherwise, I may be subject to legal action or any other action as may be determined by UM.

Candidate's Signature Date

Subscribed and solemnly declared before,

Witness's Signature Date

Name:

Designation:

ANOMALY DETECTION IN SYSTEM LOG FILES USING MACHINE LEARNING ALGORITHMS

Abstract

In recent years due to rapid growth of information technology and easy access to computers, digital devices and internet, security management and investigating malicious activity have been main concern of organization and governments. People who are greatest asset of organization, they may also be the greatest threat due to their access to highly confidential information and their knowledge of the organizational systems. Insider threat activity has huge impact on business. Therefore, there is a need for methods to detect insider threats inside an organization. Log files are great source of information which can help to detect, understand and predict these kinds of threats. However, the sheer size of log files generated by systems makes human log analysis impractical. Moreover, log files have a lot of irrelevant and redundant features that act as noise. Also, log files are heterogenous and cannot fed them directly in machine learning algorithms. Furthermore, many of the companies use the signature-based detection method which is not capable of capturing more advanced attackers that use unfamiliar attacks methods. This study uses machine learning method to detect anomalies in system log files. This study uses synthetic CERT Insider Threat v6.2 dataset that includes five different domains of file, logon/logoff, http, device and email. This study generates 200 features from raw system log files that can be fed to machine learning. This study uses principal component analysis (PCA) as a feature extraction method to extract 117 independent and discriminative features with 95% of variance. This study applies unsupervised Isolation Forest and One Class SVM as ML algorithms to detect anomalies. Isolation Forest area under curve (AUC) successfully achieved 96.6% with applying PCA

and without PCA, lowest value of AUC was 76%. In contrast, the AUC value for One Class SVM was 69.3% with applying PCA and 59.8% without PCA. Isolation Forest true positive rate (TPR) successfully achieved 93.2% with applying PCA and without PCA, value of TPR was 89.2%. On the other hand, the TPR value for One Class SVM was 68.1% with applying PCA and 55.4% without PCA. The highest FPR result of 26% was obtained by One Class SVM without PCA and the lowest FPR result of 2.8% was obtained by Isolation Forest with applying PCA.

Keywords: anomaly detection, machine learning, insider threats, feature extraction

PENGESANAN KEGANJILAN DALAM FAIL SISTEM LOG MENGGUNAKAN ALGORITMA PEMBELAJARAN MESIN

Abstrak

Dalam tahun-tahun kebelakangan ini disebabkan oleh pertumbuhan pesat teknologi maklumat dan akses mudah ke komputer, peranti digital dan internet, pengurusan keselamatan dan penyiasatan aktiviti berniat jahat telah menjadi kebimbangan utama organisasi dan kerajaan. Orang yang menjadi aset organisasi terbesar, mereka juga mungkin menjadi ancaman terbesar kerana akses mereka kepada maklumat yang sangat sulit dan pengetahuan mereka tentang sistem organisasi. Aktiviti ancaman dalaman mempunyai kesan besar terhadap perniagaan. Oleh itu, terdapat keperluan untuk mengesan ancaman dalaman di dalam organisasi. Fail log adalah sumber maklumat yang hebat yang dapat membantu untuk mengesan, memahami dan meramalkan jenis ancaman ini. Walau bagaimanapun, saiz fail log yang dihasilkan oleh sistem membuat analisis log manusia tidak praktikal. Selain itu, fail log mempunyai banyak ciri tidak relevan dan berlebihan yang bertindak sebagai bunyi bising . Juga, fail log adalah heterogen dan tidak boleh memberi mereka secara langsung dalam algoritma pembelajaran mesin. Selain itu, banyak syarikat menggunakan kaedah pengesanan berasaskan tandatangan yang tidak dapat menangkap penyerang yang lebih maju yang menggunakan kaedah serangan yang tidak dikenali. Dalam kajian ini, kami menggunakan kaedah pembelajaran mesin untuk mengesan anomali dalam fail log sistem. Kami menggunakan dataset CERT Insider Threat v6.2 sintetik yang merangkumi lima domain yang berlainan fail, log in / logoff, http, peranti dan e-mel. Kami menjana 200 ciri dari fail log sistem mentah yang boleh diberi makan untuk pembelajaran mesin. Kami menggunakan Principal Component Analysis (PCA) sebagai kaedah pengekstrakan ciri

untuk mengeluarkan 117 ciri bebas dan diskriminatif dengan 95% varians. Kami menggunakan unsupervised Isolation Forest (iForest) dan One Class SVM sebagai algoritma ML untuk mengesan anomali. Isolation Forest AUC berjaya mencapai 96.6% dengan menggunakan PCA dan tanpa PCA, nilai terendah AUC adalah 76%. Sebaliknya, nilai AUC untuk One Class SVM adalah 69.3% dengan menggunakan PCA dan 59.8% tanpa PCA. Kadar positif benar iForest (TPR) berjaya mencapai 93.2% dengan menggunakan PCA dan tanpa PCA, nilai TPR adalah 89.2%. Sebaliknya, nilai TPR untuk One Class SVM adalah 68.1% dengan menggunakan PCA dan 55.4% tanpa PCA. Hasil FPR tertinggi sebanyak 26% diperolehi oleh One Class SVM tanpa PCA dan hasil FPR terendah sebanyak 2.8% diperolehi oleh iForest yang menggunakan PCA.

Kata kunci: Pengesanan Keganjilan, Pembelajaran Mesin, ancaman dalaman, pengekstrakan ciri

Acknowledgments

First and above all, I praise God, the almighty for providing me this opportunity and granting me the capability to proceed successfully. This thesis appears in its current form due to the assistance and guidance of several people and I know it would never have happened without them. I would therefore like to offer my sincere thanks to all of them.

I would like to express my deepest appreciation to my supervisor Associate Prof Dr. Nor Badrul Anuar Bin Juma'at for engaging me in such an exciting subject and for his supervision, support, guidance and endless insight throughout the course of this research. I have learned a great deal of knowledge from him and he showed tremendous patience as I learned the art of researching.

I would like to express my deepest gratitude to Dr. Ali Feizollah for his support, dedication and patience. He devoted so much time to share his knowledge, expertise and insights. You have made a valuable impact to my research, and to the completion of this dissertation. Your help will not be forgotten.

Most importantly, I would like to express my deepest sense of gratitude to my beloved husband, Kambiz, for unconditional love, continued support and help over the years. I want to express my gratitude and deepest appreciation to my lovely sweet son, Mahan, for his great patience and understandings. I especially must restate my sincere appreciations to my parents, family, and my husband's family for the continued help, support and encouragement they always provide me.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AV	Antivirus
AI	Artificial Intelligence
AUC	Area Under Curve
CI	Confidence Interval
COF	Connectivity-based Outlier Factor
CPU	Central Processing Unit
IC3	Internet Crime Complaint Center
DNN	Deep Neural Network
FPR	False Positive Rate
GTA	Game-Theoretic Approach
GA	Genetic Algorithm
HMM	Hidden Markov Models
HTTP	Hyper Text Transfer Protocol
ISACA	Information Systems Audit and Control Association
IDE	Integrated Development Environment
ID	Intrusion Detection
iForest	Isolation Forest
KNN	Kth Nearest Neighbor
LDA	Linear Discriminant Analysis
LOF	Local Outlier Factor
ML	Machine Learning
NIST	National Institute of Standards and Technology
PCA	Principal Component Analysis
RAM	Random Access Memory
RF	Random Forest
ROC	Receiver Operating Characteristic
RDBMS	Relational Database Management System
RNN	Replicator Neural Network
SPSS	Statistical Package for the Social Science

SQL	Structure Query Language
SVDD	Support Vector Data Description
SVM	Support Vector Machine
TPR	True Positive Rate
URL	Uniform Resource Locator
VPN	Virtual Private Network

CHAPTER 1: Introduction

1.1 Introduction

This chapter shows an introduction to the research work proposed. It contains various aspects related to the research work. It presents a general background about the research area followed by the motivation, problem statement, objectives and the scope of research.

1.2 Background of Study

Information is the crown jewels of every business, but integrity, availability and confidentiality of these information are predominant concerns of companies and organizations. Information and communications technologies have resulted in increasing accessibility to the Internet and reduced costs for corporations, it has also resulted in vulnerability of organization to both insiders and outsiders threats(R. Prasad, 2009).Thus, data protection and securing networks becomes vitally important. As defined by International ISO/IEC 17799:2000, confidentiality means ensuring that information is accessible only to those authorized to have access. Therefore, there are two types of unauthorized access:

- External Penetrator: An agent from outside the organization who is not authorized to have access.
- Internal Penetrator: An agent who belongs to the organization but violates his or her legitimate access rights (A Diaz-Gomez et al., 2017)

The FBI's Internet Crime Complaint Center (IC3) 2016 report shows during 2012 until 2016 approximately 280,000 internet scams complaints per year was received by IC3. Figure 1 illustrates IC3 received a total of 1,408,849 complaints, and a total reported loss of $4.63

billion during 2012 until 2016. The complaints address a wide range of Internet scams affecting victims across the globe (Smith, 2016).

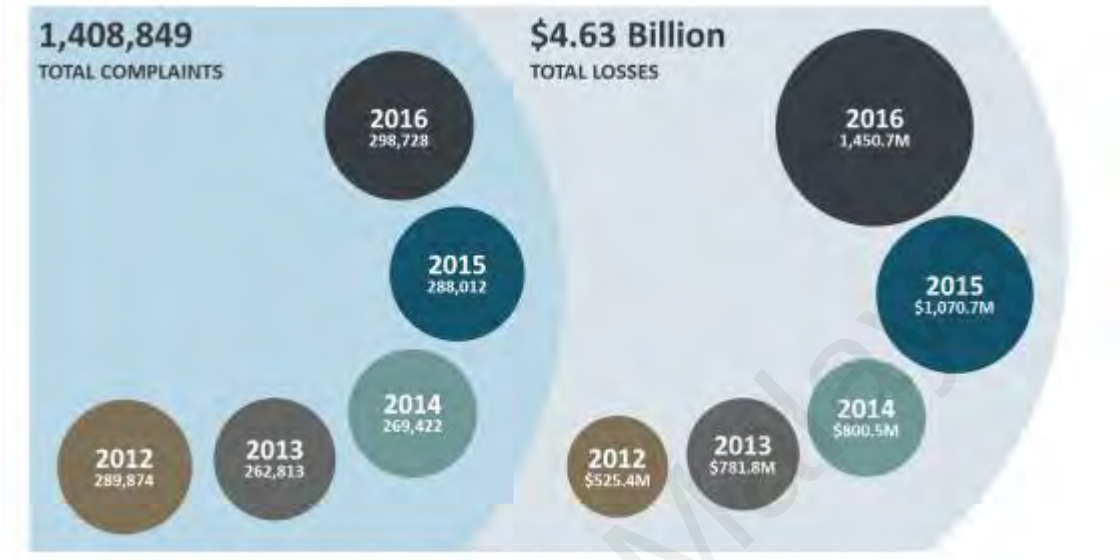

Figure 1.1: Number of complain and money loss from 2012 until 2016 (Smith, 2016)

Therefore, it is crucial for companies to realize threats that influence their assets and the areas which each threat could affect (Bauer & Bernroider, 2017). Analyzing log files is one form of defense mechanism against these kinds of attacks. Activity log or log file is a collection of event records which is occurring within a company's systems and networks. Logs are consisted of log entries; each entry contains information related to an event including the use of specific system resources, system status changes, and general performance issues (Vaarandi et al., 2016). A medium to large company tends to generate and collect sheer size of activity logs which typically contains hundreds and thousands of lines. Analyzing and classifying such huge sets of data manually, for anomaly detection or reporting purposes, is tedious and nearly impossible. Therefore, there is a need for automated analysis tools that detect peculiar and malicious behavior that is unlikely to be spotted by a human (Breier & Branišová, 2015).

Chandola *et al*. state that anomaly detection refers to the problem of finding patterns in data that do not conform to expected behavior. These nonconforming patterns are often referred to as anomalies or outliers. The anomaly detection provides very important and crucial information from a computer security perspective. It can detect malicious activity such as unauthorized use, penetrations, and other forms of computer abuse (Chandola et al., 2009). When data needs to be analyzed in order to find pattern or to predict known or unknown attacks, data mining techniques are applied. These could be categorized to clustering, classification and other machine-based learning techniques. In addition, hybrid methods are also being used to get higher level of accuracy on detecting anomalies (Agrawal & Agrawal, 2015).

There are three types of cyber analytics in support of intrusion detection:

1. Misuse (Signature) that are designed to detect known attacks by using signatures of those attacks. They are effective for detecting known type of attacks without generating large number of false alarms. However, as a downside they require frequent manual update of database with new rules and cannot detect novel attack.
2. Anomaly based techniques that model normal network and system behavior, identify anomalies as deviation from normal behavior. Their main advantage is ability to detect novel attack. Moreover, the profiles of normal activity are customized for every system, application, or network, so it will be difficult for attackers to know which activities they can carry out undetected. The main disadvantage of this method is high rate of false positive.
3. Hybrid methods combine misuse and anomaly detection. They are used to increase detection rates of known intrusions and decrease the FPR for unknown attacks (Ling Ko et al., 2016)

In anomaly-based category, behavioral based method usually is used to detect insider attack. This method can be grouped into

- System behaviors, and
- User behaviors.

The system behaviors are generated by hosts and networks and relate to the host activities and network status. In contrast, the user behaviors mainly relate to the direct interaction between the user and the system, for example, typing patterns (Peng et al., 2016). Researchers have designed various models for anomaly detection. These models have helped to refine and improve the anomaly detection concept.

1.3 Motivation

The following are the motivations for this study:

- **Increase in Internet scam attack:** According to IC3 2016 report number of Internet scam complains increase each year and the amount of money loss has increased significantly (Smith, 2016). Moreover, based on ISACA and RSA Conference Survey in 2016, 42% and 64% of respondents agreed that rapid advancement of Artificial Intelligent (AI) will lead to increase of the cybersecurity/information security risk in short term and long term, respectively. The report data reveal that 20% of companies are dealing with insider damage and theft of intellectual property at least quarterly (Nexus, 2016). considering that some cyber-attacks go undetected.
- **Lack of security awareness**: Based on ISACA and RSA Conference Survey in 2015, 25.28% of successful attack types was insiders theft and only 26.66% of the victim companies had security awareness program about this kind of attack (Nexus, 2015). Most of the companies use the signature-based detection method, which is not capable

of capturing more advanced attackers that use unfamiliar attacks method. Anomaly based techniques are able to detect these kind of attacks (Parmar & Patel, 2017).

- **Data dimensionality:** The problems of high dimensional data for the first time was introduced by Richard Bellman as "the curse of dimensionality" (Bellman & Corporation, 1957). These terms refer to organizing and analyzing of data which have hundreds or thousands of dimensions (features). Most of these features are irrelevant or redundant and may lead to complexity, overfitting, low accuracy and higher computational cost (Guyon et al., 2006). Log files which are collected by companies have a lot of irrelevant and redundant features that act as noise and cause above - mentioned problems. To mitigate this problem feature reduction techniques such as feature extraction and feature selection could be used. In feature extraction approach, features are projected into a new space with lower dimensionality. In contrast, the feature selection approach aims to select a small subset of features (Wang et al., 2016).
- **High cost of false positives:** All anomaly detection techniques suffer from a common problem. The false positive error which is a normal or expected behavior that is identified as anomalous or malicious. Security analyst needs to do deep analysis by each false alarm to disprove the maliciousness of the activity. Defiantly, this task can take up to several working days. Therefore, researchers have been searching to design methods with low false positive alarm (Jyothsna et al., 2011).

1.4 Problem Statement

Intrusion detection has been an important research problem in security analysis. It is a complex problem whose solution differs from domain to domain and application to

application (Kongsg et al., 2017). According to Ponemon's Cyber Crime report 2017, cyber-crime detection and containment activities account for 56% of total internal activity cost, as shown in Figure 1.2. Graph shows both detection and containment costs have increased since 2015. Therefore, using methods that facilitate detection process show up notable cost reduction opportunity for organizations.

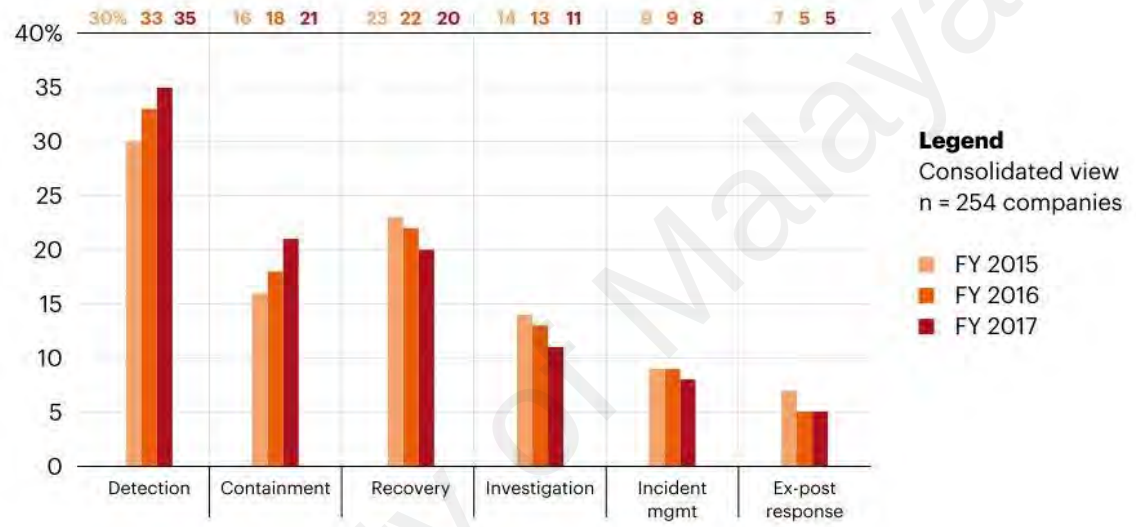

Figure 1.2: Percentage cost by internal activities (Kevin Richards et al., 2017).

As mentioned earlier, there are different methods of intrusion detection. Although signature base techniques are easy to implement but ill-intentioned hackers are aware of those patterns too and in their new attack intentionally avoiding those patterns. Moreover, it is hard to update these new patterns quickly, so signature-based methods are ineffective to detect new patterns. At this point, a new detection approach needs to be used. Anomaly detection technique is one of the important research areas that use machine learning methods to detect malicious behaviors. Ponemon's Cyber Crime report 2017 illustrates by using Automation, orchestration and machine learning techniques, organization can save up to US$2.4 million on average. Anomaly detection model behavior of users with normal traffic and interprets

deviations from this normal behavior as an anomaly. Although defining normal profiles is difficult and inappropriate normal traffic profiles leads to poor performance (Priya C V & Angel Viji 2002).

Currently, anomaly detection systems are susceptible to low rate of detection and high false positive errors thus are not completely reliable during usage. They may send too many false alarms to analyst or failing to alert about significant attack. High false alarm can cause notable amount of noise for an analyst to examine and decide if the alarm is true or not. There is a possibility these kinds of alarm lead them to missed true attacks. Another issue is low detection rate. It means true attack cannot be classified as malicious behavior by the system. Therefore it is highly unlikely the analyst will have the ability to detect them (Sun et al., 2016). Although, the current systems have improved over the time in their accuracy, there is plenty of room for improvement in terms of high accuracy and low rate of false alarm (Buczak & Guven, 2016).

Thus, this study evaluates the effectiveness of anomaly detection technique to detect insider threat in CERT dataset (Glasser & Lindauer, 2013). Among the recent works, Gavai et al (G. Gavai et al., 2015) conducted a similar study on a smaller dataset. Our study uses bigger dataset and apply feature extraction method to reduce dimensionality.

1.5 Research Objectives

The goal of this research is to evaluate anomaly detection methods for detecting insider threats in system log files. The following are the objectives of this research:

1. To comprehensively study current methods of anomaly detection in detecting insider threats.
2. To apply the feature extraction method for dimensionality reduction.

3. To evaluate the effectiveness of machine learning algorithms in detecting insider threats.

1.6 Scope of Project

This study is primarily concerned with detecting insider threat by using system behavior of employees on the CERT dataset includes five different domains of file, logon/logoff, http, device and email. This study uses authorized log files for users. The user behavior data such as psychometric data (personality traits and characteristics) is excluded from the analysis.

1.7 Layout of Thesis

This thesis includes of five chapters and it is structured as below.

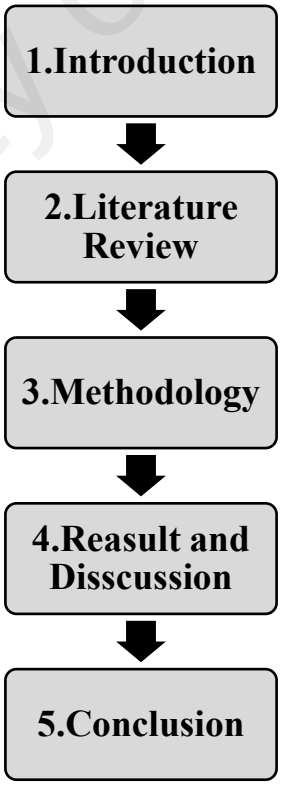

Figure 1.3: Thesis Structure

1. Chapter 1 (Introduction)

Chapter 1 contains eight sections; the first section is a brief introduction of this chapter. The second section gives a background of this study, that includes discussion on some previous researches, and the third section discusses motivation behind this study. The fourth section states the problem statements that were formulated based on findings from other researchers and considers some of the issues not addressed. The fifth section lists the objective of this study, while the sixth section explains the scope of project. The seventh section outlines the layout of the thesis and finally eighth section summarized the chapter.

2.Chapter 2 (Literature Review)

Chapter 2 contains nine sections. The first section is a brief introduction of this chapter. The second section explains log files and how they are generating. In the third section, describe intrusion detection methods and in the fourth section list down different type of machine learning algorithms as well as ML algorithms that is used in this thesis. The fifth section describes feature reduction methods. The sixth section explains the importance of feature engineering and the seventh section discusses about dataset. The eighth section provides a comprehensive overview of the previous researches that are related to this thesis and finally, the ninth session summarized the chapter.

3.Chapter 3 (Methodology)

This chapter presents the required tools and other software which is needed for the experiments and followed by complete describe of CERT dataset. The chapter also highlights the data aggregation, feature engineering and feature extraction. Moreover, in this chapter hyperparameter tuning is discussed.

4.Chapter 4 (Result and Discussion)

Chapter 4 reveals the evaluation measurement applied in the experiments, the result performance analysis, ROC curve graph, and width of confidence interval. The result obtained are analyzed, in order to determine how the objective of the experiments have been achieved.

5.Chapter 5 (Conclusion)

Chapter 5 presents the conclusion to the study and considers the results obtained as the achievement of research objectives and the contribution of this research. It highlights the significance of the proposed solution and states the limitation of the research work along with directions for the future research on this topic.

1.8 Summary

This chapter presented background of the research, which included the motivation, research problem, objectives, scope of project and layout of thesis. The next chapter discusses the existing literature on anomaly detection.

CHAPTER 2: Literature Review

This chapter presents the theoretical background on log files and intrusion detection methods to detect malicious activity inside log files and reviews the state-of-the-art anomaly detection methods in particular, the machine learning unsupervised approach. The chapter is organized into eight sections. Section 2.2 explains the concept of log files and their applications. Section 2.3 and 2.4 explains the fundamental concept of previous researches on intrusion detection and machine learning techniques. Section 2.5 and 2.6 discuss the importance of feature reduction and feature engineering. Section 2.7 explains the different source of data. Section 2.8 presents a summary of current scene in anomaly detection using machine learning techniques. Section 2.9 summarizes the chapter with concluding remarks.

2.1 Introduction

In recent years due to rapid growth of information technology and easy access to computers, digital devices and internet, cyber security management and investigating malicious activity have been main concern of organization and governments. Cyber security is the set of technologies and processes designed to protect computers, networks, programs, information and data from attack, unauthorized access, change, or destruction (Mukkamala et al., 2005). According cyber security management, organizations are facing two kinds of threat:

- External Penetrator: An agent from outside the organization who is not authorized to have access.
- Internal Penetrator: An authorized agent who belongs to the organization but trespass his/her privileges and violating organizational security policy (A Diaz-Gomez et al., 2017).

Internal penetrator or insider threat are more damaging compare to external ones due to their access to highly confidential information and their knowledge of the organizational systems. Insider threat activity has huge impact on business, which according to Ponemon's Cyber Crime report in 2017, malicious insiders are most expensive cyber-attacks when analyzed by the frequency of incidents. Companies spent an average of US$ 1.4 million on malicious insider attacks in 2017. Also, report shows that the average days to resolve malicious insider attack is about 50 days which needs most amount of time to resolve in contrast with other kind of cyber-attacks (Kevin Richards et al., 2017). Therefore, there is a vital need for methods to detect ill-intentioned insiders within an organization.

2.2 Log Files

Log files are great source of information which can help to detect, understand and predict insider threats. Based on Computer Security Log Management "Activity log or log file is a record of the events occurring within an organization's systems and networks. Logs are composed of log entries; each entry contains information related to an event including the use of specific system resources, system status changes, and general performance issues" (Kent & Souppaya, 2006). Log files are generated by many different resources like Unix and Windows system, Switches, Firewalls, Routers, Wireless Access Points, Virtual Private Network (VPN) Server, Antivirus (AV) Systems and Printers. Figure 2.1 illustrates layout of a distributed logging setup. It should be known that every device, computer system, and application in network is capable of logging. Log messages can contain different information but typically, content of the log files can be grouped in three parts:

- Timestamp: The occurrence time associated with the event.

- Source: System that generated the log file represented in IP address or hostname format.
- Data: No standard format, it could represent source and destination IP address, source and destination ports, user names, program names, resource objects like file, directory, byte transferred in or out (Chuvakin et al., 2013).

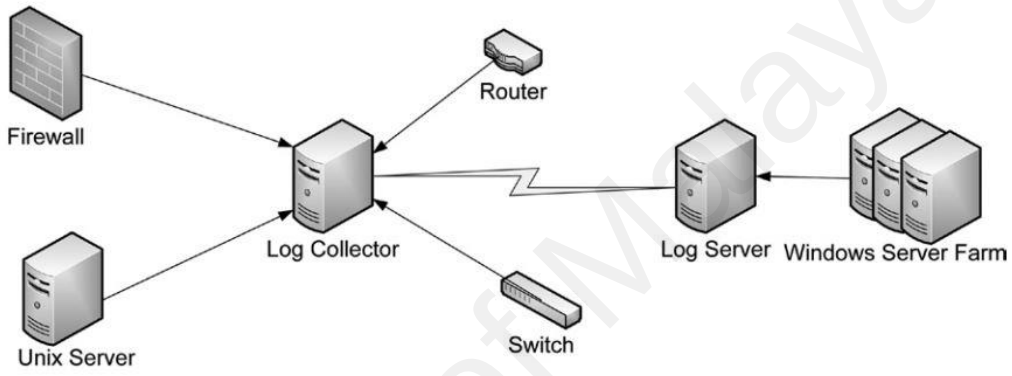

Figure 2.1: Layout of a Distributed Logging Setup

Primarily, logs were utilized for troubleshooting errors and debugging (Kinshumann et al., 2011; Lin et al., 2016) but over time researchers noticed the value of information which is hidden inside the log files. Nowadays, log files use for many functions such as performance issues (Nagaraj et al., 2012), system behavior understanding (Li et al., 2017), workload modeling (Abbors et al., 2015), recording the actions of users and investigating malicious activity (Breier & Branišová, 2017), As aforementioned, log files are best source to investigate malicious activity. However, by growing ubiquity of Internet access and emerging new digital devices a medium to large company tends to generate and collect sheer size of activity logs every day. Therefore, analyzing and classifying such huge sets of data manually, for detecting malicious activity or reporting purposes, is tedious and nearly impossible. What is required is automated analysis method which can detect peculiar and

malicious behavior that is unlikely to be spotted by a human (Agrawal & Agrawal, 2015). The intrusion detection (ID) method can be used to detect malicious activity automatically.

2.3 Intrusion Detection

According to National Institute of Standards and Technology (NIST), intrusion detection method is "the automate process of monitoring the events occurring in a computer system or network and analyzing them for signs of intrusions, defined as attempts to compromise the confidentiality, integrity, availability, or to bypass the security mechanisms of a computer or network" (Bace & Mell, 2001). The intrusion detection methods can be categorized as Misuse detection, Anomaly detection and Hybrid techniques.

2.3.1 Misuse Detection

Misuse or Signature-based detection, are designed to analyze system activity, searching for events or groups of events that conform a predefined pattern of events that classify a known attack. Many commercial products use this method which is easy to implement and powerful to detecting known attacks without generating significant amount of false alarms. In addition, system managers can reliably and quickly track malicious activities on their systems, initiating incident handling procedures. Despite the above-mentioned advantages, misuse detection techniques are not able to detect attacks with new pattern. They need constant and quick updating in order to keep up with new and emerging threats, that place an extra workload on security experts. The new attack could cause an enormous damage to the system before a security expert generates a signature for it (Anwar et al., 2017).

2.3.2 Anomaly Detection

Anomaly based techniques model normal network and system behavior, identify anomalies as deviation from normal behavior. This approach can detect unfamiliar attacks therefore make it difficult for attackers to carry out undetected. Moreover, anomaly detection technique produce output which can be employed as information sources for misuse detectors. As a downside, this method is susceptible to high false alarm owing to unpredictable behavior of user and network. Furthermore, defining normal behavior is not easy task and needs extensive "training sets" of system event records to identify normal behavior patterns (Buczak & Guven, 2016).

2.3.3 Hybrid

Hybrid methods combine misuse and anomaly detection. They are used to increase detection rates of known intrusions and decrease the false positive rate for unknown attacks. In anomaly-based category, behavioral based method usually is used to detect insider attack. This method can be grouped in

- System behaviors; and
- User behaviors.

The system behaviors are generated by hosts and networks and relate to the host activities and network status. In contrast, the user behaviors mainly relate to the direct interaction between the user and the system, for example, typing patterns (Peng et al., 2016). In this project our main focus lies on system behaviors method for detecting insider threat by utilizing log files and anomaly-based techniques.

In recent years, anomaly detection methods have been topics of many research and studies. Most successful methods to model the normal behavior of system and detect anomalous or

unexpected behavior is based on machine learning techniques (Parmar & Patel, 2017). Machine learning techniques demonstrate enormous flexibility benefits and have been validated to be accurate in detecting malicious activities, in various cyber security fields like phishing emails (Moradpoor et al., 2017), insider threats (Gheyas & Abdallah, 2016), and malware (Narudin et al., 2016).

2.4 Machine Learning

The machine learning term was used for the first time in 1959 by Arthur Samuel for study of pattern recognition (Samuel, 1988). Machine learning provides systems the ability to automatically learn and improve from experience without being explicitly programmed. It learns to build a model and recommend decision based on audit data. Machine learning is combination of data mining, probability theory, pattern recognition, adaptive control, statistics, artificial intelligence, and theoretical computer science domain (Makani & Reddy, 2018). A machine learning method typically contains of two stages: training and testing. Generally, the subsequent steps are conducted:

- Recognize features from training dataset.
- Select a subset of features which are required for classification (Feature reduction).
- Learn the model by utilizing training dataset.
- Use the trained model to classify the unknown data (test dataset).

In anomaly detection approach first, the normal behavior pattern is learned in the training stage. Second, the learned model is executed to new data, and every exemplar in the testing set is classified as either normal or anomalous (Buczak & Guven, 2016). Machine learning approaches can be divided into three groups:

- Supervised
- Semi-Supervised
- Unsupervised

In supervised learning problems, a training dataset consisting normal and anomalous instances is used to learn a model. The learned model is then applied on the test dataset to classify unlabeled instances into normal and anomalous instances. The second learning problems is semi-supervised. The training dataset containing only normal instances. Thus, the machine learning algorithm models the normal instances only. Instances that do not follow with this model are labeled as anomaly in the testing stage. In unsupervised learning problems, the main task is to find patterns, structures, or knowledge in unlabeled data. The dataset does not contain any labeling information. It assumes that only a small part of the data is anomalous which show significantly different behavior from normal instances (Das & Nene, 2017). Obtaining accurate labeled data that representative of all types of behaviors, is very expensive. This task is time consuming and done manually by human expert (Gogoi et al., 2010). Log files are unlabeled, therefore acquiring labeled log files is very difficult. Furthermore, due to privacy and ethical concerns companies are not interested to share their log files dataset especially the data that may contain insider threats (Greitzer et al., 2010). To address unlabeled log files issue, this study applies unsupervised machine learning approaches.

2.4.1 Machine Learning Algorithms

The most commonly used machine learning techniques for anomaly detection can be divided into three major methods of density-based, distance-based and model-based. These methods are designed based on following concept. Over a feature space, data points that

share similar feature value and are in the majority, are evaluated as normal points. In contrast, data points that are dissimilar to the normal points with regards to feature values over a feature space, and are in the minority, are evaluated as anomalous points (Chandola et al., 2012).

2.4.1.1 Density-based Method

Density-based algorithms design on the principle that data points in low density regions of the feature space are considered as anomalies. Many different algorithms have been developed based on this principal which have different ways to estimate density. Local Outlier Factor (LOF) that calculate LOF value as the sparseness of a point in relation to its local neighborhood. Data points with highest LOF value consider as anomalous points (Breunig et al., 2000). Another algorithm in this line is Connectivity-based Outlier Factor (COF). It uses both low-density points and isolated points to find the anomalous points. This method defines isolativity as the degree of disconnectivity to other neighboring points. However, high isolativity insinuate low density, but low density does not always insinuate high isolativity which cause low detection rate in some dataset (J. Tang et al., 2002). Density – based methods need high computational time therefore, efficiency is never a strength for them. Furthermore, density is defined according to the full dimensional distance computation between two points, which is subjected to the curse of dimensionality (B. Tang & He, 2017).

2.4.1.2 Distance-based Method

Distance-based algorithms design on the principle that anomalous points are data points which occupy a large distance to their neighboring data points. In this method, Euclidean distance is a popular choice to calculate distance measure. Byers & Raftery calculate anomaly score of a data point as its distance to its kth nearest neighbor (KNN) in

data set (Byers & Raftery, 1998). Knorr et al. describe another approach to compute anomaly score of a data point is to count the number of nearest neighbors (n) that are not more than d distance apart from the given data instance (Knorr et al., 2000). To improve the efficiency of distance-based method Wu and Jermaine apply sampling technique. Since, their proposed method selects smaller subsample for a given dataset and compute the nearest neighbor of every points within a subsample, time complexity reduced (Wu & Jermaine, 2006). Although, distance-based method has better efficiency compare to density-based method, but it cannot handle datasets with diverse densities. Also, this method due to use pair-wise distance measure in high dimensional datasets is computationally costly (Shi, 2018).

2.4.1.3 Model-based Method

Model-based algorithms build a model based on data, then data points that do not fit the model accordingly identified as anomalous points (Thearling, 2017). Well-known unsupervised methods are Replicator Neural Network (RNN), One-class SVM and Isolation Forest (iForest). In RNN normal points can construct neural network but anomalous points are unable to construct neural network or are poorly reconstructed (Lu et al., 2017). Another model-based algorithm is one-class SVM, which attempt to select the smallest area with highest number of normal data points. The anomalous points are located outside of this region (Erfani et al., 2016). Since majority of model-based are designed for classification or clustering and are not designed mainly for anomaly detection, their detection performance very depends on how well the data fit into their assumptions. Therefore, they suffer from high false alarms rate and cannot handle high dimensional data very well. In contrast, iForest is designed particularly for anomaly detection and can handle high dimensional data. Furthermore, iForest opposed to distance and density-based methods is able to distinguish

scattered and clustered anomalies (Liu et al., 2008). Consequently, this study concentrates on model-based method for detecting anomalies. Isolation Forest has been selected as machine learning algorithms and One-class SVM as a baseline. In next section this study will describe structure of both algorithms completely.

2.4.2 One-Class SVM

SVM is a supervised machine learning method which was introduces by Vapkin et al. based on Statistical Learning Theory. It learns to differentiate between two classes (Class A and Class B) in a given dataset by using a hyperplane that shows maximum boundary of the separation. Figure 2.2 illustrates how a hyperplane separates the two different classes of A and B (V. Vapnik & Lerner, 1963).

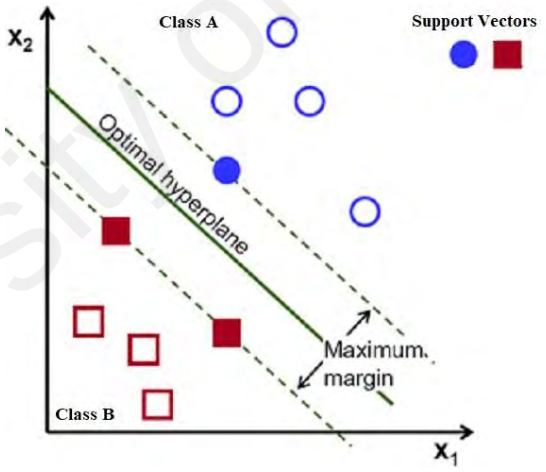

Figure 2.2: SVM classification

SVMs were not able to learn non-linear decision boundaries and were susceptible to outliers. SVMs overcome these problems by mapping method that called Kernel as well as soft margins. Figure 2.3 shows how class A and class B with non-linear boundaries have been separated by applying kernel method (Amer et al., 2013). On the other hand, one-class SVM introduced by Schölkopf et al. in 2001 is an unsupervised anomaly detection method.

Contrary to SVMs, one-class SVMs attempt to fit a hyperplane between data points and the origin (Schölkopf et al., 2001). A one-class SVM uses an implicit transformation function

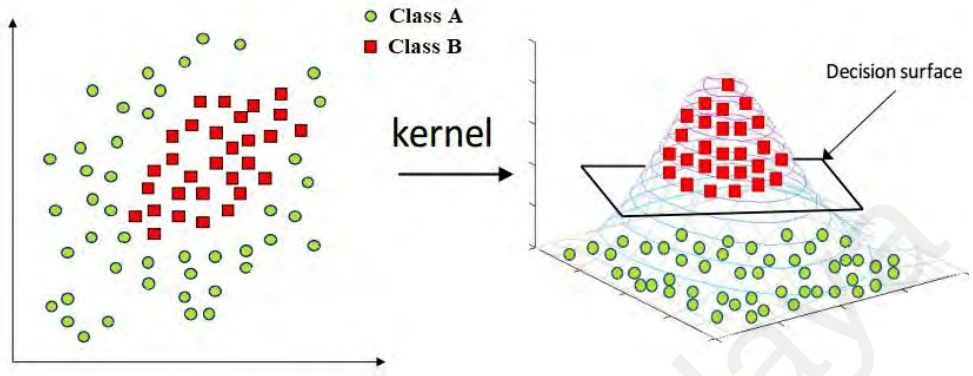

Figure 2.3: SVM for non-linear problems

defined by the kernel to project the data to a feature space with higher dimension. Subsequently, the algorithm fits a hyperplane that separates most of the data from the origin. Only a small part of data points are allowed to sit down on the other side of the hyperplane. Those data points are considered as anomalous. Figure 2.4 shows the one-class SVM model (Cui & Shi, 2014).

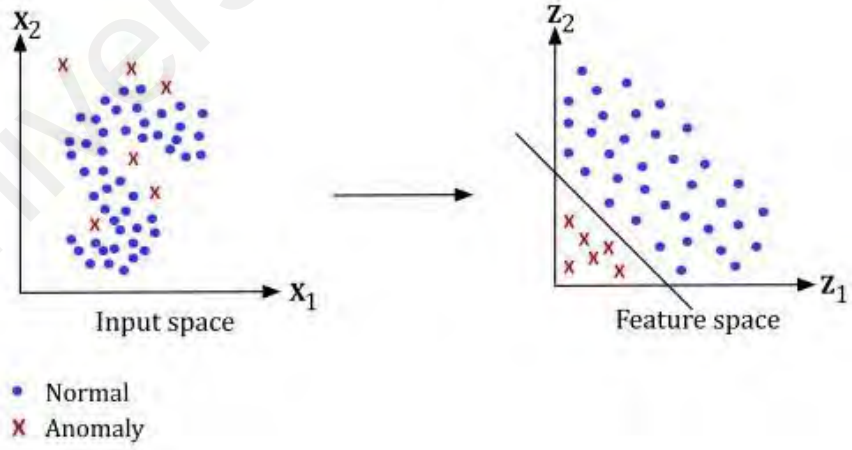

Figure 2.4: The left figure illustrates a dataset in the input space. The right figure illustrates how the data projected to a higher dimensional space by employing one-class SVM algorithm

2.4.3 Isolation Forest

Isolation forest is an unsupervised algorithm which was introduced by Liu et al. in 2008. It was mainly designed for anomaly detection proposes. Isolation Forest does not use common methods such as density or distance to detect anomalies therefore, needs less computations compare to distance and density methods. The algorithm is designed by focusing on two important characteristics of anomaly data: i) The anomalous points are small fraction of whole size of the dataset. ii) Their attribute-values are significantly different from the normal data points. As a result, anomalies points are more susceptible to isolation than the normal points, which is the key concept behind the iForest design. The iForest for a given dataset constructs an ensemble of itrees which are binary decision trees. The data is recursively partitioned until iTree differentiate each data points from other points. Since anomalous points are significantly sensitive to separation, they are closer to the root of an iTree, while the normal points are further from the root. Therefore, anomalous points need smaller number of characteristic conditions to be isolated as illustrates in Figure 2.5 (Liu et al., 2012).

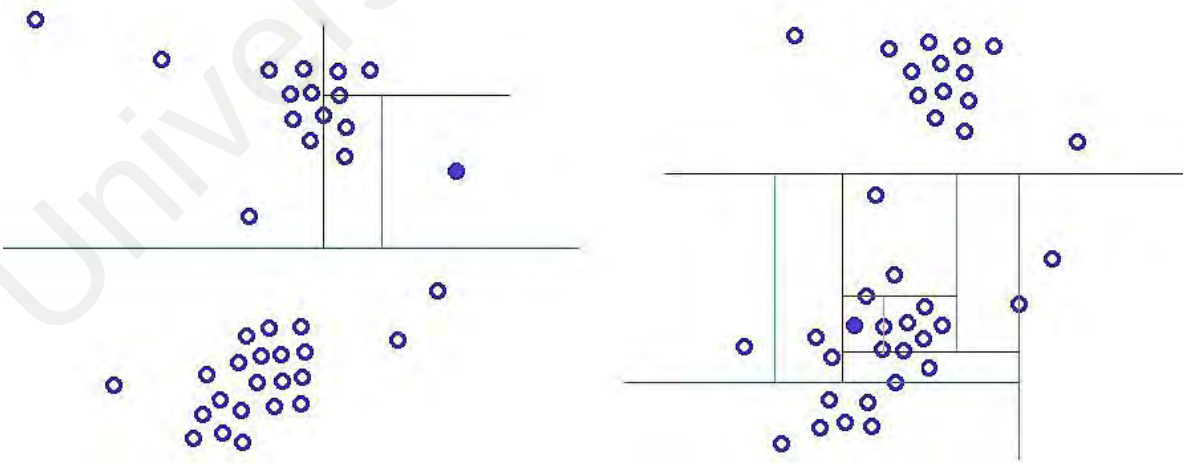

Figure 2.5: The figure on the right illustrates the normal point be isolated after 9 random partitions, on contrary figure on the left illustrates anomalous point needs only 4 random partitions.

Definition: Isolation Tree. Let T be a node of an isolation tree. T is either an external-node with no child, or an internal-node with one test and exactly two daughter nodes (T_l, T_r). The data points can be divided to T_l and T_r based on attribute value and split value.

Let $D = \{d_1, d_2 .... d_n\}$ be the given data set. To achieve the diversity of iTree, a subsample $D' \subset D$ with size of ψ is used to construct iTree. Select randomly an attribute A with split value of P from D′, next split each data object di by the value of its attribute A which is called di(A). The data point is left in T_l if di(A) < P and vice versa. The subtrees are built iteratively until either: i) There is only one instance in the D′ or ii) all data at the D′ have identical values.

Definition: Path Length h(d) of a point d is measured by the number of edges d traverses an iTree from the root node until the traversal is terminated at an external node. Path length shows the degree of susceptibility to isolation. In other word, anomalous points have short path length and normal points have long path length. Since iTrees have same structure as Binary Search Tree, the estimation of average h(d) for external node terminations is equivalent to the failed query in Binary Search Tree. It was calculated as follow by Preiss (Preiss.BR 2000)

$$c(\psi) = \begin{cases} 2H(\psi - 1) - 2(\psi - 1)/n & \text{for } \psi > 2, \\ 1 & \text{for } \psi = 2, \\ 0 & \text{otherwise.} \end{cases} \quad (2.1)$$

where H(i) is the harmonic number and it can be estimated by ln(i) + 0.5772156649 (Euler's constant). As c(ψ) is the average of h(d) given ψ, author uses it to normalize h(d). The anomaly score s of an instance d is defined as:

$$s(d, \psi) = 2^{\frac{-E(h(d))}{c(\psi)}} \quad (2.2)$$

According Equ.2 when $E(h(d)) \to 0$, $s \to 1$ means the data points are anomalous when s is close to 1; when $E(h(d)) \to \psi - 1$, $s \to 0$ means the data points are normal when s is close to

0; when $E(h(d)) \rightarrow c(\psi)$, $s \rightarrow 0.5$ means there is no obvious anomalous point in the entire sample. Isolation Forest has a low linear time-complexity of $O(t(n + \psi)\psi)$, where t represents number of trees, ψ subsampling size and n is number of data points in a data set. Liu et al. compare iForest with four state-of-the-art anomaly detection algorithms on different datasets. Their result shows iForest achieved higher detection accuracy and need less computational time as well as memory requirment particulary in large dataset due to its design which does not use density or distance measurment (Liu et al., 2012). As aforementioned density and distance based method are not efficienc and are computationally expensive. Furthermore, iForest is capable of handling high dimentional data with irrelevent features, which make it suitable for log files dataset. In addition, iForest are robust with the masking and swamping effects (Xu et al., 2017) . Swamping cites to cases when normal points are wrongly identified as anomalous. It occurs when normal points are more scattered or there are a lot of normal points. In contrary, masking refers to presence of many anomalous points which hiding their own existence. It occurs when clusters of anomalies are dense and large. Many anomaly detection methods cannot handle these situations. Swamping and masking effects are happening due to presence of many data for anomaly detection (Chiang, 2008). As we know, log files dataset is huge in size and number of anomalous points are extremely low thus, log files dataset is susceptible to swamping effect. The iForest algorithm by utilizing multiple sub-samples reduce the effects of swamping and masking. A small size subsample constructs a better performing iTree rather than whole data set. Subsamples have fewer normal points which interfering with anomalies; therefore, anomaly points can be isolated easier. In literature reviewed, iForest algorithm shows strong ability to detect anomalies in big datasets with high dimension features, therefore this study selects iForest as machine learning algorithm.

2.5 Feature Reduction

The problems of high dimensional data for the first time was introduced by Richard Bellman as "the curse of dimensionality" (Bellman & Corporation, 1957). These terms refer to organizing and analyzing of data which have hundreds or thousands of dimensions (features). Most of these features are irrelevant or redundant and may lead to complexity, overfitting, low accuracy and higher computational cost (Guyon et al., 2006). Log files which are collected by companies have a lot of irrelevant and redundant features that act as noise and cause above -mentioned problems. To mitigate this problem feature reduction techniques such as feature extraction and feature selection could be used. In feature extraction approach, features are projected into a new space with lower dimensionality. These algorithms try to extract features capable of reconstructing the original high dimensional data. In contrast, the feature selection approach aims to select a small subset of features by assign each feature a value of importance, which is used to filter the set of features (Krishnan & Athavale, 2018; Wang et al., 2016). Figure 2.6 illustrates difference between feature selection and feature extraction. These techniques are generally used as preprocessing to machine learning.

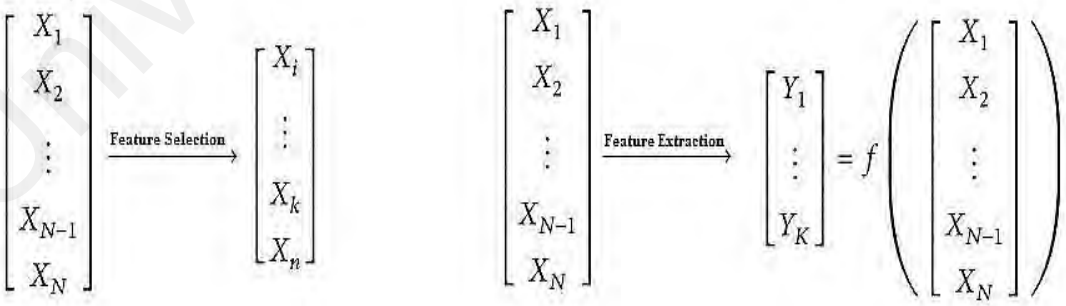

Figure 2.6: The left matrix shows feature selection and the right one is feature extraction.

According Hira et al. study, feature selection methods preserving data characteristics for interpretability, but they are more susceptible to overfitting and have less discriminative power. In the other hand, the feature extraction methods have higher discriminating power and in unsupervised state can control overfitting, also reduce the number of features without losing information. As a downside it may loss data interpretability due to transformation (Hira & Gillies, 2015).

2.5.1 Feature Extraction

This study uses feature extraction method owing to higher discriminating power, overfitting control and absence of data loss. Feature extraction algorithms are designed based on spectral decomposition theorem, therefore all need the construction of a matrix that encodes global and/or local relations between data points. The most versatile algorithms that are used as feature extraction are principal component Analysis (PCA) and linear discriminant analysis (LDA). The LDA extracts features based on maximizes the separability between classes. In the other word, LDA algorithm is a supervised method which needs labeled data (Pölsterl et al., 2016). Since log files are unlabeled, in this study PCA which is an unsupervised algorithm is used for feature extraction.

2.5.1.1 Principal Component Analysis (PCA)

The concept of PCA was proposed by Pearson in 1901 (Pearson, 1901) but it was used as feature extraction algorithm for the first time in 1978 by Kruskal (Kruskal & Wish, 1978). Since then, PCA have been employed extensively in feature extraction methods. This algorithm recognizes the data points with highest possible variance. PCA constructs a new set of variables by transforming correlated variables into a set of linearly uncorrelated variables. This new set which is called principal components, has lower dimensionality which

decrease computational costs (I. Jolliffe, 2011). Figure 2.7 attempts to describe the PCA in a simple way. Suppose have a simple object with a complex set of variables (or coordinate system) (A). PCA algorithm finds new variables (coordinate axes) orthogonal to each other and pointing to the direction of largest variances. C1 is the direction of largest variance and C2 is direction of second largest variance (B). Utilize new set of variables (coordinate axes) to describe object in a more concise way (C).

The PCA method can be encapsulate in six steps. Suppose the dataset has high dimensions of d and m is the number of dimensions of the new set.

1. Calculate the covariance matrix of the normalized d-dimensional dataset.
2. Calculate the eigenvectors and eigenvalues of the covariance matrix.
3. Sort the eigenvalues in descending order.
4. Choose the m eigenvectors that correspond to the m largest eigenvalues.
5. Create the projection matrix from the m selected eigenvectors.
6. Transform the original dataset to build a new m-dimensional feature space (I. T. Jolliffe & Cadima, 2016).

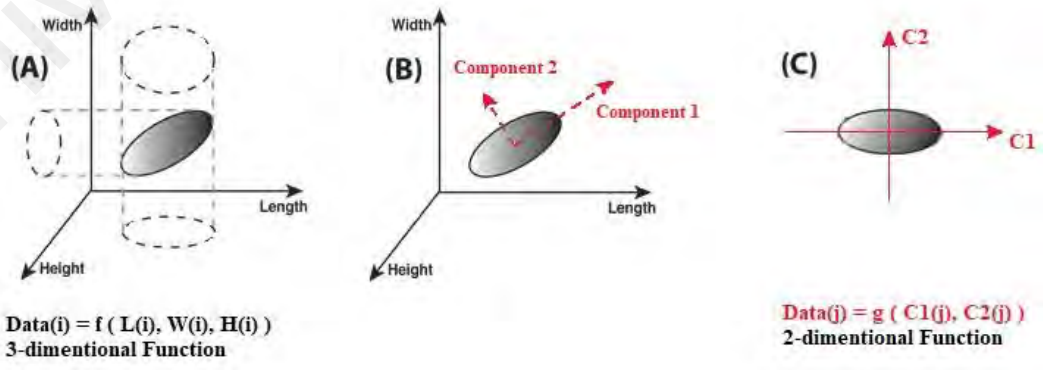

Figure 2.7: PCA algorithm

The Lakhina et al. investigate the effect of PCA on detecting anomalies in network traffic. Their method consistently detected the largest number of anomalies with a very low false alarm (Lakhina et al., 2010). In other study, Rubinstein et al. highlight the advantage of PCA algorithm as an unsupervised method which can be used in anomaly detection methods without presence of labeled data (Rubinstein et al., 2008). In the similar line, Camacho et al. show how tuning the number of principal components in PCA reduce the noise in their model and improve anomaly detection capability (Camacho et al., 2016). Wickramasinghe applied PCA along with other algorithms such as SVM and RF on different dataset with different level of noise. Their experiment shows PCA handle high level of noise and it is suitable for dataset with redundant and irrelevant data (Wickramasinghe, 2017). Ikram & Cherukuri propose a hybrid intrusion detection model by integration of PCA and support vector machine (SVM). Their method achieve higher detection rate and decrease in computational time by removing redundant and irrelevant features (Ikram & Cherukuri, 2016). Thaseen & Kumar design an intrusion detection model base on PCA and ensemble classifiers. Their experiment on two different dataset confirm the accuracy improvement by applying PCA (Thaseen & Kumar, 2016). The literature reviewed, reveals that using PCA feature extraction technique produces promising results with regards to detection rate, accuracy and computational time. This method also provides the added benefit of not requiring labelled data.

2.6 Feature Engineering

Log files are heterogenous which means contain variety or dissimilar type of data. Thus, they cannot be fed directly as input to machine learning algorithm. Furthermore, having discriminative and independent features is a key component to achieve highly accurate classifiers. As a result, log files require feature engineering process. The process that generate

features from raw data is known as feature engineering (Correa Bahnsen et al., 2016). However, many of previous researchers focus on model design and feature engineering simply being overlooked. Rashid et al. use CERT dataset and generate one simple set includes 7 features of logon, log off, file copied to removable drive, send an email, connect a removable drive and disconnect a removable drive. They aggregate data based on monthly activities of each user. Their result shows the number of generated features are not enough and need more features to improve detection rate (Rashid et al., 2016). Gavai et al. generate 42 features from real-world Vegas dataset over five domains of email usage frequency, email content, log-on log-off behavior, application activity and web activity. They collect data based on daily activities of each user. They had not access to some of data such as size of attachments in e-mails due to privacy and ethical concerns. They highlight generating more independent features could lead to higher detection rate (Gavai et al., 2015). In the similar fashion, Legg et al. generate 168 features from CERT dataset based on log-on log-off time, email usage, web activity, file activity, removable driver usage and decoy files. They define a series of features that assess the hourly and daily usage counts for each device, activity, and attribute such as Hourly usage count for device/activity/attribute and Daily usage count for device/activity/attribute. Furthermore, they define time-based features for each device, activity, and attribute like Latest logon time for user, Earliest USB time for user, and USB duration for user. They also show some of the generated features are correlated and need feature extraction (Legg et al., 2015a). This study follows the Legg et al. method and try to generate discriminative and independent features.

2.7 Dataset

Acquire and research with real-world data is challenging. Majority of companies are not interested to share their dataset especially the data that may contain insider threats, due to privacy and ethical concerns. Therefore, real-world data should go under anonymization process before being available as a public dataset. Anonymization is the process of encrypting or removing user recognizable information from data sets, therefore the user whom the data represent remain anonymous. As a drawback, some relevant factors in system log files such as user attribute metadata may have concealed during anonymization process (Gentili et al., 2017). Gheyas & Abdallah categorized different data source which was used for detecting insider threat by researcher as below:

- Real-world system log data
- Social media data
- Simulated data drawn from stochastic models
- Real data injected with synthetic anomalies.
- Simulated data drawn from stochastic models which are developed from real data
- Game-theoretic approach (GTA)

The primary sources of research data are the simulated data drawn from stochastic models which are developed from real data. These kind of datasets containing complete user information despite anonymized real-world dataset (Gheyas & Abdallah, 2016). The CERT data set is one of the well-known simulated datasets which is used widely for detecting insider threats. The CERT dataset simulated organization's computer network and generated log files record for employees (Glasser & Lindauer, 2013). Since this dataset includes all information about users, this study use CERT Insider Threat Dataset v6.2. The CERT dataset will be described completely in Chapter 3.

2.8 Related Works

In recent years, anomaly detection has been an important research problem in security analysis, thus researchers try to develop different methods to detect malicious insider threat with high detection rate and low false alarm (Bohara et al., 2016).

In this problem layout, Gavai et al. compared a supervised approach with an unsupervised approach using the iForest method and RF for detecting insider threat from CERT dataset which is a synthetic dataset includes five different domains such as file, logon/logoff, http, device and email. They generated in total 42 features and did not use any feature reduction method. Their proposed method showed 76% and 73.5 % AUC for iForest and RF, respectively. They investigated how strongly the quitting and insider events are correlated. Since their detection rate was not high, they highlighted by adding more discriminative features can achieve higher detection rate (G. Gavai et al., 2015).

Karev et al. presented work using iForest in an online setting. They aggregate HTTP log data to explore the algorithm's accuracy under various conditions. They used generic algorithm as feature selection method to find the best HTTP features which can differentiate between malicious and normal data. Their subsample includes 20 features. They tried different tuning parameters for iForest. They achieved 82% AUC by tuning the iForest algorithm parameters with number of iTrees t=200 and subsampling size ψ =8192. Their dataset was not high dimensional dataset and by using feature selection they might lose some informative features (Karev et al., 2017).

Tuor et al. implemented online unsupervised deep learning approach to detect anomalous network activity from system logs in CERT dataset. They modeled the stream of system logs as interleaved user sequences based on daily activity and used deep neural network (DNN) and recurrent neural network (RNN) algorithms to detect insider threats.

They used iForest as base line and used default tuning parameters t=100 and ψ =256. In total they generated 414 features and not apply any feature reduction method. Their model showed promising result to detect insider threats with 90% recall. They excluded weekends and holidays activity from dataset that may cause more false alarm but there is a great chance that attacks happen during weekend or holidays (Tuor et al., 2017).

McGough et al. designed a system to identify anomalous behavior of user by comparing individual user's activities against their own routine profile, as well as against the organization's rule. They applied two independent approaches of machine learning and statistical analyzer on data. Then results from these two parts combined to form consensus which then mapped to a risk score. They used synthetic dataset consists of five different domains such as file, logon/logoff, http, device and email with 24 features. They applied supervised Support Vector Data Description (SVDD) algorithm for machine learning. Their system showed high accuracy and minimum effect on the existing computing and network resources in terms of memory and CPU usage. Average of false positive was about 0.4. The rate of false positive alarm was depending on risk threshold (McGough et al., 2015). Some researcher suggested finding the relationship between user and his colleagues with similar roles may help to have better understanding of normal behavior of an employee.

Bhattacharjee et al. proposed a graph-based method that can investigate user behavior from two perspectives: (a) anomaly with reference to the normal activities of individual user which has been observed in a prolonged period of time, and (b) finding the relationship between user and his colleagues with similar roles/profiles. They utilized CERT dataset in unsupervised manner. They generated 23 features and tried to find best subset based on relationship between user and his colleagues with the same role. In their model, Boykov Kolmogorov algorithm was used and the result compared with different algorithms including

One-Class SVM, Individual Profile Analysis, k-User Clustering, and Maximum Clique. Their proposed model showed 95% AUC that was much higher compare to other algorithms (Bhattacharjee et al., 2017). Their method was computationally expensive particularly for companies with a lot of employees.

Normally, log files include a lot of irrelevant data which act as noise therefore, using feature reduction methods can help to improve the result. Legg et al. offered an automated system that construct tree structured profiles based on individual user activity and combined role activity. This method helped them to attain consistent features, which provide description of the user's behavior. They generated 168 features so to reduce redundant and irrelevant features they applied feature extraction PCA method with only two dimensions. Then, they computed anomaly scores based on different metrics of standard deviation, mahalanobis and covariance distance and finally calculate anomaly score by averaging these three metrics. Their system was tested on synthetic dataset which ten malicious data injected. Their system found seven out of ten insider threats. They emphasized due to visualization constrains had to choose PCA with two dimensions (Legg et al., 2015b).

In a similar line, Agrafiotis et al. applied same model as offered by Legg et al., instead of synthetic data they used real -world data set from multinational organization contains of five different domains such as email, logon/logoff, http, device and email. Their approach abided the ethical and privacy concerns. They used PCA as a feature reduction method. Their result showed by using PCA false alarm 33% decreased compare to not applying PCA (Agrafiotis et al., 2015). However, they used PCA with three dimensions which did not add enough variation to the data.

Although finding a sequence is a common choice for modeling activities and events through time but catching anomalous sequence in a dataset is not an easy task. One of the

widely used algorithms that has ability to recognize temporal pattern is Hidden Markov Models (HMM). Rashid et al. proposed a model based on HMM to identify insider threat in CERT dataset. They generated 16 features. They tried to model user's normal behavior as a week-long sequence. Their modeled showed 83% AUC with false positive rate of 20%. The authors mentioned using shorter time frame for instance a day long sequences could build a more accurate model of employee's daily behavior. Moreover, their system was trained based on first 5 weeks, so it is unable to detect insider threats amongst short-term users such as contractors whose are a real threat. They accentuated the generating discriminative features, number of features and selecting appropriate algorithm hyperparameters play important role to improve detecting rate (Rashid et al., 2016). One of the common problems in temporal anomaly detection mechanism is flagging common changes mistakenly as attack therefore to avoid this kind of issue, Eldardiry et al. proposed a clustering model which form different clusters based on user behavior and peer groups for each day for five different domains such as file, logon/logoff, http, device and email. Then, they model user behavior over time as a Markov sequence, where a user will belong to one cluster each day, and transition between clusters each day. They calculated anomaly score for each user so, user with the rarest transitions compared to her/his peers will be the most suspicious. They tested their model on a synthetic system log file. Their result showed 90% detection rate for each domain. In their study, the users were ranked separately within each domain and they did not rank users based on the entire domains therefore, their method might not be appropriate for detecting complex attacks (Eldardiry et al., 2013). Table 2.1 shows the summary of related works.

Table 2.1: Summary of related works

Author	Dataset	Method	Machine learning algorithm	Result	Limitation
Gavai et al., 2015	CERT* dataset	Supervised & Unsupervised	**ML:** iForest, RF	**iForest** :76% AUC **RF**: 73.5 % AUC	Based on AUC point system their detection rate was fair and need improvement by adding more discriminative features
Karev et al., 2017	HTTP log data	Unsupervised	**ML:** iForest **Feature selection:** GA	82% AUC	Their dataset was not high dimensional dataset and by using feature selection they might lose some informative features
Tuor et al., 2017	CERT* dataset	Unsupervised	**ML:** RNN, DNN & iForest	**RNN:** 90% TPR **DNN & iForest** Less than 90% TPR	They excluded weekends and holidays activity from dataset where there is a great chance that attacks happen during weekend or holidays
McGough et al., 2015	Synthetic* dataset	Supervised	**ML:** SVDD	**Average of FPR:** 0.4	The rate of false positive alarm was depending on risk threshold
Bhattacharjee et al., 2017	CERT* dataset	Unsupervised	**ML:** Boykov Kolmogorov	95% AUC	Their method was computationally expensive particularly for companies with a lot of employees.

*All datasets include five different domains such as file, logon/logoff, http, device and email.

Continue Table 2.1

Author	Dataset	Method	Machine learning algorithm	Result	Limitation
Legg et al., 2015	CERT* dataset	supervised	**Feature reduction:** PCA **Anomaly metrics:** Standard Deviation, Mahalanobis and Covariance Distance	Found 7 out of 10 insider threats	They had to choose PCA with two dimensions due to visualization constrains which did not add enough variation to the data.
Agrafiotis et al., 2016	CERT* Dataset & Real-World dataset	Supervised	**Feature reduction:** PCA **Anomaly metrices:** Standard Deviation, Mahalanobis and Covariance Distance	With using PCA 33% decrease in FPR	They used PCA with three dimensions which did not add enough variation to the data.
Rashid et al., 2016	CERT* dataset	Supervised	**ML:** HMM	83% AUC FPR 20%	Their system was trained based on first 5 weeks, so it is unable to detect insider threats amongst short-term users such as contractors whose are a real threat and does not add enough granularity to their model
Eldardiry et al., 2013	Synthetic* dataset	Supervised	**Clustering:** K-mean **Anomaly metrics:** HMM	90% AUC	In their study, the users were ranked separately within each domain and they did not rank users based on the entire domains therefore, their method might not be appropriate for detecting complex attacks

*All datasets include five different domains such as file, logon/logoff, http, device and email.

2.9 Summary

This chapter discussed the concept of log files, intrusion detection, anomaly detection, machine learning algorithm, feature reduction methods, dataset and finally reviewed related works that design and implement models to detect anomalies in system log files. Additionally, this chapter reviews the importance of feature engineering. It was found that discriminative and independent features lead to higher detection rate. The literature demonstrates the commonly used unsupervised feature extraction is PCA which can significantly reduce noise and irrelevant data and improve detection rate, accuracy and computational time. As mentioned throughout literature review, there are some limitations with the current research works. This work tries to overcome the weaknesses mentioned in the related works and achieve better results and improvements. As an instance, some works only used PCA with low dimensionality that reduce variability of data. we use PCA with more dimensionality to achieve better variability on dataset and evaluate its effectiveness. Some other works studied only weekdays activity but there is a chance malicious activity happen during weekends or holidays so, this work includes all weekdays, weekends and holidays activities. This study aims to build model based on daily activities since monthly base not able to detect short term or contractor users. In addition, this study tries to use different algorithms along with PCA to achieve higher AUC.

CHAPTER 3: Methodology

3.1 Introduction

This chapter presents the methodology that is used for this thesis and discuss how this work was down in details. The chapter is organized into nine sections. Section 3.4 and 3.5 explains the details of data set and how data aggregated based on user daily activities. Section 3.6 describes list of engineered features generated from the raw dataset. Section 3.7 discuses about feature extraction and the optimum number of features. Section 3.8 interprets the machine learning algorithms involved in this study. Section 3.9 summarizes the chapter with concluding remarks.

3.2 General Overview

This thesis aims to study detection of anomalies in system log files by using machine learning techniques. Figure 3.1 illustrates the study workflow. There are five different processes: data aggregation, feature engineering, feature extraction, and finally machine learning algorithms and anomaly detection. In data aggregation for each user data is aggregated based on his/her daily activities. In the next step, independent and appropriate features are generated. Afterwards, the feature extraction is used to reduce high dimensionality of data set and in the last process the prepared dataset is fed to machine learning algorithms to detect anomalies. The processes are described in detail in the following sections.

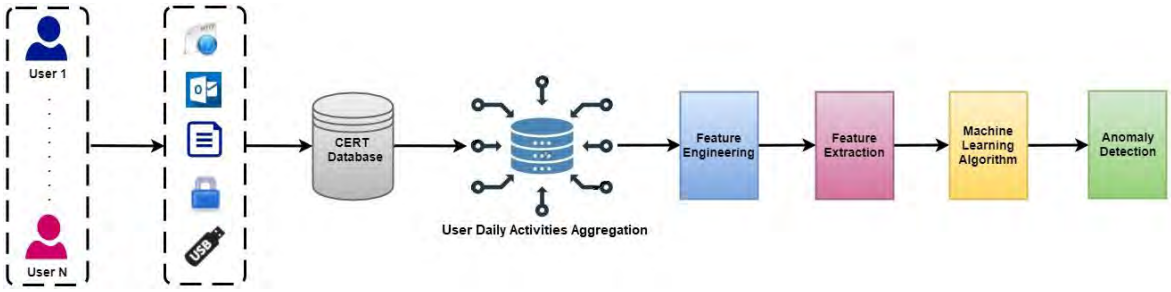

Figure 3.1: Process Workflow

3.3 Experimental Setup

The following section describes hardware requirement and number of tools have been used to carry out this study. Each part explains about hardware, tools and its application.

3.3.1 Hardware Requirement

This experiment has been performed on a machine with an Intel Core i7 CPU at 2.90 GHz, 8 GB RAM and RAM frequency of 2400.0 MHz. The operating system of this machine is Microsoft Windows 10.

3.3.2 Tools

The tools require to run this experiment are described as follows.

3.3.2.1 Microsoft SQL Server 2017

Microsoft SQL Server is a relational database management system (RDBMS) that developed by Microsoft in 1989. The primary function of this product is storing and retrieving data as required by other software applications. It can be run either on the same computer or on another across a network. Structure Query Language (SQL) are used for queries as well as reporting. Figure 3.2 illustrate Microsoft SQL Server stored the CERT dataset.

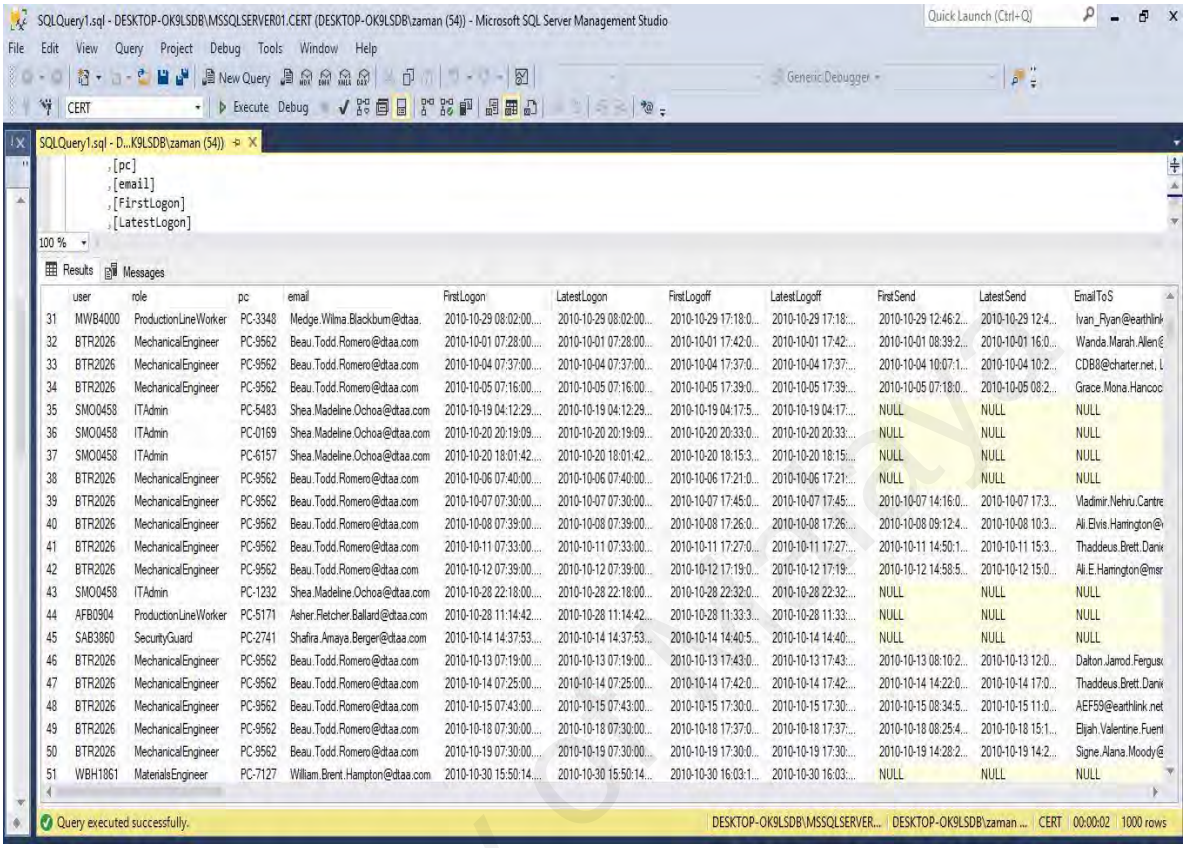

Figure 3.2: The Microsoft SQL Server environment

3.3.2.2 R Studio V1.1.383

RStudio is an integrated development environment (IDE) for R programming language that is written in C++ and java script. It is part of R community, free and open source that was founded by JJ Allaire in 2011. It uses for statistical computing and graphics that includes tools for data preprocessing, machine learning, and visualization. RStudio is an exquisite choice since it has tools for workspace management, debugging and plotting also it provides graphical user interface. Figure 3.3 shows the RStudio software environment to join tables from different months.

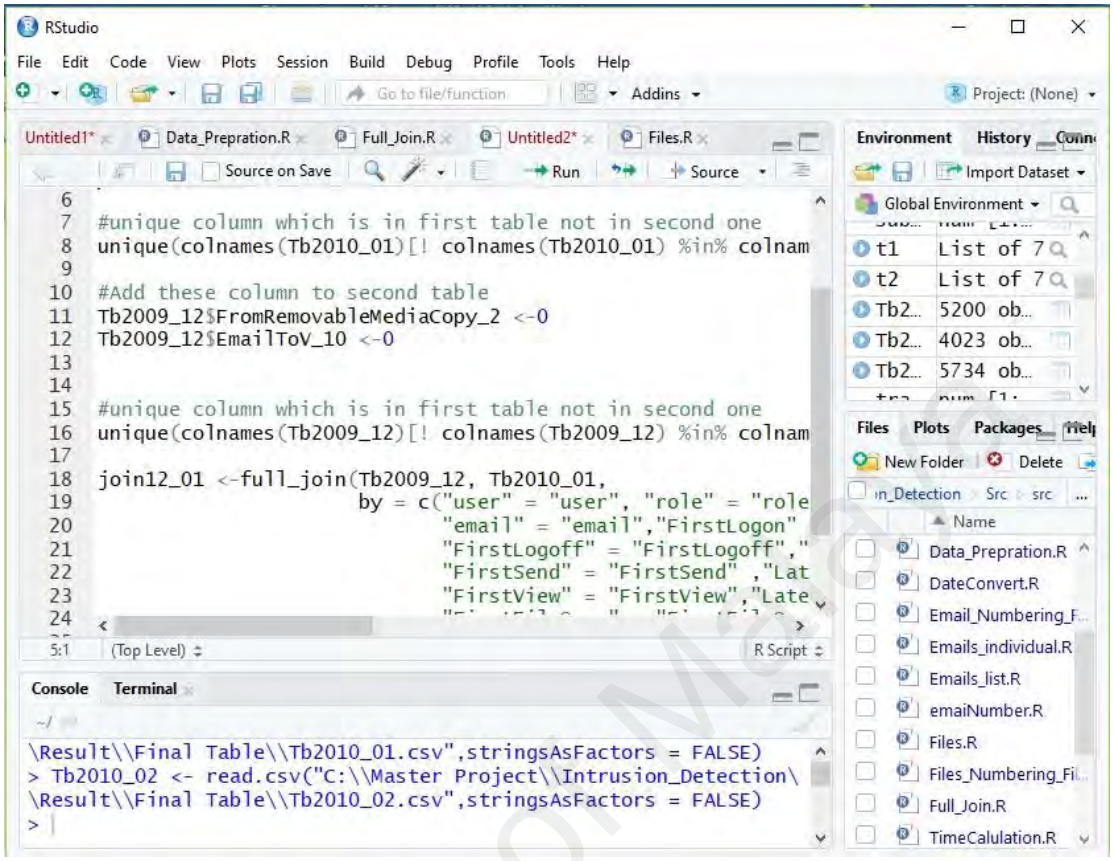

Figure 3.3: The R Studio environment

3.3.2.3 IBM SPSS V.24

SPSS is stand for Statistical Package for the Social Science. It is very popular in statistical analysis due to its simplicity to manipulating and analyzing high complex data. The SPSS software was released in 1968 by SPCC Inc. Later, in 2009 it was acquired by IBM . It was written originally in Fortran but from version 16.0 they used Java. It can read different types of files such as spread sheet and data base software. It includes many statistical functions such as Descriptive Statistics, Reliability tests, Correlation, T-tests, ANOVA and many more. Figure 3.4 illustrates the SPSS software environment as the result file was imported for computing Confidence Interval.

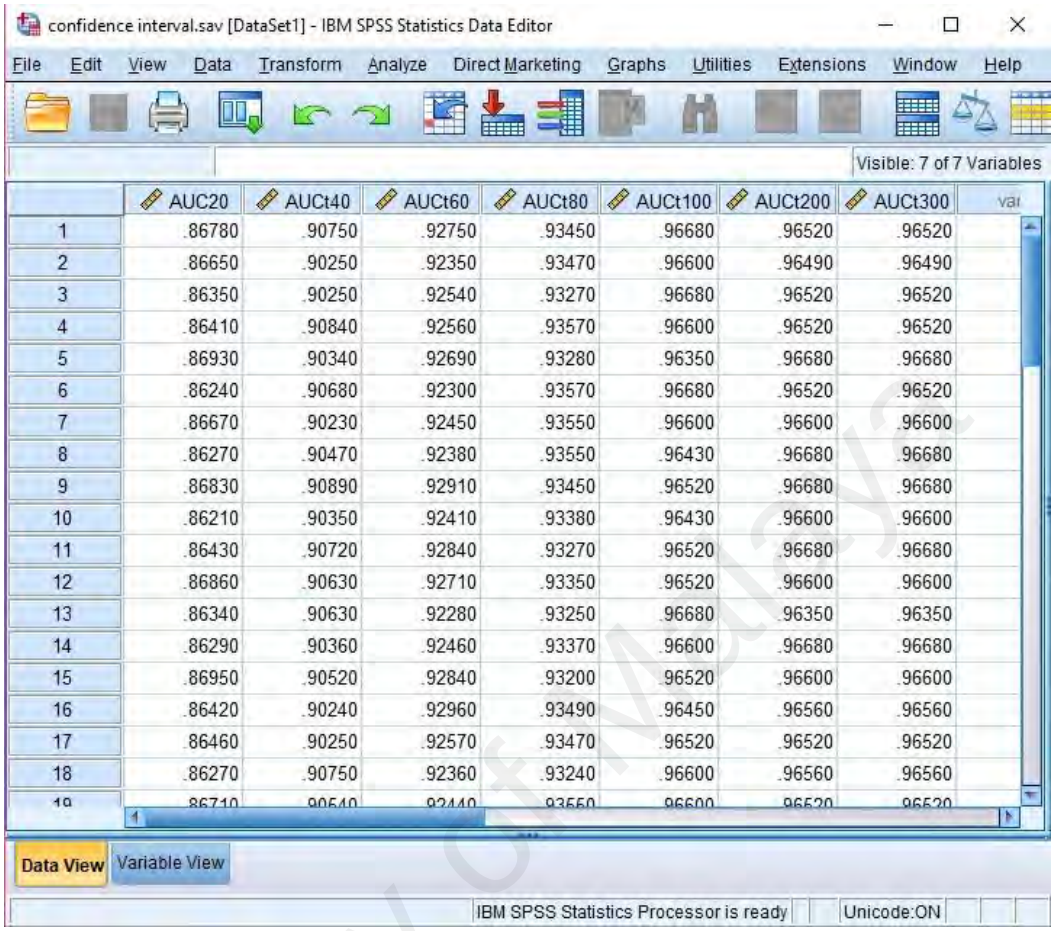

Figure 3.4: The IBM SPSS environment

3.4 CERT Dataset

As discussed completely in Chapter 2, obtain and research with real-world data is challenging. Most of companies are not interested to share their dataset especially the data that may contain insider threats, due to privacy and ethical concerns. Therefore, real-world data should go under anonymization process before being available as a public dataset. As a drawback, some relevant factors in system log files such as user attribute metadata may have concealed during anonymization process.

This study uses the synthetic CERT Insider Threat v6.2. dataset. It consists of event log lines from a simulated organization's computer network, generated with sophisticated user models. It includes system logs for 4000 employees over a 516 days period. Total of 5 insider threats were injected in dataset.

The CERT dataset includes different tables:

- LDPA (User monthly information)
- Logon
- Device
- File
- Http
- Decoy
- Email

In following part each table is described in detail. The unique value between all tables was User.

3.4.1 LDPA

It contains 18 tables from 2009-12 until 2011-05 which includes every employee information for each month. Each table consists fields of Employee name, email, role, projects, business unit, department, team, and supervisor. Table 3.1 shows an example of user monthly information for one user.

Table 3.1: LDPA features

Features	
User	NFP2441
Email	Nicholas.Fletcher.Pruitt@dtaa.com
Role	ITAdmin
Project	Project 31
Business Unit	1
Functional Unit	1 - Adminstration
Department	5 - Security
Team	8 - ElectronicSecurity
Supervisor	Madison Charissa Malone

3.4.2 Logon/Logoff

This table illustrates logon and logoff activity for each user. It consists fields of date, user, pc, and activity (Logon/Logoff). Table 3.2 shows an example of logon and logoff for one user.

Table 3.2: Logon/Logoff features

Features		
Date	1/4/2010 8:32:55 AM	1/4/2010 9:50:58 AM
User	NFP2441	NFP2441
PC	PC-2051	PC-2051
Activity	Logon	Logoff

3.4.3 Device

This table illustrates the thumb drive usage. It consists fields of date, user, pc, file_tree, and activity (connect/disconnect). Table 3.3 shows an example of thumb drive usage for one user.

Table 3.3: Device features

Features		
Date	1/4/2010 9:10:18 AM	1/4/2010 9:37:13 AM
User	NFP2441	NFP2441
PC	PC-2051	PC-2051
File_tree	R:\; R:\22B5gX4; R:\SDH2394;	
Activity	Connect	Disconnect

3.4.4 File

This table illustrates activity (open, write, copy or delete) involving a removable media device. It consists fields of date, user, pc, filename, activity (open, write, copy or delete), to_removable_media, from_removable_media, and content. Table 3.4 shows an example of file activities for one user.

Table 3.4: File features

Features	
Date	1/4/2010 9:19:41 AM
User	NFP2441
PC	PC-2051
File name	R:\SDH2394\7XRCV2N5.pdf
Activity	File Copy
to_removable_media,	TRUE
from_removable_media	FALSE
Content	25-50-44-46-2D Although he restored some of the lands that…….

3.4.5 Http

This table illustrates activity "WWW Download", "WWW Upload", or "WWW Visit" involving surfing the Internet. It consists fields of date, user, pc, URL, activity (WWW Download, WWW Upload, or WWW Visit), and content. Table 3.5 shows an example of web exploring for one user.

Table 3.5: Http features

Features	
Date	1/4/2010 8:45:13 AM
User	NFP2441
PC	PC-2051
URL	http://cbssports.com/Trial_by_Jury/prichole/Wbuafba_Perrx_ _EvireYFJE_A15_pynff2033164944.php
Activity	WWW Visit
Content	were withdrawn from Basingstoke shed, with No. 30738 "King Pellinore" the final example to cease operation in March 1958…

3.4.6 Decoy

A list of decoy files and the hosts on which they reside. It consists fields of decoy_file name, and pc. Table 3.6 shows an example of decoy files.

Table 3.6: Decoy features

Features	
Decoy_ filename	C:\LJE2413\795JW126.jpg
PC	PC-0102

3.4.7 Email

This table illustrates activity (send or view) for email. It consists fields of date, user, pc, to, cc, bcc, from, activity (send, view), size, attachments, content. Table 3.7 shows an example of email activities for one user.

Table 3.7: Email features

Features	
Date	1/4/2010 9:30:08 AM
User	NFP2441
PC	PC-2051
To	Benjamin.Phillip.Dyer@dtaa.com, Justina.Patricia.Short@dtaa.com, Casey.Amery.Gutierrez@dtaa.com,
CC	Richard.Matthew.Odonnell@dtaa.com
Bcc	-
From	-
Activity	send
Size	7854059
Attachment	C:\39f28L6\NRIWY3BQ.doc(1456240); C:\76w5237\O6Q3FKGT.pdf(164742); C:\MNW2267\FF26YMGS.pdf(539516); C:\U6W5HSDR.doc(624204);
Content	These files are related to our new project that will be discussed in our next meeting …….

3.5 Data Aggregation

It can be seen every part of information for each user is in different tables, our aim is to have all of user activities for each day in one row. To achieve this, needs two steps process. In the first step, all tables were joined together based on model design in Figure 3.5 by

utilizing Microsoft SQL Server 2017. All of data was grouped based on user daily activities. In other word, each user has different number of rows that represent his/her daily activities. In the next step, all data from these rows was aggregated to form only one row. Thus, this study created new dataset that each row represents user daily activity. Some users might have not done some activities during a day for instance using external driver or send an email, therefore those activities marked by NULL. Later, all NULL replaced by zero.

3.6 Feature Engineering

As mentioned in Chapter 2 section 2, log files are heterogenous and cannot be used directly as input for machine learning. This study needs feature engineering process to generate features from raw data in our new dataset. In this stage, we generated discriminative and independent numeric features as much as possible. We engineered features base on employee's daily activities over five domains: logon/log off behavior, email usage, web activity, external storage device usage and file operation. In order to have more granularity, this study splits 24 hours into four equal parts start from 6am which call it time period. For example, number of files copied between 6am-12pm/12pm-6pm/6pm-12am/12am-6am. It generated in total 200 activities that a user has performed in 24 hours. Some example of the engineered features is listed in Table 3.8.

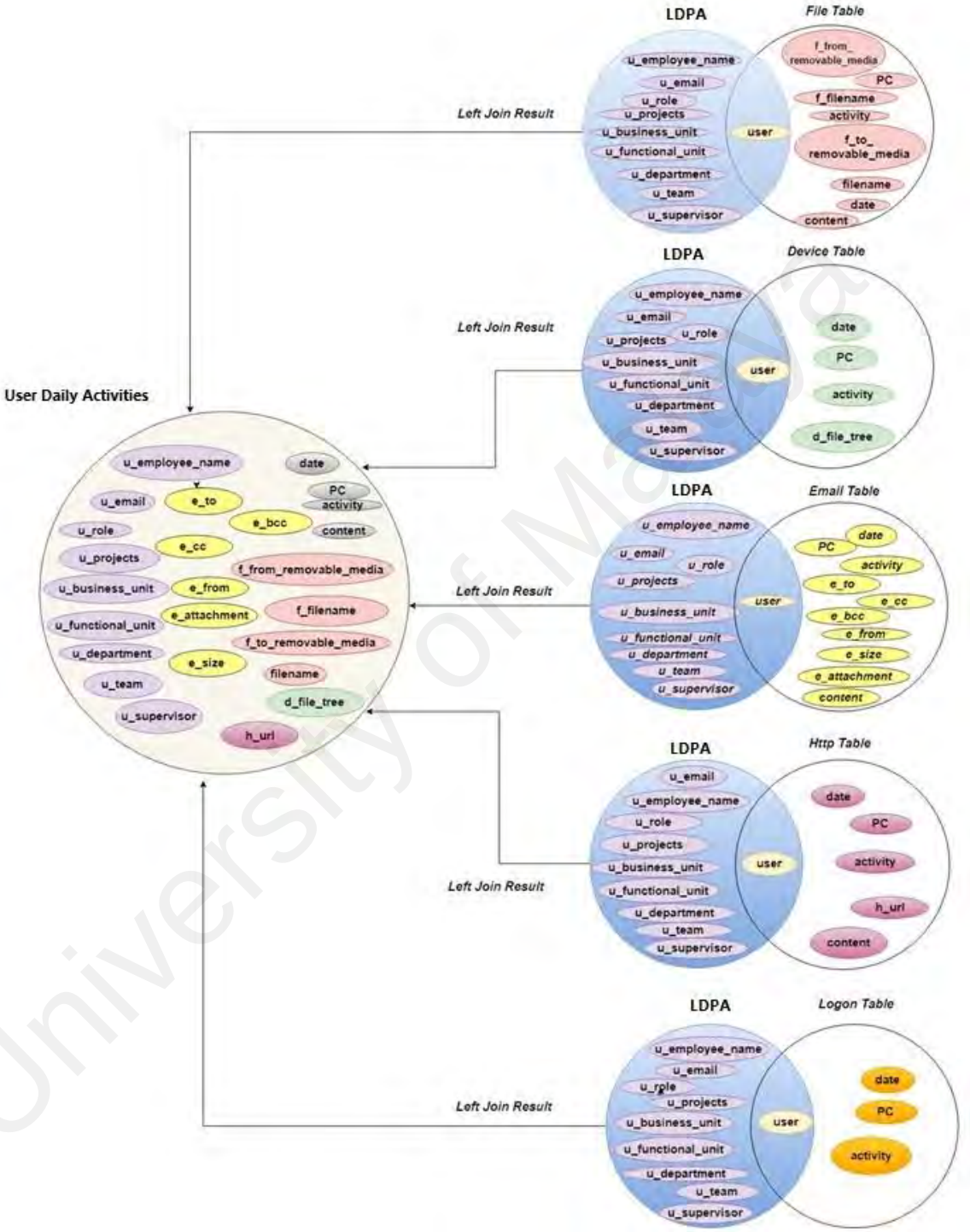

Figure 3.5: User daily activities model

Table 3.8: Engineered Features

Log on/Log off Features	Number of files copy from External device - tp
First log on/log Off	Number of files delete from External device
First log on/log Off time period (tp)	Number of files delete from External device - tp
Last log on/log Off	Number of files open from External device
Last log on/log Off - tp	Number of files open from External device - tp
Number of log on/log Off	Hourly usage of External device - tp
Hourly computer usage - tp	Daily usage of External device
Total computer usage	**Email Usage Features**
File Operation Features	Number of Email Send/View
Number of File copy/delete/open/write	Number of Email Send/View - tp
Number of File copy/delete/open/write - tp	Number of files attached
First File copy/delete/open/write	Size of file attached
First File copy/delete/open/write - tp	First Email Send/View
Last File copy/delete/open/write	First Email Send/View - tp
Last File copy/delete/open/write - tp	Last Email Send/View
External device usage Features	Last Email Send/View - tp
Number of connect/disconnect	**Web usage features**
Number of connect/disconnect - tp	Number of WWW download/upload/visit
First connect/disconnect	Number of WWW download/upload/visit - tp
First connect/disconnect - tp	First WWW download/upload/visit
Last connect/disconnect	First WWW download/upload/visit - tp
Last connect/disconnect - tp	Last WWW download/upload/visit
Number of files copy to External device	Last WWW download/upload/visit - tp
Number of files copy to External device - tp	Total Time spend on websites
Number of files copy from External device	Total Time spend on websites - tp

3.7 Feature Extraction

In total 218 features were generated but wanted to assess the amount of variance and redundant features in our feature engineered dataset. To do this, PCA was used as feature extraction method and to select number of principal components 95% of variance was applied. It means the algorithm select number of principal component based on 95% variance between features. Figure 3.6 illustrates by increasing number of principal component the variance increased until in 117 dimensions reaches to 95% of variation. From this point onward, by increasing the number of dimensions the variance changes slightly. It means the remaining 101 dimensions are not independent and discriminative, so they are redundant. This study assessed the effect of including or excluding the PCA in final output.

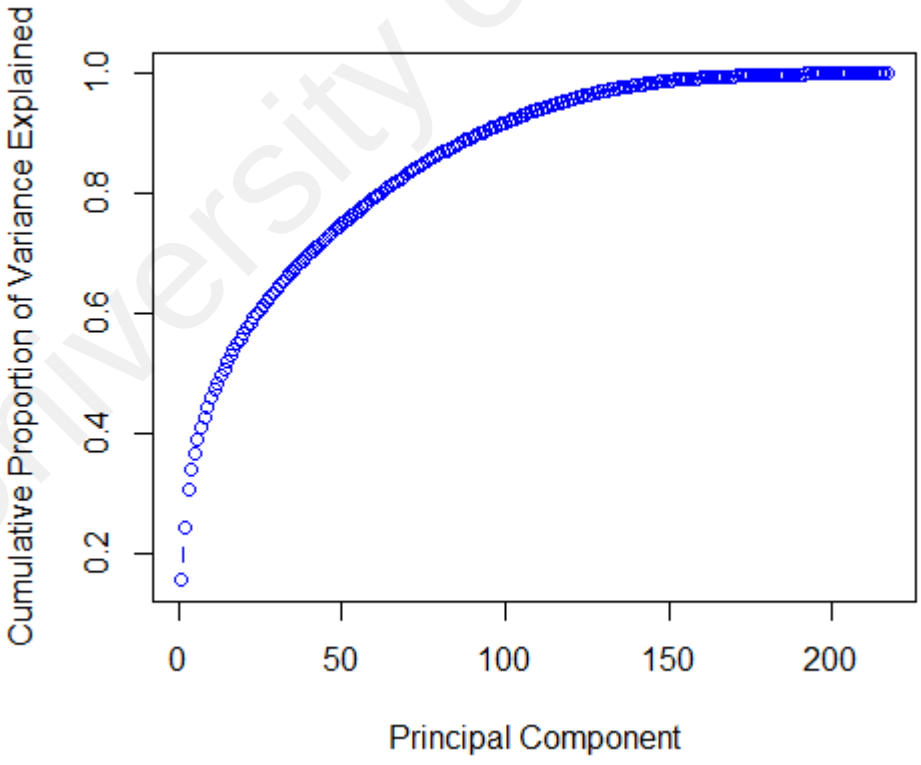

Figure 3.6: Optimum number of principal Component with 95% of variance

3.8 Machine Learning Algorithms

The CERT dataset has gone through data aggregation, feature engineering and feature extraction. The final dataset includes 117 features and daily activities for 200 employees over 516 days. Since this study use the unsupervised method, there is no training set. Therefore, the whole dataset chronologically split to development set and test set. The first 70% subset is used for model selection and hyperparameter tuning and remaining 30% is used to assess the performance. The final dataset is fed to the machine learning algorithms. The complete description of the machine learning algorithms utilized in this study was presented in chapter two. Each machine learning algorithms are applied on development set and the hyper parameters were tuning. After obtaining the optimum hyper parameters, this study applied them on test set and the result was compared by CERT dataset answer. The result was represented by ROC curve.

3.9 Hyperparameter Tuning

As mentioned in Chapter 2, iForest algorithm has two hyperparameter of number of iTrees and subsample size which can be tuned. In this study, to find the optimum number of iTrees we used confidence interval (CI). Confidence interval means a range of values that probably contains the population value. The formula to calculate CI can be seen in equation 3.1.

$$\bar{x} \pm z * \frac{\sigma}{\sqrt{n}} \tag{3.1}$$

Which $\bar{x}$ is mean of population value, σ is standard deviation, n is sample size and z is critical value. For 95% CI the z value is equal 1.96. In our problem, we run the program 40

times for different number of iTrees t= {20, 40, 60, 80, 100, 200, 300}, $\bar{x}$ and σ are mean and standard deviation of AUC for each t, respectively. Then by utilizing SPSS tools, calculate t based 95% CI for the mean of AUC.

3.10 Summary

This chapter discussed general overview and details of experiment conducted. It described different tools that have been employed in this study. The details of CERT dataset, feature engineering, feature extraction and machine learning algorithms have been presented to show how it works. Next chapter presents the empirical results of this study.

CHAPTER 4: Results and Discussion

4.1 Introduction

This chapter represents the experimental results and discusses the effect of hyperparameter tuning and feature extraction on improving the detection rate. This chapter is organized into seven sections. Section 4.2 explains the evaluation metrics used in the experiments. Section 4.3 presents tuning of hyperparameters. Section 4.4 presents the results of AUC, TPR, FPR and ROC curves obtained by applying PCA and without PCA. Section 4.5 presents the performance of iForest and One- class SVM. Sections 4.6 discusses this study results compared to other research works. Finally, Section 4.7 concludes the chapter.

4.2 Evaluation Measurements

In order to evaluate detection performance, it is essential to choose a proper performance metrics. In this study, the impact on performance of employing feature extraction method and anomaly detection approaches is assessed by the ROC analysis. A ROC curve is a plot of true positive rate (also called recall)) against the false positive rate (also called false alarm rate) for all decision thresholds. ROC curves are practical for studying the performance of an algorithm under different operating conditions, and for comparing the performance of different algorithms. These two parameters can be calculated as follows:

$$\text{True Positive Rate (TPR)} = \frac{\text{TP}}{\text{TP+FN}} \tag{4.1}$$

$$\text{False Positive Rate (FPR)} = \frac{\text{FP}}{\text{TN+FP}} \tag{4.2}$$

Figure 4.1 is a sample of a ROC curve, where the x-axis illustrates false positive percentages while y-axis illustrates true positive percentages. By improving the detection performance

curve lie towards the top left corner of the graph. The line y = x represents to a classifier that randomly assigns one of the two classes to each data sample with equal probability.

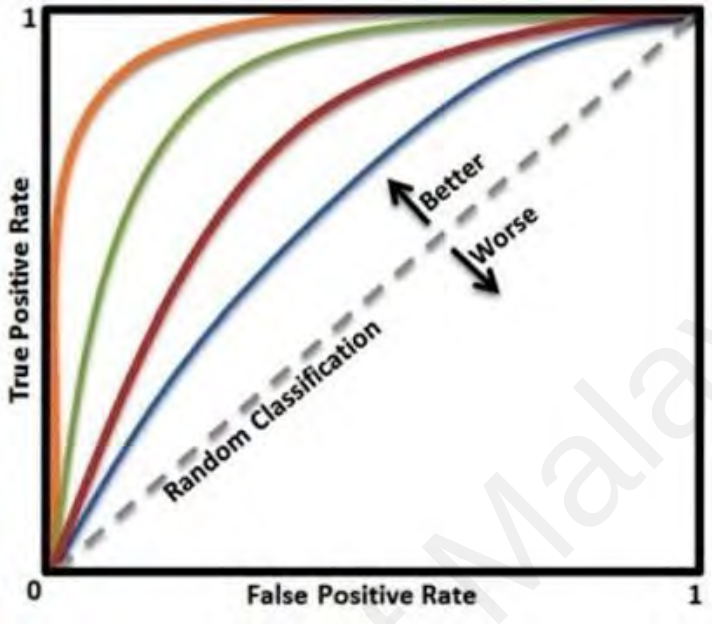

Figure 4.1: Example of ROC curve

Table 4.1 describes meaning of each label that use for computing TPR and FPR. Also, we calculate the area under the ROC curve (AUC) to compare different results.

Table 4.1: Label for computing TPR and FPR

Label	Description
TP-true positive	A data sample representing an attack is correctly classified as an attack.
FP-false positives	A data sample representing normal point is incorrectly classified as an attack.
TN-true negatives	A data sample representing normal points is correctly classified as normal.
FN-false negatives	A data sample representing an attack is incorrectly classified as normal.

An AUC value of 1 means a perfect result while a 0.5 value is a worthless result. The AUC point system is as follows: 1.00 - 0.90= excellent, 0.90 - 0.80 = good ,0.80 - 0.70 = fair ,0.70 - 0.60 = poor, and 0.60 - 0.50 = fail (Fawcett, 2006).

4.3 Hyperparameter Tuning

This study designs different experiments to evaluate proposed model's performance. In order to achieve reliable result, needs to find the optimum parameters for iForest algorithm. According Liu.et al. (Liu et al., 2012) study, the number of iTrees t=100 and subsample size ψ =265 are optimal for any type of data. Since used data in this research are completely different from those used in original paper, we expect different optimal parameters. As the first experiment, the AUC was measured for different values of t and ψ. Figure 4.2 reveals by increasing the number of subsamples, detection accuracy raises significantly. In proposed model, empirical evidence shows the AUC converges at ψ =1500 which is different from original paper. We hypothesize the number of data points and dimensions determine the number of subsamples.

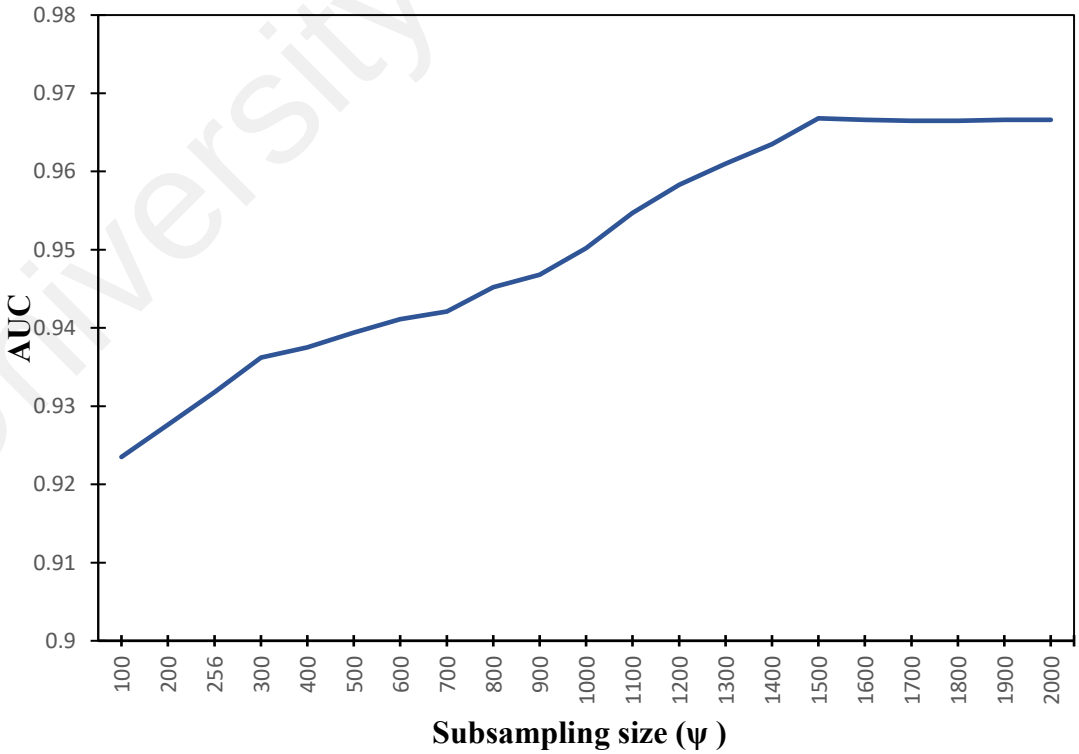

Figure 4.2: Relation between AUC and subsample size ψ

This study performs the second experiment to find the optimal number of iTrees which gives us reliable result. This study runs the program 40 times for different number of t= {20,40,60,80,100,200,300} and calculate t based 95% confidence interval (CI) for the mean of AUC by using SPSS tool. The result in table 4.2 shows by increasing the number of iTrees, the width of the confidence interval decreases. In the other word, the probability that CI contains the AUC value is 95%. By comparing the result for different t, it can be seen t=100 can gives us enough reliability of the later outcomes. Therefore, there is no need to select larger t which result in higher computational time as well as memory requirement.

Table 4.2: Width of confidence interval

t	Mean of AUC	95% Confidence Interval		Width of CI
		Lower Limit	Upper Limit	
20	0.86516000	0.8643575	0.8659625	0.001605
40	0.90525000	0.9045296	0.9059704	0.001441
60	0.92569000	0.9250280	0.9263520	0.001324
80	0.93409000	0.9337005	0.9344795	0.000779
100	0.96567250	0.9653906	0.9659544	0.000563
200	0.96572750	0.9654881	0.9659669	0.000279
300	0.96579500	0.9655662	0.9660238	0.000210

As third experiment, the computational time was measured for different values of ψ. It can be seen in figure 4.3, by increasing the subsample size the computational time grows slightly, for instance in ψ=300 and ψ=1500 the computational time is 13.91 and 33.73 seconds, respectively. Although the subsample size increase by 5 times but the computational

time increase by less than 2.5 times. In the other word, the accuracy of models can be increased for relatively small cost of time.

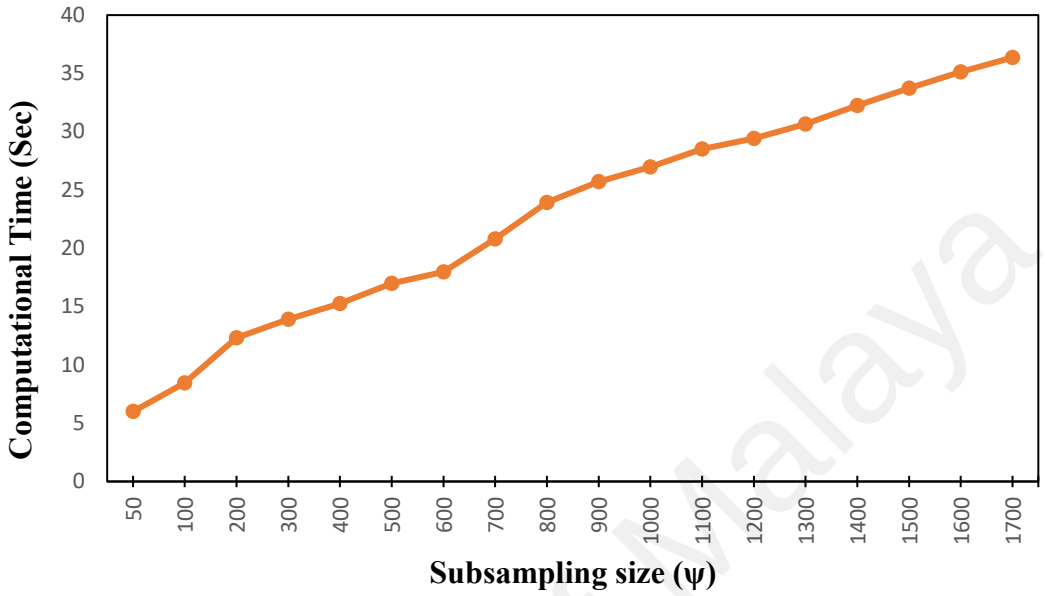

Figure 4.3: Relation between Computational Time and subsample size ψ

4.4 Effect of Feature Extraction Technique

Since we have the optimum number of itrees t=100 for iForest, wants to evaluate the effect of including or excluding the PCA in output. In this experiment, we run iForest and One-Class SVM algorithms with PCA and without PCA. This study plots ROC and calculates AUC for different conditions and compare the results. From figure 4.4 and figure 4.5, it is visible that by applying feature extraction method PCA, higher AUC in both One-Class SVM and iForest was achieved. It shows some of 218 engineered features are highly correlated therefore extracting these features from dataset improve the prediction accuracy about 10% and 17% in On-Class SVM and iForest, t=100, $\psi = 256$, respectively. Though this influence

does not impact all the subsample size in iForest uniformly, subsample size 1000, 1500 and 2000 record 7%,6.9% and 3.62% improvement in prediction accuracy, respectively.

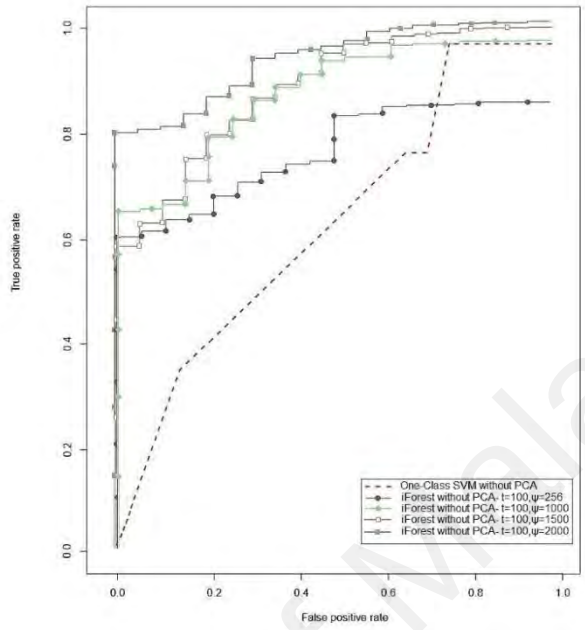

Figure 4.4: ROC curve without applying PCA

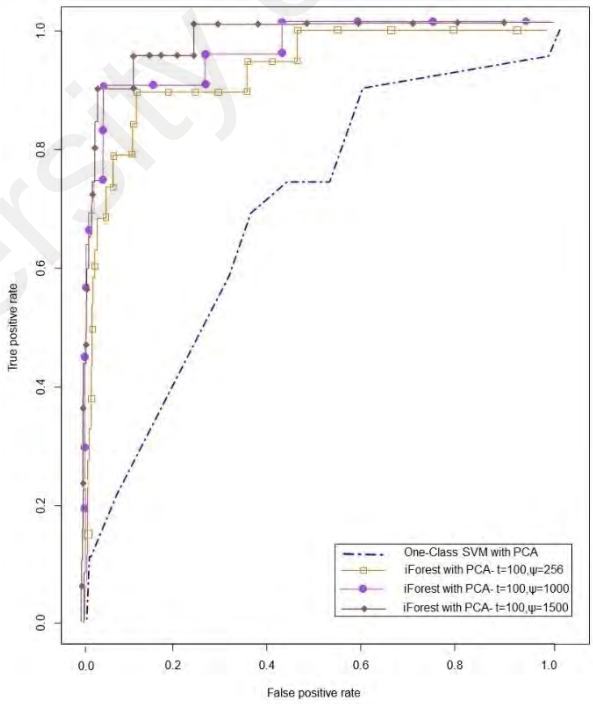

Figure 4.5: ROC curve with applying PCA

Moreover Figure 4.6 illustrates iForest algorithm without applying PCA performs better in presence of noise and redundant features compare to One-Class SVM. The iForest algorithm by utilizing multiple sub-samples reduce the effects of swamping. A small size subsample constructs a better performing iTree rather than whole data set. Subsamples have fewer normal points which interfering with anomalies; therefore, anomaly points can be isolated easier which result in higher prediction accuracy. Also, the AUC for iForest without PCA converges at the $\psi = 2000$, as mentioned before most likely there is relation between the feature dimensions and subsamples size, but further experiments need to be done.

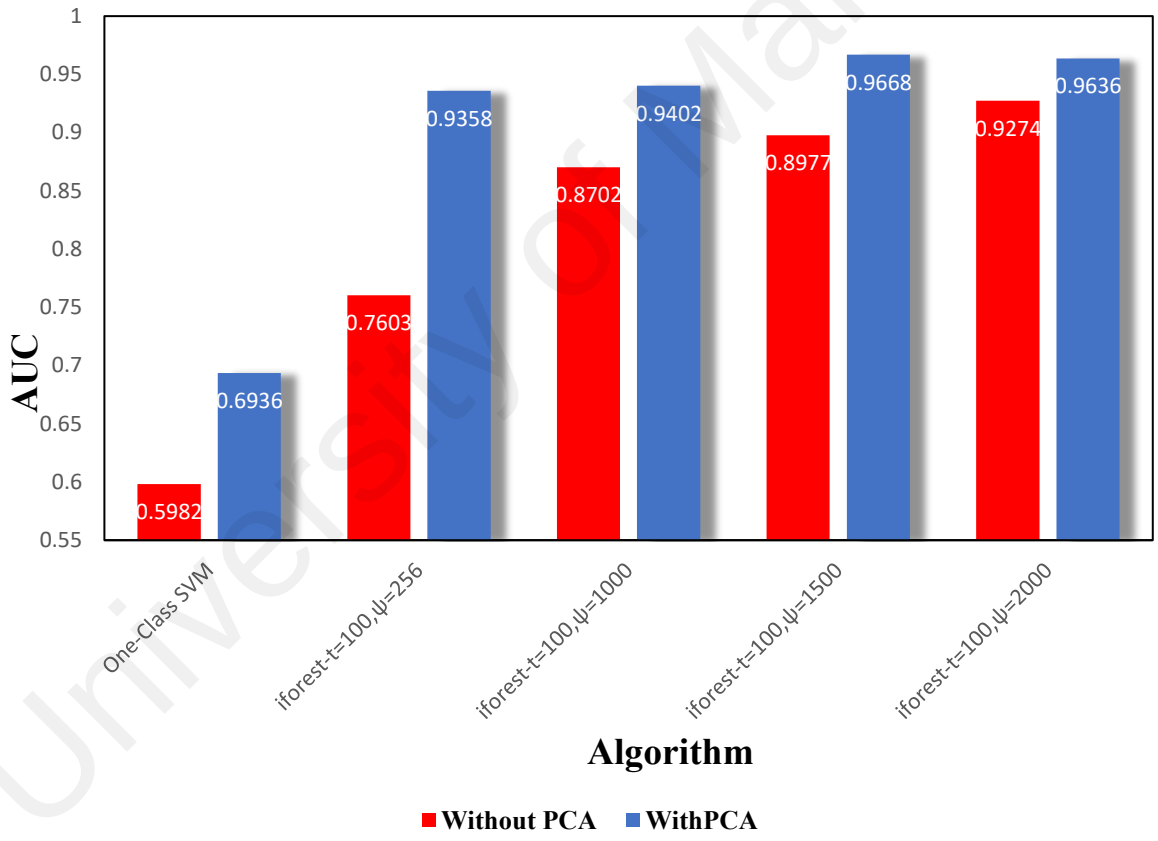

Figure 4.6: AUC value based on algorithm

In addition, this study calculates TPR which is the probability of correctly detecting an instance as an anomaly and FPR which is defined as falsely detection of normal points as

anomaly one. The TPR value as close as one hundred percent and the FPR as close as zero is ideal result. Figure 4.7 shows the empirical result for TPR. It is obvious form the graph that iForest has higher TPR compare to One Class SVM algorithm. Also, by increasing the sample size in iForest the TPR increase significantly until it reaches to its highest value of 93.2% in $\psi = 1500$ and 82.6% in $\psi = 2000$ with applying PCA and without PCA, respectively. The graph illustrates removing the irrelevant and redundant features by using PCA effectively increase TPR from 55.4% to 68.1% in One Class SVM and from 69.9% to 93.2% in iForest t=100, $\psi = 1500$. It is evident that the TPR result improves notably by employing the PCA feature reduction method and iForest is more powerful than One Class SVM to detect anomaly points.

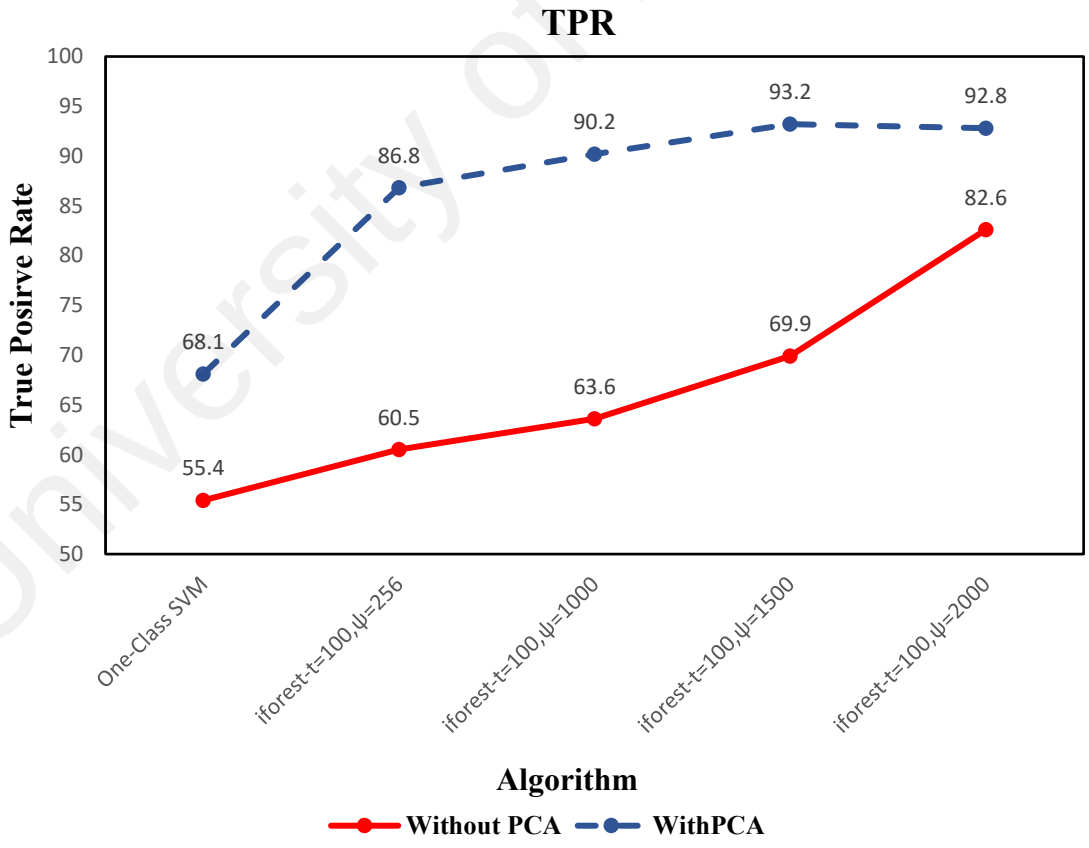

Figure 4.7 TPR value based on algorithm

Figure 4.8 shows the empirical result for FPR. The One Class SVM algorithm has higher rate of false positive rather than iForest. Moreover, by increasing the sample size in iForest the FPR decrease significantly until it reaches to its lowest value of 2.8% in $\psi = 1500$ and 9.3% in $\psi = 2000$ with applying PCA and without PCA, respectively. The figure reveals removing the irrelevant and redundant features by using PCA, FPR decrease dramatically from 26% to 19% in One Class SVM and from 10.7% to 2.8% in iForest t=100, $\psi = 1500$. It is observable that the FPR result improves remarkably by employing the PCA feature reduction method and iForest can handle swamping effect much better than One Class SVM.

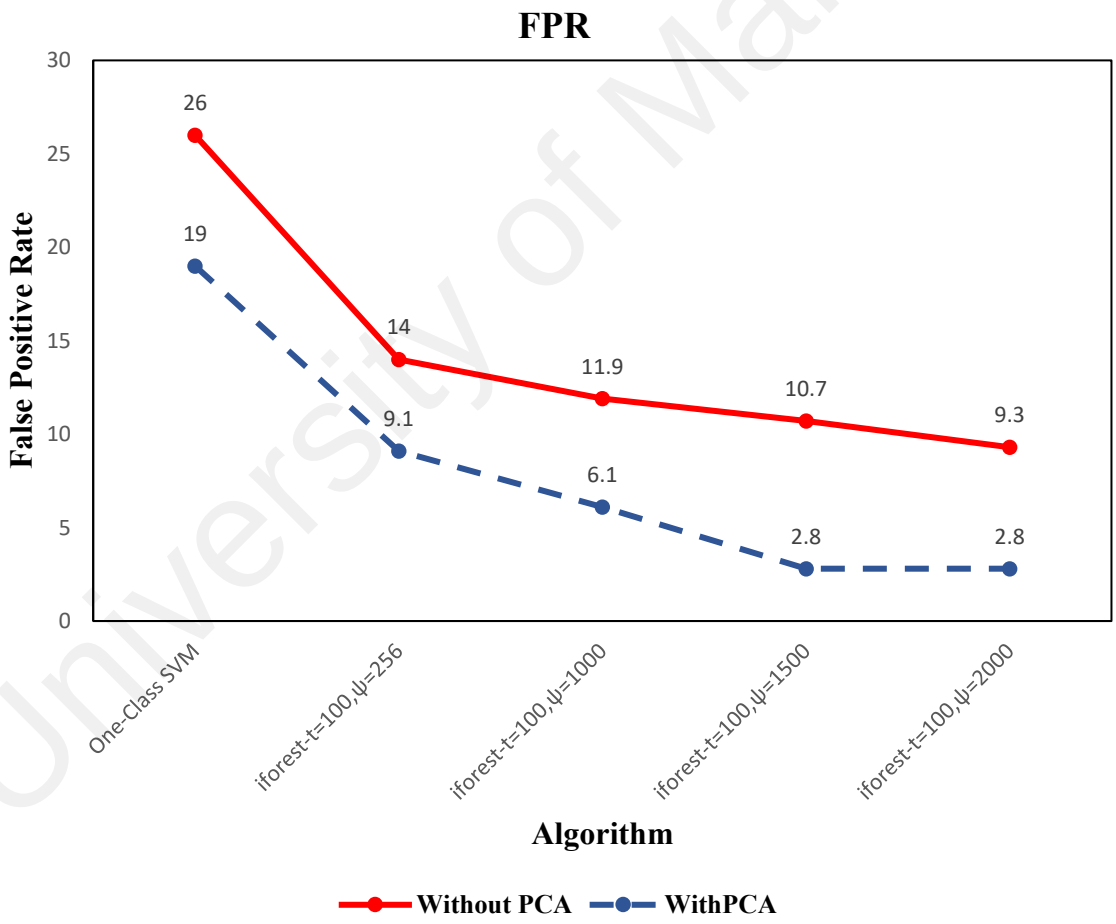

Figure 4.8 FPR value based on algorithm

4.5 Performance

As last experiment, this study compares the computational complexity and stability of One-Class SVM and iForest. In order to make our algorithms more objective, both algorithms are executed 20 times**.** Figure 4.9 shows iForest algorithm has less fluctuation degree and more stable compare to One-Class SVM. The standard deviation of iForest is σ=0.001 where as standard deviation is σ= 0.016 for One-Class SVM.

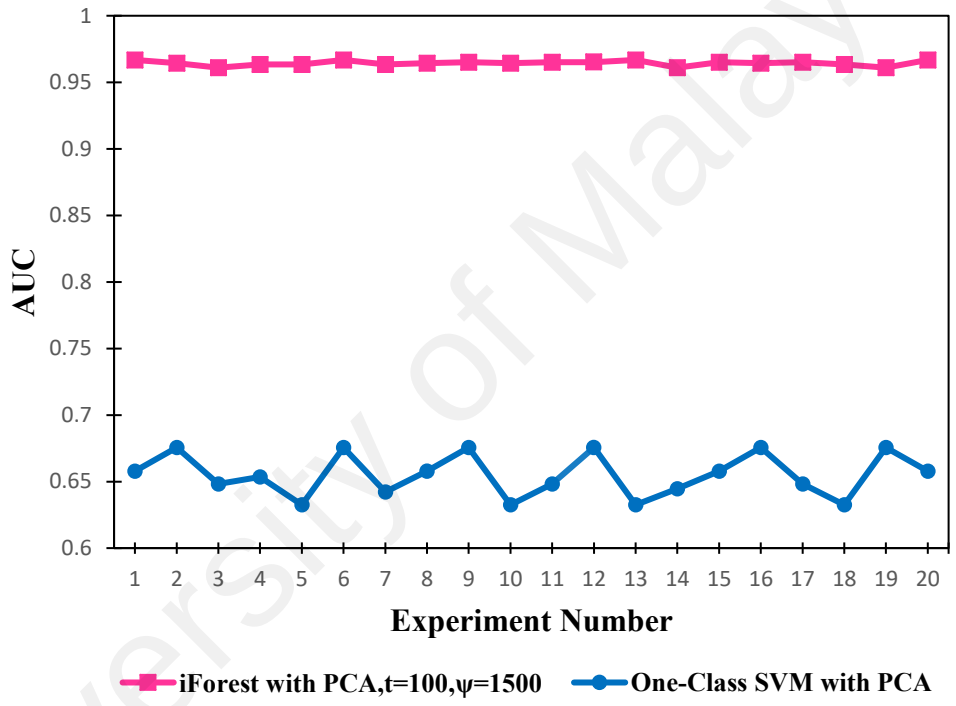

Figure 4.9: Stability test for anomaly detection

The graphs in Figure 4.10 illustrates the execution time of iForest is significantly below the One-Class SVM because iForest does not use distance or density values to detect anomalous points. Therefore, it has lower time complexity.

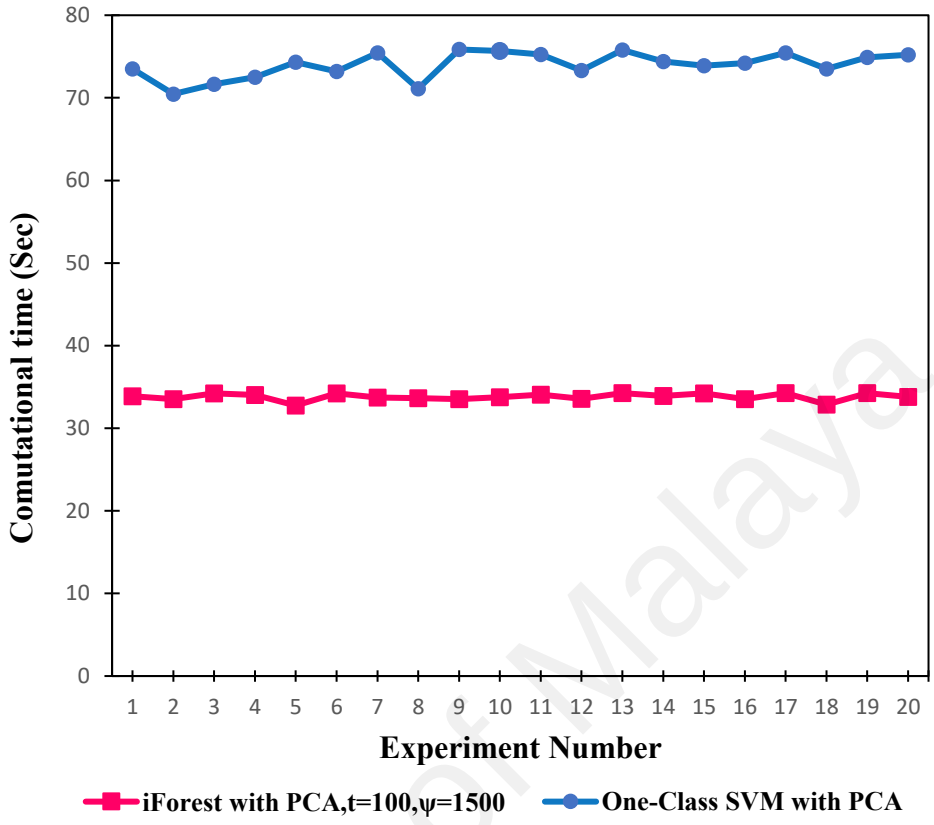

Figure 4.10: Computational Time for anomaly detection

4.6 Results Comparisons to Other Studies

One of the main goals of this study is to evaluate anomaly detection method. Thus, conducting a comparison with other studies is common practice to exhibit the enhancements over related works. As aforementioned, a study conducted by Gavai et al to detect anomalies in system log files over five different domains such as file, logon/logoff, http, device and email. They applied feature engineering to create new features and used iForest as unsupervised ML method but did not use any feature reduction method neither parameter tuning. The highest detection rate attained in their research was 76% which is fair based on AUC point system. On the other hand, by applying hyperparameter tuning and PCA feature

extraction method proposed method achieved 96.6% AUC which is significantly higher than their result.

It is evident that the results of this study are better than the similar work done by Gavai et al. (Gavai et al., 2015).

Tuor et al. (Tuor et al., 2017) recently published a study in which they worked on the same dataset, CERT and applied RNN and DNN as ML algorithms and iForest as a base line. They employed feature engineering to create new features and did not use any feature reduction method. They used default hyperparameters for iForest and performed the experiment only on weekdays and excluded the weekends and holidays. The highest true positive rate (TPR) result was 90% for RNN algorithm. DNN and iForest showed less than 90% TPR to detect insider threat. In contrast, this study includes all the weekends and holidays since there is a great chance attack is happening during these days. This proposed model by applying feature extraction and hyperparameter tuning, achieved 93.2% TPR. Surpassing the results obtained by Tuor et al. Thus, it is evident that this work has achieved better results than other related works by applying feature extraction method and hyperparameter tuning.

4.7 Effect of insider threat

Insider threat are more damaging compare to external ones due to their access to highly confidential information and their knowledge of the organizational systems. It can be seen from this study how malicious employees use their knowledge of the organizational systems to download company's confidential information in their emails or external devices. In other case, they use management email to send false emails. Based on ISACA and RSA Conference Survey 2016, 20% of companies are dealing with insider damage and theft of intellectual property at least quarterly. Moreover, rapid advancement of Artificial Intelligent (AI) will

lead to increase of the cybersecurity/information security risk in short term and long term. Therefore, it is crucial for companies to realize threats which influence their assets and the areas which each threat could affect.

4.8 Summary

This chapter discussed the results of the experiment and their interpretation. Different number of performance measurements and ROC curves were presented to ease the understanding of the results. The results were compared to two research works done by other researchers to show an enhancement in results. The next chapter concludes this work which includes achievements, limitations and future works, and contributions of this work.

Chapter 5: Conclusions

5.1 Introduction

This chapter sums up and reviews the current study by outlining the achievements of this work. It highlights the most important findings as well as the limitations. Moreover, it discusses the contributions made throughout this work. In addition, it makes suggestion for future research as well as the possible expansion of the current study.

5.2 Achievements

This study is aimed to using machine learning method to detect anomalies in system log files. It also aims to identify the most suitable unsupervised ML algorithms and feature reduction method to use based on true positive rate (TPR) values. In this line, anomaly detection methods and feature reduction techniques were explored by reviewing the current works. The different topics were summarized in Chapter 2, under three sub section intrusion detection techniques, machine learning methods and feature reduction.

The research methodology employed for experiments was explained in Chapter 3. The experimental setup and configuration needed to implement Microsoft SQL Server, R Studio and SPSS was discussed. The CERT dataset aspects were described completely. The methodology was implemented in three phases: data aggregation and engineering, features extraction, and machine learning algorithms.

Analysis of the experimental results and discussion of each experiment was described in Chapter 4. A comparison of AUC, TPR and FPR for each unsupervised ML algorithms was discussed, followed by discussion on the influence of hyperparameter tuning and feature extraction. Furthermore, study considered the performance and stability of each unsupervised ML algorithms.

Chapter 4 shows the outcomes of the experiments. The significance of this research is the use of the feature extraction method, together with finding suitable hyperparameter for the state of the art iForest algorithm. The results show that employing feature extraction along with hyperparameter tuning have significantly improved the detection rate. The results obtained are summarized as follows:

It is believed that this study has attained compelling results in detecting anomalies in system log files by using the machine learning approach. The comparison to previous study, Gavai et al. (Gavai et al., 2015), which was done in the chapter 4, shows this study achieved highly acceptable result 96.6% AUC by using hyperparameter tuning in iForest and feature extraction. The second comparison was also done with a study conducted by Tuor et al. (Tuor et al., 2017) and the results of this study surpassed the result in the work done by Tuor.

1. The PCA result showed that only 117 of engineered features have 95% of variance and 101 remaining features are not independent and discriminative.
2. The experiment result illustrates by applying feature extraction the AUC increase about 10% and 17% in On-Class SVM and iForest, t=100, $\psi = 256$, respectively. Also, the results showed 16% increase in TPR and 5% decrease in FPR after applying feature extraction in iForest, t=100, $\psi = 256$. It is evident that applying feature extraction significantly enhanced the AUC, TPR and decrease FPR notably.
3. In the result, we noticed hyperparameter tuning along with feature extraction has a remarkable effect on the detection rate. The AUC and TPR reached to highest values of 96.6% and 93.2%, respectively and FPR shows lowest value 0f 2.8% in iForest, t=100, $\psi = 1500$.

5.3 Contribution

This study provides several contributions, which are as follows:

1. One of the contributions of this work is data aggregation that aggregate data based on each user daily activities. In this study we aggregated daily activities for 200 employees over 516 days.
2. Another contribution is converting the raw system log files to proper features that can be fed to ML algorithm by feature engineering. Thus, building a dataset from the raw data is a fundamental step in conducting this work.
3. Applying feature extraction method to extract the most independent and discriminative features amongst 218 engineered features. The presented results confirm the effectiveness of using this method in this thesis.
4. Tuning hyperparameters in iForest algorithm which showed significant improvement in outcome.

5.4 Limitations

During conducting this study, we faced several challenges in the methodology section, which are as follows:

1. Obtaining a dataset which represents complexity of real - world dataset was one of main concerns in this study. The CERT data set was found by extensive research which simulated organization's computer network, generated with sophisticated user models.
2. Data aggregation due to huge number of activities by using R studio was very difficult because of high memory consumption. Therefore, to overcome this problem Microsoft SQL Server was used.

3. Generating independent features that represents the user activities was another challenge that was solved by comprehensive study of previous works.

5.5 Future works

The following are suggestions for future work outside the scope of this study, that can be undertaken:

1. The presented study was done only on 200 employees. Adding more employees increase the complexity of the system. Therefore, developing a detection method by analyzing more employees can be suggested as a future work.
2. Using feature selection methods and explore effect of feature selection on outputs can be considered as another future work.
3. Design a system tools which measure anomaly score for each user based on his/her previous activities and send an alert if the score is higher than threshold be another future work.

5.6 Summary

This chapter concludes the thesis with discussions of the achievements, contribution, limitations, and future extensions of the project. This project has successfully produced a model that can detect anomalies in system log files via unsupervised learning algorithms. Although we have improved the prediction accuracy significantly, this obtained outcomes should not put a stop to our research work as the attackers use more complex and cunning methods.

References

A Diaz-Gomez, P., Vallecarcamo, G., & Jones, D. (2017). *Internal Vs. External Penetrations: A Computer Security Dilemma*.

Abbors, F., Truscan, D., & Ahmad, T. (2015). Mining Web Server Logs for Creating Workload Models *Software Technologies: 9th International Joint Conference, ICSOFT 2014, Vienna, Austria,* (pp. 131-150). Cham: Springer International Publishing.

Agrafiotis, I., Nurse, J. R. C., Buckley, O., Legg, P., Creese, S., & Goldsmith, M. (2015). Identifying attack patterns for insider threat detection. *Computer Fraud & Security,* 2015(7), pp. 9-17.

Agrawal, S., & Agrawal, J. (2015). Survey on anomaly detection using data mining techniques. *Procedia Computer Science,* 60, pp. 708-713.

Amer, M., Goldstein, M., & Abdennadher, S. (2013). Enhancing one-class support vector machines for unsupervised anomaly detection. *Proceedings of the Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description*, pp. 8-15.

Anwar, S., Mohamad Zain, J., Zolkipli, M. F., Inayat, Z., Khan, S., Anthony, B., & Chang, V. (2017). From intrusion detection to an intrusion response system: fundamentals, requirements, and future directions. *Algorithms,* 10(2), pp. 39.

Bace, R., & Mell, P. (2001). NIST special publication on intrusion detection systems: BOOZ-ALLEN AND HAMILTON INC MCLEAN VA.

Bauer, S., & Bernroider, E. W. N. (2017). From Information Security Awareness to Reasoned Compliant Action: Analyzing Information Security Policy Compliance in a Large Banking Organization. *SIGMIS Database,* 48(3), pp. 44-68.

Bellman, R., & Corporation, R. (1957). *Dynamic Programming*: Princeton University Press.

Bhattacharjee, S. D., Yuan, J., Jiaqi, Z., & Tan, Y.-P. (2017). *Context-aware graph-based analysis for detecting anomalous activities*. Paper presented at the Multimedia and Expo (ICME), 2017 IEEE International Conference on.

Bohara, A., Thakore, U., & Sanders, W. H. (2016). Intrusion detection in enterprise systems by combining and clustering diverse monitor data. *Proceedings of the Proceedings of the Symposium and Bootcamp on the Science of Security*, pp. 7-16.

Breier, J., & Branišová, J. (2015). Anomaly Detection from Log Files Using Data Mining Techniques. In K. J. Kim (Ed.), *Information Science and Applications* (pp. 449-457). Berlin, Heidelberg: Springer Berlin Heidelberg.

Breier, J., & Branišová, J. (2017). A Dynamic Rule Creation Based Anomaly Detection Method for Identifying Security Breaches in Log Records. *Wireless Personal Communications,* 94(3), pp. 497-511.

Breunig, M. M., Kriegel, H.-P., Ng, R. T., & Sander, J. (2000). LOF: identifying density-based local outliers. *Proceedings of the ACM sigmod record*, pp. 93-104.

Buczak, A. L., & Guven, E. (2016). A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. *IEEE Communications Surveys & Tutorials,* 18(2), pp. 1153-1176.

Byers, S., & Raftery, A. E. (1998). Nearest-neighbor clutter removal for estimating features in spatial point processes. *Journal of the American Statistical Association,* 93(442), pp. 577-584.

Camacho, J., Pérez-Villegas, A., García-Teodoro, P., & Maciá-Fernández, G. (2016). PCA-based multivariate statistical network monitoring for anomaly detection. *Computers & Security,* 59, pp. 118-137.

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Comput. Surv.,* 41(3), pp. 1-58.

Chandola, V., Banerjee, A., & Kumar, V. (2012). Anomaly detection for discrete sequences: A survey. *IEEE Transactions on Knowledge and Data Engineering,* 24(5), pp. 823-839.

Chiang, J.-T. (2008). The algorithm for multiple outliers detection against masking and swamping effects. *Int J Contemp Math Sciences,* 3, pp. 839-859.

Chuvakin, A., Schmidt, K., & Phillips, C. (2013). Chapter 2 - What is a Log? In A. Chuvakin, K. Schmidt & C. Phillips (Eds.), *Logging and Log Management* (pp. 29-49). Boston: Syngress.

Correa Bahnsen, A., Aouada, D., Stojanovic, A., & Ottersten, B. (2016). Feature engineering strategies for credit card fraud detection. *Expert Systems with Applications,* 51, pp. 134-142.

Cui, L., & Shi, Y. (2014). A Method based on One-class SVM for News Recommendation. *Procedia Computer Science,* 31, pp. 281-290.

Das, S., & Nene, M. J. (2017, 22-24 March 2017). A survey on types of machine learning techniques in intrusion prevention systems. *Proceedings of the 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, pp. 2296-2299.

Eldardiry, H., Bart, E., Liu, J., Hanley, J., Price, B., & Brdiczka, O. (2013). Multi-domain information fusion for insider threat detection *2013 IEEE Security and Privacy Workshops*. San Francisco: IEEE.

Erfani, S. M., Rajasegarar, S., Karunasekera, S., & Leckie, C. (2016). High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning. *Pattern Recognition,* 58, pp. 121-134.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters,* 27(8), pp. 861-874.

Gavai, Sricharan, K., Gunning, D., Rolleston, R., Hanley, J., & Singhal, M. (2015). *Detecting Insider Threat from Enterprise Social and Online Activity Data*. Paper presented at the Proceedings of the 7th ACM CCS International Workshop on Managing Insider Security Threats, Denver, Colorado, USA.

Gavai, G., Kumar, S., Dave, G., John, H., Mudita, S., & Rob, R. (2015). Supervised and Unsupervised methods to detect Insider Threat from Enterprise Social and Online Activity Data. *JoWUA,* 6(4), pp. 47-63.

Gentili, M., Hajian, S., & Castillo, C. (2017). *A Case Study of Anonymization of Medical Surveys*. Paper presented at the Proceedings of the 2017 International Conference on Digital Health, London, United Kingdom.

Gheyas, I. A., & Abdallah, A. E. (2016). Detection and prediction of insider threats to cyber security: a systematic literature review and meta-analysis. *Big Data Analytics,* 1(1), pp. 6.

Glasser, J., & Lindauer, B. (2013). Bridging the gap: a pragmatic approach to generating insider threat data *2013 IEEE Security and Privacy Workshops*. San Francisco: IEEE.

Gogoi, P., Borah, B., & Bhattacharyya, D. K. (2010). *Anomaly Detection Analysis of Intrusion Data Using Supervised & Unsupervised Approach* (Vol. 5).

Greitzer, F. L., Frincke, D. A., & Zabriskie, M. (2010). Social/ethical issues in predictive insider threat monitoring. *Information Assurance and Security Ethics in Complex Systems: Interdisciplinary Perspectives*, pp. 132-161.

Guyon, I., Gunn, S., Nikravesh, M., & Zadeh, L. A. (2006). *Feature Extraction: Foundations and Applications (Studies in Fuzziness and Soft Computing)*: Springer-Verlag New York, Inc.

Hira, Z. M., & Gillies, D. F. (2015). A review of feature selection and feature extraction methods applied on microarray data. *Advances in bioinformatics,* 2015.

Ikram, S. T., & Cherukuri, A. K. (2016). Improving accuracy of intrusion detection model using PCA and optimized SVM. *Journal of computing and information technology,* 24(2), pp. 133-148.

Jolliffe, I. (2011). Principal component analysis *International encyclopedia of statistical science* (pp. 1094-1096): Springer.

Jolliffe, I. T., & Cadima, J. (2016). Principal component analysis: a review and recent developments. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences,* 374(2065), pp. 20150202.

Jyothsna, V., Prasad, V. R., & Prasad, K. M. (2011). A review of anomaly based intrusion detection systems. *International Journal of Computer Applications,* 28(7), pp. 26-35.

Karev, D., McCubbin, C., & Vaulin, R. (2017). *Cyber Threat Hunting Through the Use of an Isolation Forest*. Paper presented at the Proceedings of the 18th International Conference on Computer Systems and Technologies, Ruse, Bulgaria.

Kent, K., & Souppaya, M. P. (2006). Guide to Computer Security Log Management: National Institute of Standards \& Technology.

Kevin Richards, Ryan LaSalle, & Devost, M. (2017). "Cost of cyber crime study": Ponemon Institute and Accenture.

Kinshumann, K., Glerum, K., Greenberg, S., Aul, G., Orgovan, V., Nichols, G., . . . Hunt, G. (2011). Debugging in the (very) large: ten years of implementation and experience. *Commun. ACM,* 54(7), pp. 111-116.

Knorr, E. M., Ng, R. T., & Tucakov, V. (2000). Distance-based outliers: algorithms and applications. *The VLDB Journal—The International Journal on Very Large Data Bases,* 8(3-4), pp. 237-253.

Kongsg, K. W., Nordbotten, N. A., Mancini, F., & Engelstad, P. E. (2017). *An Internal/Insider Threat Score for Data Loss Prevention and Detection*. Paper presented at the Proceedings of the 3rd ACM on International Workshop on Security And Privacy Analytics, Scottsdale, Arizona, USA.

Krishnan, S., & Athavale, Y. (2018). Trends in biomedical signal feature extraction. *Biomedical Signal Processing and Control,* 43, pp. 41-63.

Kruskal, J. B., & Wish, M. (1978). Multidimensional scaling. Number 07–011 in Sage University Paper series on quantitative applications in the social sciences: Sage Publications, Beverly Hills.

Lakhina, S., Joseph, S., & Verma, B. (2010). Feature reduction using principal component analysis for effective anomaly–based intrusion detection on NSL-KDD.

Legg, P. A., Buckley, O., Goldsmith, M., & Creese, S. (2015a). Automated insider threat detection system using user and role-based profile assessment. *IEEE Systems Journal,* 11(2), pp. 503-512.

Legg, P. A., Buckley, O., Goldsmith, M., & Creese, S. (2015b). *Caught in the act of an insider attack: Detection and assessment of insider threat*. Paper presented at the

Technologies for Homeland Security (HST), 2015 IEEE International Symposium on.

Li, H., Shang, W., Zou, Y., & E. Hassan, A. (2017). Towards just-in-time suggestions for log changes. *Empirical Software Engineering,* 22(4), pp. 1831-1865.

Lin, Q., Zhang, H., Lou, J.-G., Zhang, Y., & Chen, X. (2016). *Log clustering based problem identification for online service systems*. Paper presented at the Proceedings of the 38th International Conference on Software Engineering Companion, Austin, Texas.

Ling Ko, L., Divakaran, D. M., Siang Liau, Y., & L. L. Thing, V. (2016). *Insider Threat Detection and its Future Directions* (Vol. 12).

Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation forest. *Proceedings of the Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pp. 413-422.

Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2012). Isolation-based anomaly detection. *ACM Transactions on Knowledge Discovery from Data (TKDD),* 6(1), pp. 3.

Lu, W., Cheng, Y., Xiao, C., Chang, S., Huang, S., Liang, B., & Huang, T. (2017). Unsupervised Sequential Outlier Detection With Deep Architectures. *IEEE Transactions on Image Processing,* 26(9), pp. 4321-4330.

Makani, R., & Reddy, B. V. R. (2018). Taxonomy of Machine Leaning Based Anomaly Detection and its suitability. *Procedia Computer Science,* 132, pp. 1842-1849.

McGough, A. S., Wall, D., Brennan, J., Theodoropoulos, G., Ruck-Keene, E., Arief, B., . . . Alwis, S. (2015). *Detecting Insider Threats Using Ben-ware: Beneficial Intelligent Software for Identifying Anomalous Human Behaviour*. Paper presented at the Proceedings of the 7th ACM CCS International Workshop on Managing Insider Security Threats, Denver, Colorado, USA.

Moradpoor, N., Clavie, B., & Buchanan, B. (2017). Employing machine learning techniques for detection and classification of phishing emails. *Proceedings of the Computing Conference, 2017*, pp. 149-156.

Mukkamala, S., Sung, A., & Abraham, A. (2005). Cyber security challenges: Designing efficient intrusion detection systems and antivirus tools.

Nagaraj, K., Killian, C., & Neville, J. (2012). Structured comparative analysis of systems logs to diagnose performance problems. *Proceedings of the Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*, pp. 26-26.

Narudin, F. A., Feizollah, A., Anuar, N. B., & Gani, A. (2016). Evaluation of machine learning classifiers for mobile malware detection. *Soft Computing,* 20(1), pp. 343-357.

Nexus, C. (2015). State of Cybersecurity:Implications for 2015. *"An ISACA and RSA Conference Survey"*.

Nexus, C. (2016). State of Cybersecurity:Implications for 2016. *"An ISACA and RSA Conference Survey"*.

Parmar, J. D., & Patel, J. T. (2017). Anomaly Detection in Data Mining: A Review. *International Journal,* 7(4).

Pearson, K. (1901). LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science,* 2(11), pp. 559-572.

Peng, J., Choo, K.-K. R., & Ashman, H. (2016). User profiling in intrusion detection: A review. *Journal of Network and Computer Applications,* 72, pp. 14-27.

Pölsterl, S., Conjeti, S., Navab, N., & Katouzian, A. (2016). Survival analysis for high-dimensional, heterogeneous medical data: Exploring feature extraction as an alternative to feature selection. *Artificial Intelligence in Medicine,* 72, pp. 1-11.

Preiss.BR (2000). Data structures and algorithms with object-oriented design patterns in Java.

Priya C V, & Angel Viji , K. S. (2002). A Brief View of Anomaly Detection Techniques for Intrusion Detection Systems *International Journal of Computer Technology & Applications,* 8(2), pp. 270-278.

R. Prasad. (2009). *Insider Threat to Organizations in the Digital Era and Combat Strategies*. Paper presented at the Indo-US conference and workshop on "Cyber Security, Cyber Crime and Cyber Forensics, Kochi, India.

Rashid, T., Agrafiotis, I., & Nurse, J. R. C. (2016). *A New Take on Detecting Insider Threats: Exploring the Use of Hidden Markov Models*. Paper presented at the Proceedings of the 8th ACM CCS International Workshop on Managing Insider Security Threats, Vienna, Austria.

Rubinstein, B. I., Nelson, B., Huang, L., Joseph, A. D., Lau, S.-h., Taft, N., & Tygar, D. (2008). Compromising PCA-based anomaly detectors for network-wide traffic.

Samuel, A. L. (1988). Some Studies in Machine Learning Using the Game of Checkers. I. In D. N. L. Levy (Ed.), *Computer Games I* (pp. 335-365). New York, NY: Springer New York.

Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., & Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural computation,* 13(7), pp. 1443-1471.

Shi, Y. (2018, 8-10 Jan. 2018). An attempt to analyze data distribution for abnormal behaviors. *Proceedings of the 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 275-280.

Smith, S. S. (2016). INTERNET CRIME REPORT "FBI's Internet Crime Complaint Center ".

Sun, Y., Xu, H., Bertino, E., & Sun, C. (2016). A Data-Driven Evaluation for Insider Threats. *Data Science and Engineering,* 1(2), pp. 73-85.

Tang, B., & He, H. (2017). A local density-based approach for outlier detection. *Neurocomputing,* 241, pp. 171-180.

Tang, J., Chen, Z., Fu, A. W.-C., & Cheung, D. W. (2002). Enhancing effectiveness of outlier detections for low density patterns. *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 535-548.

Thaseen, I. S., & Kumar, C. A. (2016). Intrusion Detection Model using fusion of chi-square feature selection and multi class SVM. *Journal of King Saud University-Computer and Information Sciences*.

Thearling, K. (2017). An introduction to data mining.

Tuor, A., Kaplan, S., Hutchinson, B., Nichols, N., & Robinson, S. (2017). Deep Learning for Unsupervised Insider Threat Detection in Structured Cybersecurity Data Streams.

V. Vapnik, & Lerner, A. (1963). Pattern Recognition using Generalized Portrait Method. *Automation and Remote Control,* 24, pp. 774-780.

Vaarandi, R., Kont, M., & Pihelgas, M. (2016). Event log analysis with the LogCluster tool. *Proceedings of Military Communications Conference MILCOM 2016-2016 IEEE*, pp. 982-987.

Wang, S., Tang, J., & Liu, H. (2016). Feature selection. *Encyclopedia of Machine Learning and Data Mining*, pp. 1-9.

Wickramasinghe, R. I. P. (2017). Attribute Noise, Classification Technique, and Classification Accuracy. In I. Palomares Carrascosa, H. K. Kalutarage & Y. Huang (Eds.), *Data Analytics and Decision Support for Cybersecurity: Trends,*

Methodologies and Applications (pp. 201-220). Cham: Springer International Publishing.

Wu, M., & Jermaine, C. (2006). Outlier detection by sampling with accuracy guarantees. *Proceedings of the Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 767-772.

Xu, D., Wang, Y., Meng, Y., & Zhang, Z. (2017). An Improved Data Anomaly Detection Method Based on Isolation Forest. *Proceedings of the Computational Intelligence and Design (ISCID), 2017 10th International Symposium on*, pp. 287-291.

Publication

Zamanian, Z., Feizollah, A., Anuar, N., & Mat Kiah, M. L. (2017). Anomaly Detection in Policy Authorization Activity Logs. *International Journal of Engineering Research in Computer Science and Engineering*, Malaysia, pp. 283-288

Zamanian, Z., Feizollah, A., Anuar, N. B., Kiah, L. B. M., Srikanth, K., & Kumar, S. (2019). User Profiling in Anomaly Detection of Authorization Logs. *Proceedings of the Computational Science and Technology*, Singapore, pp. 59-65.