An Intelligent Insider Threat Detection using ML Techniques

By

Muhammad Faisal Nawaz Janjua

(Registration Number: 00000281274)

Thesis Supervisor: Dr Asif Masood

Department of Information Security

Military College of Signals (MCS)

National University of Sciences and Technology (NUST)

Islamabad, Pakistan

November 2021

An Intelligent Insider Threat Detection using ML Techniques

By

Muhammad Faisal Nawaz Janjua

(Registration Number: 00000281274)

Thesis submitted to the National University of Sciences and Technology, Islamabad, in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in

Information Security

Thesis Supervisor: Dr Asif Masood

Department of Information Security

Military College of Signals (MCS)

National University of Sciences and Technology (NUST)

Islamabad, Pakistan

November 2021

Thesis Acceptance Certificate

Certified that PhD Thesis written by **Muhammad Faisal Nawaz Janjua**, (Registration No. 0000281274), of information Security, Military College of Signals has been vetted by undersigned, found complete in all respects as per NUST Statutes / Regulations / PhD Policy, is free of plagiarism, errors, and mistakes and is accepted as partial fulfillment for award of PhD degree. It is further certified that necessary amendments as pointed out by GEC members and foreign/local evaluators of the scholar have also been incorporated in the said thesis.

Signature: _____

Name of Supervisor: Dr Asif Masood

Date: _____

Signature (HOD): _____

Date: _____

Signature (Dean/Principal): _____

Date: _____

Declaration

I, Muhammad Faisal Nawaz Janjua hereby state that my PhD thesis titled

"An Intelligent Insider Threat Detection using Machine Learning Techniques"

is my own work and has not been submitted previously by me for taking any degree from this university

“National University of Sciences and Technology”

Or anywhere else in the country/world.

If my statement is found to be false even after my Graduate the university has the right to withdraw my PhD degree.

Muhammad Faisal Nawaz Janjua,
Reg No 00000281274

Plagiarism Undertaking

I solemnly declare that research work presented in thesis titled **"An Intelligent Insider Threat Detection using Machine Learning Techniques"** is solely my research work with no significant contribution from any other person. Small contribution/help wherever taken has been duly acknowledged and that completely written by me.

I understand the zero tolerance policy of the HEC and National University of Sciences and Technology (NUST), Islamabad towards plagiarism. Therefore, I as an Author of the above titled thesis declare that no portion of my thesis has been plagiarized and any material used as reference is properly referred/ cited.

I undertake that if I am found guilty of any plagiarism in the above titled thesis even after award of PhD degree, the University reserves the rights to withdraw/revoke my PhD degree and that HEC and the University has the right to publish my name on the HEC/University Website on which names of students are placed who submitted plagiarized thesis.

Signature: _____

Name: Muhammad Faisal Nawaz Janjua

List of Publications

It is certified that the following publications have been produced as a result of this research:

1. Janjua, F., Masood, A., Abbas, H., Rashid, I. (2020). Handling Insider Threat Through Supervised Machine Learning Techniques.11th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2020), Procedia Computer Science, 177, 64-71. (Published)

2. Janjua, F., Masood, A., Abbas, H., Rashid, I., Murtaza, Z. Textual Analysis of Traitor-Based Dataset through Semi Supervised Machine Learning. Journal of Future Generation Computer Systems. (IF: 7.187) (Published)

3. Janjua, F., Masood, A., Abbas, H., Rashid, I. Dynamic Weighted-Voting Ensemble Framework for Insider Threat Detection. International Journal of Information Security. (IF: 2.42) (Under Review)

4. Janjua, F., Masood, A., Abbas, H., Rashid, I. S2M: Supervised Stacked Model for Handling Class Imbalance in Insider Threat Dataset. Journal of PeerJ Computer Science. (IF: 3.09) (Accepted)

Abstract

Organization's data confidentiality with strong cryptographic primitives is primarily not threatened by extramural elements, but from within the organizational boundaries i.e insider attacks. It results in breach of confidentiality, integrity and availability of the organization's assets. Insider Threat caused by malicious abuse of authority has exceeded the traditional Trojan attacks and has become the main threat to organizations. Therefore, detection and prevention from Insider Threat is a real challenge due to enormous raw data. This issue is being dealt by research community through machine learning techniques for past few years. In the absence of a carefully crafted middle ground an employee although provided access to effectively perform his/her duty, is able to wreck scaled havoc. Which in turn hampers the organizational productivity and force the organization to shift its focus. Therefore, it is necessary to carefully design the access architecture and a system bounded by the ultimate cherry-on-top to mitigate such attacks.

In this dissertation, we address this critical issue of Insider Threat through comprehensive machine learning based Frameworks.We present four different machine learning-based frameworks that aim to thwart Insider Attacks through multi-dimensional user information by including user logs, emails and psychometric features. Our first machine learning based framework named Supervised Stacked Model (S2M) is tailored towards reporting the class imbalance problem. Multiple low variance filters were tried followed by correlation filters on the output data. As part of this framework, we propose a hybrid ensemble S2M that correctly classifies and differentiate the insider samples from normal activities. Vertical and horizontal re sampling techniques were applied and tested on re sampled data set. The proposed solution is tested on CERT 4.2 dataset which has normal and malicious activities of 1000 users recorded for the year 2010 to 2011 with more than 31 M records. Our second framework is named as Dynamic Weighted-Voting Ensemble

(DWvEn). An ensemble model established on the weighted-voting approach for Insider Threat detection. We have brought together the feature engineering methods and ensemble learners that amicably classify the majority of malicious activities. Our proposed framework dynamically assigns weights to base learners predicted on their competency. We evaluated DWvEn on a substantial and largest publicly available datasets CERT 4.2 and CERT 6.2 by using multiple pre-processing and feature engineering techniques.

As part of our email-based frameworks, we have applied semi supervised machine learning taxonomy on valuable collection of Enron corpus and TWOS datasets for the identification of unlabeled malicious emails and handling the Over-fitting issue in small dataset respectively. The former research is devoted to "traitor detection" which has remained very restricted as compared to "masquerader detection". In this research Class label identification done through clustering algorithm and prediction of malicious emails is carried out by using multiple Machine Learning Classifiers. The frameworks and methodologies presented in this dissertation can assist a broad spectrum of organizations in attenuating Insider Threats.

Conclusively, this thesis presents a comprehensive Intelligent Framework for effective classification of Insider Threats and essential to have multiple Models/ Frameworks depending on the type of datasets being handled.

Dedication

My research work is dedicated to **my wife, teachers and companions** for their Support, Consistent Guidance and Inspiration

Acknowledgments

"All praises to Allah for the strengths and His blessings in successful completion of my Research Work."

With utmost gratitude to **Dr Asif Masood**, for his valuable recommendations and persistent guidance. In accumulating I would like to thank my committee members; **Dr Haider Abbas, Dr Imran Rashid, Dr Amer Ahsan Gilani and Dr Baber Aslam** for all the vital support of profitable commentaries and educational suggestions through out my research work are major assistance to the success of my study and in exploring this focus area.

Table of Contents

List of Figures

List of Tables

Acronyms and Symbols

Abbreviations

ML	Machine Learning
S2M	Supervised Stacked Model
DWvEn	Dynamic Weighted-Voting Ensemble Learning Framework
CERT	Computer Emergency Response Team
TWOS	The Wolf of SUTD
NB	Naive Bayes
DT	Decision Tree
RF	Random Forest
GB	Gradient Boosting
LR	Logistic Regression
KNN	K Nearest Neighbour
SVM	Support Vector Machine
RNN	Recurrent Neural Network
LVF	Low Variance Filter
TF-IDF	Term Frequency–Inverse Document Frequency

Chapter 1

Introduction

1.1 Insider Threat

An "Insider Threat" involves the activities of a privileged and trusted user, who is infiltrating and accessing secret information inappropriately. INSIDER THREAT to the global management is a complex and growing challenge [1]. Any action taken by employees, which is potentially damaging to the establishment is generally unacceptable, that can be an unauthorized data transferal, unlawful use or spoiling of any organizational assets. The two rational reasons existing for this similar threat: firstly, the employees with malicious intentions steal or modification of confidential data, customer information or trade secrets from the organization for their malevolent goals. For example, they use classified data in order to gain commercial benefits or sell them to unauthorized individuals/ organizations. Secondly, employees unintentionally disclose the sensitive material or any key assets to external adversaries [2] [3]. The American Institute of Computer Security (ICS) dispensed a publication in 2006. The gist of which is that an in-house outbreak with the power to abuse the system went beyond traditional Trojan or malware attacks and became the foremost threat to administrations worldwide. Similarly, the Annual Global Fraud Survey of 2012 exposed in their research of about approximately 60% of those fraudulent happenings was actually launched by the same in-House out breakers [4] [5]. The US Cybercrime Survey released by CERT in 2014 publicized in their review of approximately 46% of these (In-House) attacks was more harmful as compared to other outsider threats. Credit card data of more than 27M account holders was stolen in Korean Credit Bureau (KCB) due to abusive access rights

by insiders and also referring to the CSO Cybercrime Survey 1.1 of 2019 almost 70% of the organizations have had minimum one insider attack during 2018.

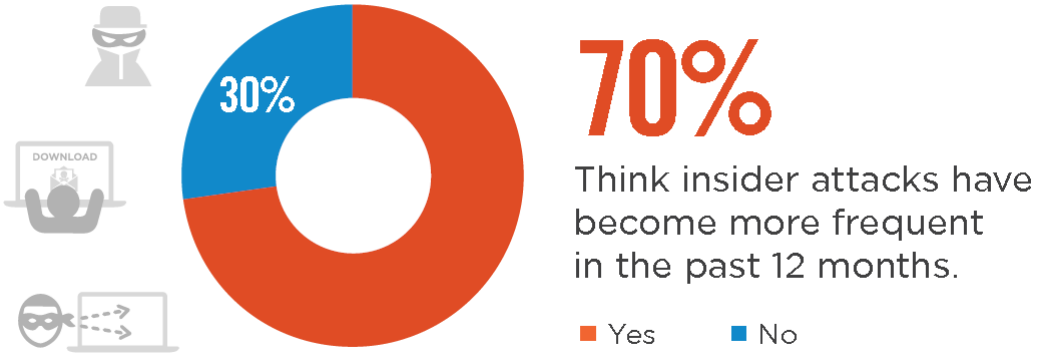


Figure 1.1: Cyber Survey 2019

In today's world, insiders can prove a severe threat to the organization in which they operate due to several reasons, to prevent them from malicious act in any organizational system is an important challenge for cyber security to this very day. Structural security measures are known to insider and he can easily find the loop holes, also policies made by any administration focusing on external threats and safeguards have been implemented accordingly due to which vacuum is created for an insider to steal crucial data from the organization without any hurdle and cause irreparable damage to the organization. Therefore, significant importance has been given to "Insider Threat" when compared with external threats.

1.1.1 Types of Insider Threat

With the revelation of an insider stated above, and to have more understanding of their operations they are further categorized into three types as shown in Fig 1.2.

- **Malicious Insider:** A malicious insider is a kind of threat, where a user intentionally wants to snip data, disclose information or through any other means harm the organization.
- **Careless Insider:** A Threat by means of a careless user happens when employees don't know security guidelines or abide by security procedures, placing the company in vulnerability for mischievous software infections and data disclosures.

Figure 1.2: Types of Insider Threats

- **Compromised Insider:** A compromised insider threat is a malicious user whose email credentials has been manipulated by hacker through credential harvesting, social engineering, phishing e-mail messages or techniques that exploits a vulnerability in order to snip the data or through illegal fiscal transactions.

1.1.2 Signs of Insider Threat

Generic digital and behavioral warning indicators of an insider threat are following.

Digital Warning Signs

- Downloading or acquiring extensive volume of data
- Acquiring secret data not linked with their job description
- Emailing secret data to outsider not linked with the organization

- Accessing data that is outside of their unique behavioral profile
- Various requests for access to resources not linked with their persona
- Unauthorized use of storage devices (e.g., USB drives or CDs)
- Search of secret data on an unsecure LAN
- Moving/Coping files from official restricted folder

Behavioral Warning Signs

- Try to violate security standard operating procedures
- Often visiting office on holidays and during off-time hours
- Shows irritated behavior toward colleagues
- Breach of corporate regulation's
- Pretend to sign a resignation and proposals of new job opportunities

1.2 Artificial Intelligence

Artificial Intelligence (AI) is the newly paved road to the future with its limitless possibilities all around us and is by far evolving gradually. Mainstream AI talks about a human made simulated formation of human-like intellect which can grasp, aim, design, recognize, or practice natural language like any other human being. Machine-learned algorithms are routinely deployed to perform event reconstruction, particle identification, event classification, and other tasks [6]. AI has been under the spotlight over the last few years all around the world. With the innovation of new advancements, and the exposure possible by the Internet today, has brought AI to our doorsteps. This advancement, along with an interest in the social and economic impact of technology which it brings, AI has been placed to the vanguard of many modern debates and researches. Industrial investment in AI is growing rapidly, and the administrative sector is all trying to comprehend how this is beneficial for their populaces. With the proposed Cluster of "Big Data Analysis" and the progress of the Internet as a global rural area has made a haven for innovative AI development and growing services. Innovations

centered, AI have already marked exceptional in health care testing, community safety projects, shipping, targeted cure of diseases, self-serving robots, educational and training simulations and also very evident in entertainment industry. With the internet as convince, AI revolutionizes the way we understand the world around us and it even has the capabilities in boosting the economic growth of any country. AI brings along the philosophy, practice, and method that helps the computers to evaluate, assess, simulate, exploit and discover human intellectual development and conduct in an artificial way [7] [8] [9] [10]. Fig. 1.3 below is depicting human activities corresponding to AI fields.

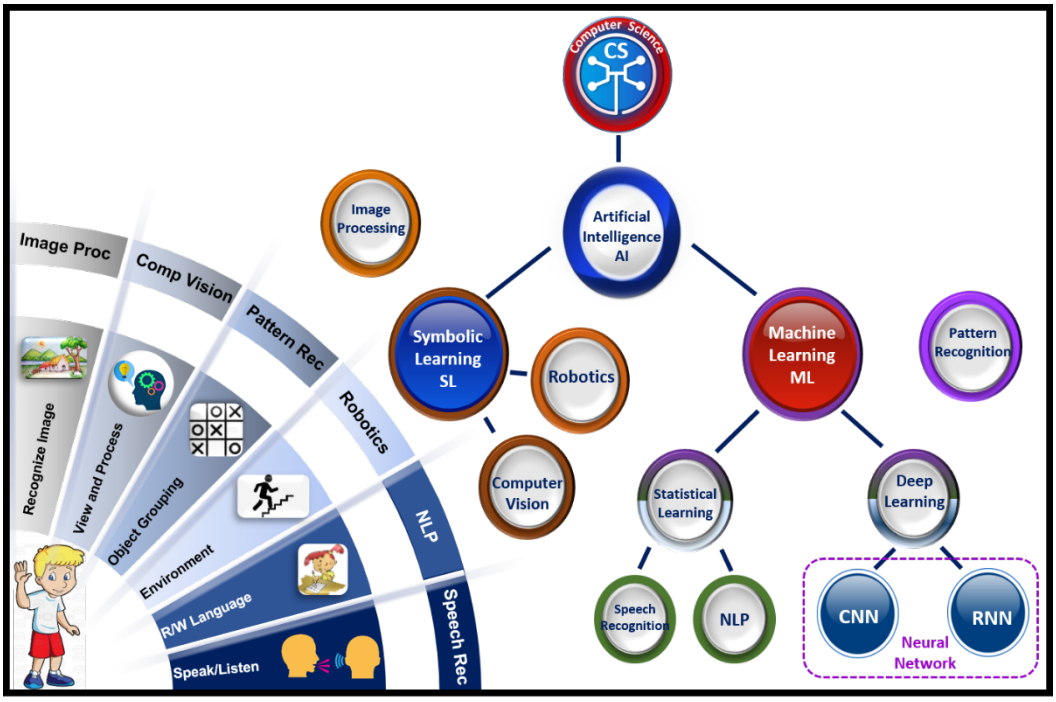


Figure 1.3: Tasks performed by Artificial Intelligence and Human Being

Humans speak/listen ability link with Speech Recognition in AI, environmental learning ties with Robotics whereas read/write capability correspond with Natural Language Process (NLP) and so on. Contributions of AI in real world applications as Siri-Speech Recognition, Gmail-Blocking spams, Netflix-Movie suggestion, Google/Tesla-Self driving cars and Facebook-Suggesting things based on our interest.

1.2.1 Broad Categories of AI

AI ordinarily is distributed into the following two broad types:

1. **Narrow AI:** Every now and then researchers are denoting this to "Weak AI" is

actually a simulation of human intellect at a very limited environment to perform a singular task assigned with extreme precision, with far more limitations than any elementary human mind. Narrow AI is all around us and is obviously more visible in mainstream projects till date. With the machines focus on a singular task its performance has experienced major precision breakthrough in the last decide with significant contributions to the economic growth of the nations all around the world.

Innovations provided by the Narrow AI are:

- Search Engines (Google, Bing, Yahoo,)
- Human Recognition Software's (Siri, IRIS, Facial recognition)
- Self-Driving Automobiles (Tesla, Ducati, BMW)

2. **Artificial General Intelligence (AGI):** AGI refers to "Strong AI" by scientists. It is the future technology that is shown nowadays in movies, like the robots from iRobot dreaming and preforming social activities, the smart intelligence detection system in movie in the EYE or similarly Ultron or Jarvis like intelligent indigenous systems introduced in the famous Iron man, a machine having general intelligence much like the human intellect but without the physical limitations of a human mind which it applies to solve any problem provided by the host efficiently with precision. A Machine having these capabilities and intellect is surely very attractive to many AI researchers, but the road to the achievement in AGI has been bumped with the horrors of Super-machines taking over the humanity or the very host for which it is intentionally assigned to preform like in the famous Terminator movies. But expert's debate that it is something that is more fiction than reality because in reality host is in control not the machine.

1.3 AI Implementation Fields

1.3.1 Military

All over the world, most of the countries are implementing AI in their military infrastructure. China give the impression to be prominent in making drone swarm like in the movie "Angel has Fallen" which can operate autonomously and destroy multiple tar-

gets at a single instance. Similarly, their aerospace manufacturing is developing cruise missiles with inbuilt intelligent indigenous systems to seek out high profile targets in combat. The US has initiated Project Maven [11], which is to gather intelligence from the battlefield and as it outputs the next move to defeat the enemy. The US Army Research Laboratory (ARL) researchers developed an AI technique capable of producing automatic facial recognition for Soldiers working covert operations at night to identify individuals of interest or on a watch list and a similar facial recognition to assist airports and border management. Recent developments in artificial intelligence and machine learning have provided tools with which a computer can now outperform the analytic capability of a human, particularly when data sets are large or when a system relies on many free parameters. The application of machine learning methods has led to dramatic advances in many scientific fields and contexts [12].

1.3.2 Internet of Things (IOT)

IOT has already altered the manner we relate by means of our physical realm. The point in case could be the concept of a Smart City [13], which entails gathering of intuited data from Automobiles, traffic signals, surveillance cameras, Social media, and geo-positioning to help prevent accidents and further improve the response time of necessary action. Further, environmental information that includes weather conditions, rain information, natural calamity all necessary information is collected, analyzed and forwarded to the authorities for earliest possible action.

1.3.3 Cyber Security

AI approaches are also applicable to cyber security solutions to identify newly innovative variations of malware and Trojan attacks [14]. There are many such examples, who have developed an algorithm for detecting malware that have used deep networks for malware classification.

1.3.4 Social Media

Social Media has also incorporated AI in photo tagging, geo tagging, friend suggestions, and sponsored suggestions. The technology identifies a person in a shot and automat-

ically tags them. AI can be used for image recognition in Open Source Intelligence (OSINT) to identify linkages in the available information.

1.3.5 Education

The application of AI to education (AIEd) [15] explores knowledge wherever it transpires, in conventional schoolrooms or in offices, with the principle focus in backing formal learning in line with lifelong learning and endorses the growth of adaptive new learning atmospheres.

1.3.6 Health Care

Intelligent hospitals are the production of cutting-edge networking expertise, particularly the extensively used internet of things, in creating regional medical data podiums for patient archives anywhere, whenever it is required. By the side of the same time, through the mixture of machinery and medication, the remedial procedure can be accusatively with precision and speed. Such as, deep learning for gene prediction, NLP for Electronic medical record [16], pictorial representation as well as imaging for radiology in addition to the use of trained robots to perform surgeries.

1.3.7 E-Commerce

AI methodologies are beneficial in the growth of (Business to Consumer) B2C [17] and (Business to Business) B2B e-commerce structures. AI is endorsed predominantly for product choice and confirmation, arbitration, sales, solving real-world preparation complications and enhancing servers' possible scalability, generating automated replies to the customers, and assessments on bundling deliveries or pricing of goods more proficiently.

1.4 AI for Insider Threat

With a substantial growth in computing, increased investment and power of data algorithms, today we can leverage AI for detection of Insider Threats. Organizations are looking into using Artificial Intelligence (AI) to detect, contain and neutralize Insider Threats. Much better approach is to prevent Insider Threats from happening to begin

with. Primary objective is to develop the AI schemes to efficiently pick up and start reading classical patterns of Insider Threats [18] [19] [20] [21] [22]. Which might contain but not limited to; data or files transfer through computer networks, email accounts or any other abnormal or peculiar behavior or account activity which does not correlate with routine work and employees' given role and tasks. In isolation, these indicators could be outliers but when these signs are put through the trained models of AI, they can create an alarming scenario for the organization. AI tools assist in detecting those anomalies which are very complicated in nature and challenging to detect or at worst could not be detected at all by the human eye.

When we look into the data that is available for checking the behavior of Insider Threat, some times we ignore the human side of issue [23]. To solve these issues, many researchers have made a significant contribution over the years. Textual analysis with machine learning algorithms has shown remarkable success. Salima et al. [24] has suggested that supervised learning algorithms offer better recall when compared with semi-supervised and unsupervised techniques, when they obtain more details. The TWOS data set has been acquired, processed and analyzed on different algorithms. TWOS data were collected during a gamified competition designed to detect relevant cases of malicious Insider Threats. Competition activates user interactions within competitors/companies, in which two sorts of behaviors (normal and malicious) are generated. When we look into the malicious behavior, two sorts of malicious stages were designed to reach the activities of two different varieties of insiders i.e Masqueraders and Traitors [25].

1.5 Overview of the Proposed Research

The proposition of this reading is that the various in-house attacks and their undesirable occurrences can be sidestepped by designing an AI which is capable to adjust to the undesirable variations on textual and contextual evidence. While being precise, we hypothesize insider threat exposure by comprising factors such as the email logs, user profiles, and psychometric data. In the direction of substantiating our assumption, we have established the following numerous frameworks depicted in Fig 1.4. Each one of the proposed frameworks is designed to alleviate insider attacks using multiple techniques on diverse datasets.

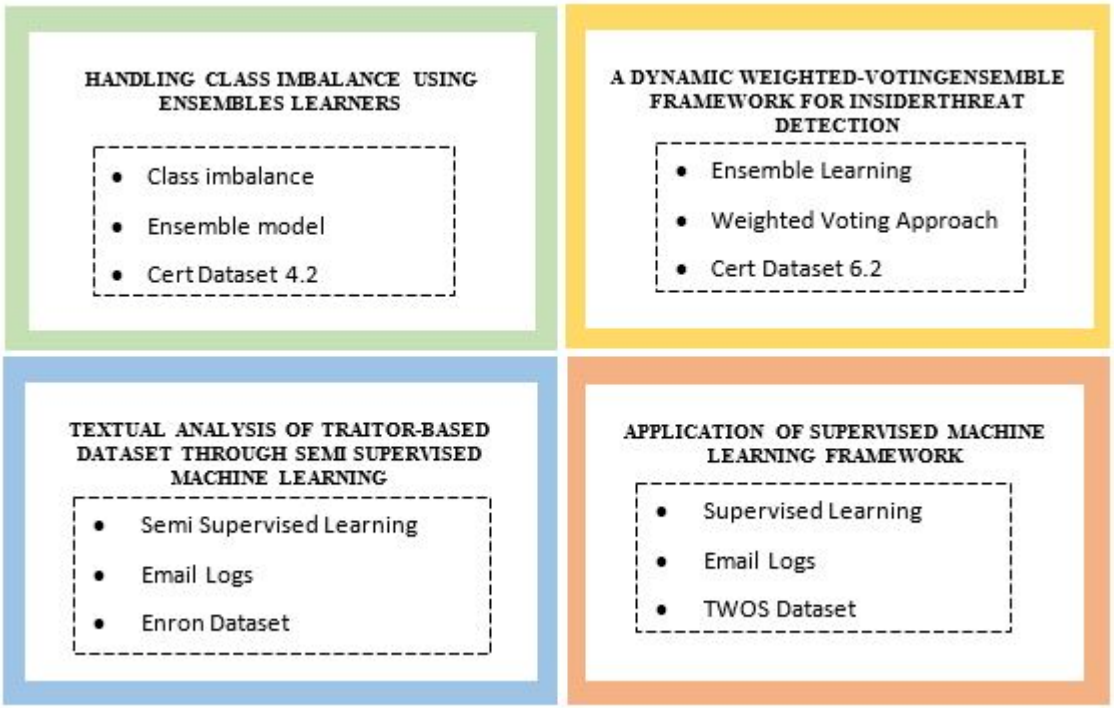


Figure 1.4: Abstract outlook of Frameworks

1.6 Thesis Document Organization

The main cardinals of this research work are prepared subsequently as follow:-

- Chapter 2 discusses Contextual and Associated Work.
- Chapter 3 talks about Handling Class Imbalance Using Ensembles Learners.
- Chapter 4 presents Dynamic Weighted-Voting Ensemble Framework For Insider Threat Detection.
- Then, in Chapter 5 we present Semi Supervised Machine Learning For Un-Labelled Traitor-Based Data Set.
- In Chapter 6 we discuss our Application of Supervised Machine Learning Framework.
- Lastly, in Chapter 7, we propagate our assumptions and the yet to come research blueprint.

Chapter 2

Preliminaries

This chapter focuses on different approaches of Insider Threat detection along with in-depth review of research papers and case studies on frameworks, models, and techniques being used in this field. Different dimensions and main elements of the problems related to Insider Threat detection are introduced in section 2.1 while literature review on different tactics for the charge of Insider Threat exposure is mentioned in section 2.4.

Insider Threats are one of the major factors causing an increase in cyber security threats faced by the information systems within an organization. Nonetheless, there exist appropriate network mechanism for safety and security of the information yet insiders being part of the system can easily intercept the data. This authority to the information give insiders a legitimate access to confidential documents. As a result, identifying and averting the dangers of Insiders attack have turn out to be a growing challenge for the organizations thus making it a tough task to ascertain the root cause.

While investigating these challenges; a dataset is prepared containing email logs which plays a vital role as it helps in stalking Insider Threats involving collaborating traitors, Textual Analysis and Social Media exploration. This dataset often results in class imbalance. According to previous researches, technological progress in data mining has led to significant increase in raw data during recent years which on the other hand has resulted in serious class imbalance among major and minority classes adversely affecting and producing bias in predictive analysis of intelligent algorithms. Class imbalance is a common problem found in most domains including churn prediction, medical disease diagnosis and malicious threat detection. Ensemble techniques have gained much

popularity over the years due to their high classification rate.

2.1 Background

Insider attacks are becoming more destructive to any administration than outsider outbreaks, afterwards there are substantial expenses connected in vindicating insider outbreaks. Utmost safety implementations and practices established as a result are underfitting to knob diverse Insider Threats ever since their resolve is to avert incursions and occurrences from outer world only. Statistically, the safety exploration communal remained feeble to officially term an "insider" attack outstanding due to the dissimilarities towards their appearance, circumstances and more explanations. In a broad nous, an individual can describe an insider as a mischievous employee who is presently or at some instance had authorization to an establishment's protected possessions as well as tangling in any one of the subsequent deeds [26]:

- Unsanctioned withdrawal of data
- Meddling with the possessions of an administration
- Damage otherwise confiscation of acute files as well as resources
- Snooping and package sniffing by mean of hostile intent
- Satirize new consumers via public profiling

Insider Threat classification [27] can be distributed into two core groupings namely:

- Classification based on inside mis-users
- Classification based on various types of knowledge

2.1.1 Classification Based on Inside Mis-users

One of the most primitive cataloging of in-house abuse of machines was suggested by Anderson [1980] who later then extricates among three sorts of unlawful insider users, well-arranged by the arising struggle of their discovery from audit tracks.

1. **Masqueraders:** Masqueraders can be either one outside attacker who has bypassed the security controls and breached the machine, or in-house employee whose propose is to exploit a different user's authorizations in a directive to achieve some mischievous accomplishment.

2. **Misfeasors:** Misfeasors are the employees who do not pretense, but in its place misuse self-rights in directive of manipulating the machine rendering to their necessity.

3. **Clandestine:** Clandestine represent super users with the competency of residing underneath the detector of security protocols thus not activating them, which they achieve and comprehensively identify how the machine protocols work and making them utmost tough to identify.

2.1.2 Classification Based on Various Types of Knowledge

Malicious insider dangers are further separated into two crowds based upon the volume of information they have:

1. **Traitors:** Traitors have thorough awareness about the systems they interact with every day, as well as the authentic safekeeping protocols of the system. Traitors ordinarily action on their personal behalf and for that reason use their individual authorizations for mischievous activities.

2. **Masqueraders:** Masqueraders ought to require not as much of understanding of the system than traitors. They are assailants who snip the credentials of a different authentic employee, and then use them for accomplishing a mischievous action on behalf of another manipulator.

Insider menace investigation led as a result to report the subsequent inquiries:-

- In what way we recognize an insider surely?
- The perspective of a user being an insider?
- Exactly how to fortify resistances in occurrence of an insider outbreak?

Insider attacks practices are largely pigeonholed as shown in Fig 2.1 below.

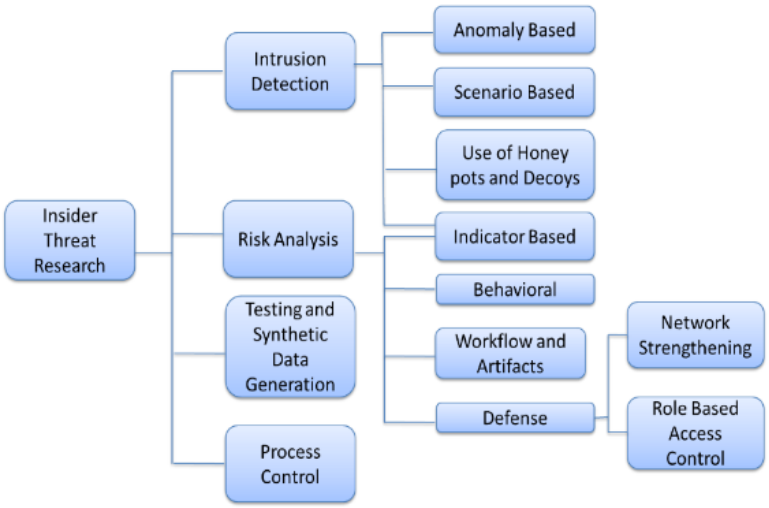


Figure 2.1: Taxonomy of Insider Threat Detection

2.2 Publishers

This part of document briefly describes the previous state of arts techniques that are used for dynamic ensemble selection of classifiers. Methodology and results of previous studies in literature are discussed here.

Research procedure is passed out in an organized method. Statistics is collected by an exploration progression comprising discovery of researches appropriate to the working and picking out the most significant amongst them. These nominated researches are then investigated for quality and data is mined out from them. Following research publishers are considered for this research.

- IEEE
- Springer
- Elsevier

Fig 2.2 represents the papers selected from different databases.

2.3 Quality Assessment

There were definite quality dynamics that were used for carrying out learning for the research papers selected from above databases. These quality aspects are defined below:

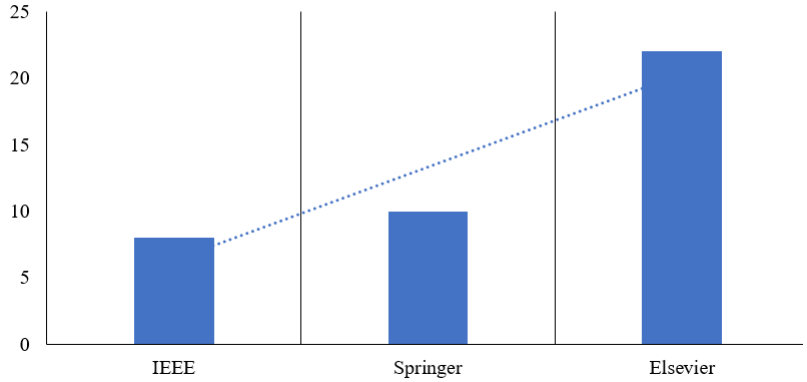


Figure 2.2: Papers Selected from Scientific Databases

2.3.1 Effective Technique Proposed

The utmost significant element about the nominated paper was that it ought to have proposed some practice, method or model regarding dynamic classifier selection. All remaining papers were eliminated from the search results.

2.3.2 Results Validation

Altogether those researches that do not contribute any valuation of the outcomes reinforced by the endorsement of some dataset are omitted from the search results.

2.3.3 Repetition

Only those papers that contain some innovative and exceptional studies are reflected. Those signifying identical new approach or simulations are encompassed.

2.3.4 Recent Research Work

Maximum numbers of the publications from the latest 5 years that are from 2015-2020 and to the present-day year (2021) studies are collected for scrutiny as they are the peak updated ones. Fig 2.3 represents the papers selected per year.

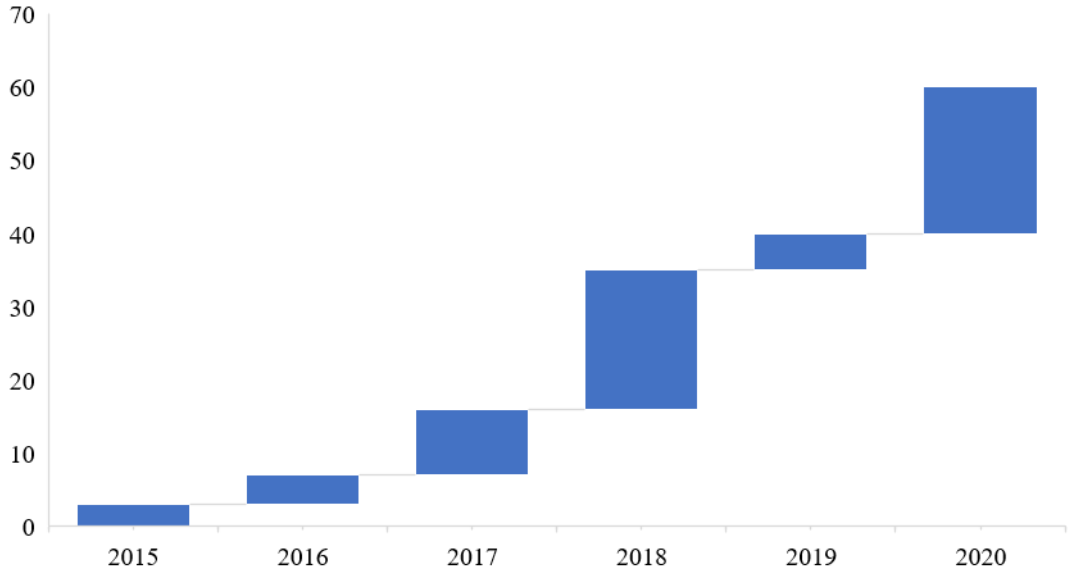


Figure 2.3: Papers Selected Per Year 2015-2020 (Year 2021 in progress)

2.4 Related Work

2.4.1 Insider Threats

One of the serious concerns for any organization is the damage caused by insiders. Eventually, significant acknowledgement was given from both the research and industrial communities. Though, it is very difficult to completely abstain the malicious insider during its launching stage when it is being executed. However, to stop and reduce the malicious attacks, different models have been proposed by researchers from all over the world.

In 2016, Hongmei Chi, et al. [28] studied the linguistic analysis to find whether a person is an insider threat to an organization. Text data has been analyzed using Linguistic Inquiry Word Count (LIWC) then it is evaluated on Dark Traid Model. Enron, Cert and Real or Spiel dataset are used to study the proposed algorithm. Datasets are categorized in different clusters using K-means and an average score is assigned to them. The level of threat of any employee will be based upon how many categories the employee scores above the average.

In 2018, Jianguo Jiang, et al. [29] have anticipated an algorithm to construct an operator psychological outline grounded on the emotional scrutiny of their email content and network surfing. To assess the presentation of the anticipated detection system,

CMU-CERT dataset v4.2 and Enron Dataset have been used. The results have shown that model can precisely and proactively predict the malicious insider. This technique lacks the behavioral aspects of insiders. Accumulating of this feature can improve the outcomes.

In 2019, Charlie Soh, et al. [30] have proposed framework built on deep learning methods stated as Gated Recurrent Unit (GRU) and skip gram. The sentiment profiles of the employees are created, and anomaly detection is performed. Then employees are ranked on their anomaly score. Real-world Enron email corpus dataset has existed to be used to assess the framework. The outcomes have revealed that the framework out performed previous models. Profile generated through framework are of great benefit for future research.

In 2020, Duc C. Le, et al. [31] proposed an intelligent user centric arrangement, centered on machine learning for insider threat discovery. For closely monitoring malicious insiders; users' malicious behavior was observed and watched through granular machine learning analysis under realistic conditions. To enable the actual approximation of the system performance, a comprehensive analysis of popular insider threat scenarios with multiple performance measures are provided. Assessment outcomes suggest that the Artificial Intelligence (AI) based monitoring and detection system can detect new insider threats from unlabeled data with a high accuracy since it can be very effectively trained from restricted ground truth. The research says approximately 85% of such insider threats are successfully identified with as low as 0.78% of false positive (FP) rate. Naghmeh Moradpoor, et al. [32] addressed the detection of insider threat on massively imbalanced dataset that employ well known balancing technique "spread subsample". The results showed that using this technique for balancing of dataset did not enhance outcome metrics; but it only expands the while taken to construct and test the ideal. Moreover, it was realized that in succession to the nominated classifiers with constraints other than the default ones for both stable and imbalanced situations has an impact nevertheless using an imbalanced dataset the impact is significantly stronger. Esteban Castillo, et al. [33] described different methodologies to identify malicious data in email communications through a combination of natural linguistic processing and machine learning utensils. To recognize doubtful communications and separate them from non-suspicious, nonthreatening email neural network was designed and tested on word embedding representations. Results through experiment showed that with and without

recurring neural layers back-propagation outdoes present state of the art procedures that contain supervised learning algorithms with features of texts. Mohamed Abdul hussain, et al. [34] recommended a filtering method for junk mail which consisted of artificial back propagation neural network (BPNN) practice to filters the spam emails. Enron dataset was used with TF-IDF algorithm which extracted the features and transformed these into frequency after the pre-processing. Mutual information technique was used to select best features. By means of BoW, n-gram, and chisquared approaches, performance of classifiers were measured. BPNN ideal was matched with Naıve Bayes and support vector machine based on accuracy, recall, precision and f1-score. The spam system claimed to attain the accuracy of approximately 98% with cross-validation. Gaoqing Yu, et al. [35] proposed a technique grounded on data insertion for producing phishing emails, this proposed technique can intensify the phishing sample counts even without altering malicious elements; thereby resolving the problematic spatial bias for training of the given model and can decrease the variance in numerical individualities amongst malicious and benign illustrations to a limited range. Six source initiators and a message association chooser was executed founded on variances in the emails HTML content of the Phishing dataset and the Enron dataset. Newly produced mockups can be castoff as a trainer to a classifier with stronger simplification capacity.

In recent studies of 2021, Shuhan et al. [36] carried a survey on growing challenges of Insider Threat and identified the class imbalance issue as one of the major and important issues. Duc C. Le et al. [37] created anomaly detection ensembles by combining different computational schemes that improve the performance of insider threat detection. Evaluation results show that 60% of the malicious activities are detected under 0.1% budget. Following that, Ujwala Sav et al. [38] proposed a study on anomalous behaviours of insiders by using data processing and anomaly detection algorithms.

2.4.2 Artificial Intelligence

Insider Threat recognition is the most vital trial for safety in their administrative setups. These are the users of a body, posing peril to it by acting out any roguish actions. Present approaches to the exposure of insider extortions are grounded on artificial intelligence approaches. In 2019, Sergiu Eftimie, Radu Moinescu and Ciprian Rcuciu [39] epitomize an interdisciplinary skirmish to proactively recognize in-House dangers, using

ordinary language processing and character profiling in an organization. Contours were established for germane insider threat varieties by means of the five-factor prototypical of behavior and remained castoff in a conceptual recognition structure. The system hires a third-party cloud facility which habits ordinary language processing near profile diverse kinds of users and their behaviors centered on personal content. An evaluation was made over the probability of the system using a community dataset. Azamat Sultanov and Konstantin Kogos [40] shares non-invasive technique for identification of insider threat established on anxiety recognition using employee's keystroke forces at work, let's say that the invader is in impressions of stress through conducting unlawful activities, as it disturbs the behavioural appearances of an individual. Anticipated technique practices equally supervised and unsupervised machine learning algorithms. Outcomes exposed that stress can deliver exceedingly valued evidence for insider threat discovery amongst workforces.

In 2020, Mathieu Garchery and Michael Granitzer [41] introduced ADSAGE in the direction of detecting abnormalities in review log events demonstrated as graph ends. This is the leading technique to achieve irregularity recognition at edge level although supportive to equally edge sequences and attributes, can be a numeric value, clear-cut or textual. ADSAGE is used for finest, occurrence level insider threat recognition in diverse audit archives from the CERT instance. The suggested development was appraised on validation, email tributary of traffic and web surfing logs from the CERT insider threat datasets, as fine as on real-world verification proceedings. ADSAGE is actively efficient towards identifying variances in verifications, demonstrated as consumer to computer collaborations, and in email transport network. Jari Jääskelä [42] shares a research of insider threat exposure, UN supervised and semi-supervised irregularity exposure. The performances of several unsupervised anomaly finders were assessed then to increase interpretability practices of building rule-based accounts for the isolation forest remain assessed. Tests were executed on CMU-CERT dataset, and are freely obtainable insider threat dataset with logon, detachable device and HTTP log data. The upshots exposed that active anomaly detection aids into placing correct positives upper arranged in the list, dropping the sum of data analysts must analyze. Malvika Singh and B. M. Mehtre [43] offered behavior-based insider threat exposure technique. The conduct was pigeonholed by user activity and Isometric Feature Mapping (ISOMAP) existed for feature mining and Emperor Penguin Algorithm is the best feature assortment. The

structures comprise a period feature and frequency feature. To end with; a Multi-fuzzy classifier was injected with mainly three corresponding inference engines F1, F2, F3, to categorize consumers as standard or mischievous. Recommended technique was verified using CMU-CERT Insider Threat dataset for its analytics, suggested way outclasses on the subsequent attributes: accurateness, remembrance, f-measure, and AUC-ROC parameters. The Insider Threat detection outcomes display a noteworthy development above current routine.

2.4.2.1 Machine Learning Classification Techniques

In 2015, Zahra Nematzadeh et al. [44] presents the effect of using K-fold cross validation proceeding accuracy through applying different algorithms of machine learning. Neural Network, Decision Tree, Naïve Bayes and Support Vector Machine algorithms were used with diverse kernel values to classify Wisconsin Diagnostic Breast Cancer (WDBC). Different datasets from UCI were used for comparison. Results were tested on different values of K for K-fold cross validation. Study revealed that by increasing value K, computational cost increases as more folds are required for training and it see to not have any trivial influence on accurateness i-e by exhausting higher value K does not mean that accuracy will be increased.

In 2017, R. Ani et al. [45] investigated that better accuracy can be provided for predicting diseases by using machine learning algorithms and proposed a model that uses random forest as base classifier and for feature projection Linear Discriminant Analysis was used. Results showed that Linear Discriminant Analysis gives better results than Principle Component Analysis. Highest accuracy achieved by using this model was 95% that outperforms other techniques in state-of-art.

In 2018, David A. Omondiagbe et al. [46] examined machine learning practices by means of other feature reduction approaches and suggested a technique that uses Linear Discriminant Analysis to diminish features dimensionality. This condensed feature dataset was served to the Support Vector Machine for sorting. Wisconsin Diagnostic Breast Cancer (WDBC) Dataset was recycled for exercise and endorsement. An accuracy of 98.82% was achieved by using this technique.

In 2019, Quinlan D. Buchlak et al. [47] presented a systematic review on different machine algorithms and their usage in machine learning applications. Systematic study

provided 6866 results by using accuracy, specificity and sensitivity as performance statistics. Results showed that frequently Neural Network, Support Vector Machine or Linear Regression remained in use. Out of which Neural Network have sufficiently higher accuracy then Support Vector Machine and Support Vector Machine have sufficiently higher accuracy then Linear Regression. Neural Network outperformed other supervised learning techniques. Abdoulaye Diop et al. [48] present his research a step in the direction to development of a consumer and individual behavior study agenda by suggesting a behavior anomaly detection model. The ideal chains machine learning sorting practices and graph-based approaches, entrusting in linear algebra and in parallel processing methods. With the usage of some discovered classifiers, adds the outcomes up to 99% precision.

2.4.2.2 Hybrid Classifiers

Hybrid approaches have been used for classification, making use of multiple classifiers for classification toward expansion of the performance of individual classifiers. In 2017, Seok-Jun B and Sung Bae Cho [49] proposed a hybrid structure of convolutional neural network (CNN) and learning classifier system (LCS) for IDS, called Convolutional Neural-Learning Classifier System (CN-LCS). CNN, a unique of the deep learning techniques for appearance and pattern sorting, organizes questions thru forming standard activities of database. LCS, amongst the the reformed heuristic search algorithms centered on genetic algorithm, determines fresh directions to distinguish abnormal behaviors in addition to the CNN. A trial with TPC-E benchmark database demonstrates that CN-LCS harvests the finest sorting accurateness matched near additional state-of-the-art machine learning algorithms. Supplementary breakdown by t-SNE algorithm discloses shared patterns amongst extremely unclassified probes.

In 2018, Lipo Wang et al. [50] suggested a practice aimed at insider threat detection commencing time-series sorting of user actions. Primarily, group of single-day features is added as of the operator action logs. A time-series function route is next assembled from the data of each single-day, concluding a dated epoch of interval. The marker of every one time-series feature vector is mined from the broken up reality. To categorize the excessive ground-truth insider threat statistics entailing of only an insignificant amount of occurrences, we work on a costly statistics modification method that under

samples the non-malicious class occurrences arbitrarily. Two-layered deep auto encoder neural network were engaged and associated its attributes with other frequently castoff classifiers: random forest and multilayer perceptron. Positive outcomes were acquired by weighing suggested method by means of the CMU Insider Threat Data, which is the only widely accessible insider threat data set residing of around 14-GB of web-surfing and email records, alongside with logon, device linking logs, and file transmission information.

In 2019, Ivan Homoliak et al. [51] suggested structural taxonomy and novel classification of investigation that subsidize to the institute and disambiguation of insider threat occurrences and the defensive elucidations recycled counter to them. The goal of the tagging was to arrange information in insider threat study however by means of a present ashore model technique for demanding writings. The offered classification portrays the workflow amongst certain types that comprise of incidents and datasets, scrutiny of events, models, and security solutions. Exceptional devotion was compensated to the explanations and taxonomies of the insider threat.

In 2020, Jahanzaib Malik et al. [52] recommended a tool that includes of a hybrid Cuda-enabled DL-driven architecture which employs the prognostic influence of Long short-term memory (LSTM) and Convolutional Neural Network (CNN) for an effective and well-timed discovery dangers and occurrences. An existing unique dataset CICIDS2017 and average performance assessment attributes were engaged to comprehensively assess the offered research. Fallouts were matched with fashioned hybrid DL-architectures and current benchmark algorithms. Study exhibited that the offered method out-performs in standings of recognition correctness with minor trade-off with speediness. Following R. G. Gayathri et al. [53] recommended a tactic for insider threat sorting that was encouraged through the efficiency of pre-trained deep convolutional neural networks (DCNNs) for image classification. In the offered method, structures were mined from practice patterns of insiders besides these structures were symbolized as imageries. Henceforth, imageries were castoff to signify the source access patterns of the personnel inside an institute. The suggested line of attack was assessed consuming the MobileNetV2, VGG19, and ResNet50 pre-trained models, and a standard dataset. Investigational outcomes presented the proposed manner is operative in addition out does other state-of-the-art approaches.

[54] proposed a stacked model that is more accurate than Random Forest, Naive Bayes, and Decision Tree C4.5 in detecting insider attacks. Various techniques are explored to classify association rules. Experimentation was performed by using Weka's library. Results indicate that the proposed method is light weight and has better performance accuracy than state-of-art techniques. [55] carried out a survey on deep learning techniques for insider threat detection. Study show that the existing classification algorithms were based on supervised learning methods, which failed to categorise unknown scenarios. It is critical to accurately predict normal or malicious activities, both known and new to the model, and to detect them with a low false positive rate. However, stacked auto encoder is a deep learning architecture that finds complicated patterns of data and generates the best representation of inputs, even for attacks that are unknown to the model.

2.4.3 CERT Datasets

Over the years, researchers have been studying different techniques [56] [57] [58] [59] for insider threat detection. In 2017 [60] Tuor, A., Kaplan, S., Hutchinson, B., Nichols, N., and Robinson, S. conducted experiments on CERT 6.2 dataset of 4000 users records. Their proposed framework collects raw data from system logs and feeds them into a feature extraction system. In return, it outputs one vector for each user per day and finally passes it to a deep neural network (DNN) giving an (average) anomaly score of 95.53 u percentile. Authors used only weekdays and ex- cluded weekends for their framework. Repalle, S. A., and Kolluru, V. R. [61] used different datasets including CTU 13 dataset that contains botnet behavior, normal and background traffic. Further, applied both supervised and unsupervised algorithms like K-Means clustering and One-class Support Vector Machines. K Nearest Neighbor (KNN) performed best on the dataset.

In 2018 [62] Yuan, F., Cao, Y., Shang, Y., Liu, Y., Tan, J., and Fang, B. used CERT 4.2 dataset to test their framework. The paramount phase draws the distant chronological features of employee conduct by the Long Short Term Memory (LSTM) networks and outputs the anticipated courses. These feature trajectories are then reformed into fixed feature matrices. Which are fed to the CNN in the second phase, to categorize these as normal or otherwise. Lo, O., Buchanan, W. J., Griffiths, P., and Macfarlane, R.

[63] Tested Hidden Markov Method (HMM) on a CERT 4.2 data. The authors applied distance measurements comprising HMM, Damerau–Levenshtein (DL) Distance, Jaccard and Cosine Distance. Results show that HMM technique products the uppermost disclosure ratio at 0.69 (48 out of 70) while Jaccard Distance produces the lowermost discovery ratio at 0.35. Their model offers better speed as compared to heavy algorithms like Neural Network. However, the HMM practice acquired more than 24 hours.

In 2019 [64] Le, D. C., Zincir-Heywood, A. N. performed experiments on CERT 5.2 dataset, that has events of 2000 users recorded for 18 months. Experiments were performed using data from first 37 weeks i.e. 50% of the time period. Results showed that false alarm rates were comparatively low with insider detection rate of 75% with Random Forest. However, training data was limited to 400 normal and malicious users. Lu, J., and Wong, R. K. [65] experimented with CERT 6.2 dataset and proposed a system based on Recurrent Neural Networks (RNN). Their system is divided in two parts; (1) historical user's computing usage behavior analysis, (2) instance online monitoring detection process. The proposed framework performs well with an AUC of 0.9. However, do not address the class imbalance issue and also reduced the dataset by 70% for training. In the same year, Hu, T., Niu, W., Zhang, X., Liu, X., Lu, J., and Liu, Y. [66] applied deep learning for user authentication by mapping five mouse actions (click/ move/ drag/ stay/ scroll) generated by a user to imageries and exercise these projections through the Convolutional Neural Network (CNN). The results were reported with 2.94%, of incorrect approval rate and 2.28% of incorrect dismissal rate in an authentication time of 7.072 seconds for 100 images. Kim, J., Park, M., Kim, H., Cho, S., and Kang, P. [67] applied Gaussian density assessment, Parzen window density estimation, Principal component analysis and K-means clustering on CERT 6.2 dataset. They reported that the best detection rate was yielded by Parzen; 8 out of 21 cases and works well for imbalanced datasets. Singh, Malvika and Mehtre, BM and Sangeetha, S. [68] also proposed an Ensemble of LSTM and CNN on CERT 6.2 dataset with Attack Detection Rate of 85%. However, no preprocessing or featuring engineering techniques were applied in their framework. Recent studies [69], [70] [71] [72] [73] [74] focus on insider threats but still do not address the class imbalance issue.

2.4.4 Dynamic Weighted-Voting Ensemble Learning

Insider Threat detection is presenting itself as a resource taxing and a testing problem for the research community, thereby creating a challenging environment not only for government but also for cyber security organizations. Recent analysis and survey reports [75], depict an impression of the insider threat research literature. Liu et al. [75] examined the different natures of insider threats such as traitor, masquerader and unintentional perpetrator and also highlighted cybersecurity matters, such as malware and advanced threats. Padayachee et al. [76] concentrates on the opportunity concept. In order to undermine and neutralize the threats, criminology related opportunity theories are being applied, also opportunity-reducing techniques were introduced by them. Legg et al. [77] projected a conceptual model for insider threat basing on psychological and behavioral notes and observations. Cybersecurity Analysts can argue and draw hypothesis pertaining to potential insider threat basing on observations from the real-world.

A hefty amount of data is obtained by Cybersecurity organizations every day. Machine Learning techniques [78] are one of the significant solutions available with us in this era. Machine Learning models train on huge quantities of data available and sense malicious activities. Gavai et al. [79] used administered techniques to detect deviation from normal behavior applying features and unsupervised techniques for early "quitter". ROC score of 0.77 and a classification accuracy of 73.4% was achieved by the structure. Several other ML techniques, for example Bayesian-based methodologies [80], decision tree, and self-organizing map [81] were also examined and analyzed for insider threat recognition.

Few models in the research signify the imperativeness of sequential information while addressing the insider threat and is profoundly linked with human factors such as behaviors [82], [83], [84], [85].

From the literature, it can be observed that there is a lack of pre-processing methodology pooled with ensemble techniques. Moreover, a small portion of CERT dataset is subjected to tests. Whereas Insider threat detection poses a critical class imbalance issue when tested/applied on the complete dataset. Dealing with such datasets needs the application of certain practices and methodology. However, studies and research work does not highlight the number of insider class spotted, and the performance metrics applied are also not appropriate for such hitches. Many studies and researches are

lacking Confusion Matrices for the applied models.

2.4.5 Traitor Based Dataset

In 2017, Muhammad Nabeel Asim, et al. [86] has compared the effectiveness of nine well known features ranking (FR) metrics. They have used six benchmark datasets including Enron dataset using Support Vector Machine (SVM) and Naïve Bayes (NB) classifiers. Enron dataset using SVM in micro and macro cases, Distinguishing Feature Selector (DFS) outperformed the other metrics. In all intentions OR metric shows trivial outcomes were matched to other metrics. Tarannum Zaki, et al. [87] conducted a study to examine big data security challenges in the arena of email communication setup on the Enron email dataset. The results of this study have shown, by using big data analysis phishers or hackers can find out the comportment and behaviour of the electronic mail consumers. These behavioral patterns can provide the sustenance to them for malicious cyber activities.

In 2018, for the taxonomy of spam email, a classification technique was suggested by Shi L., et al. [88] which was grounded on ensemble learning and decision tree. Researchers argue that ensemble learning is effective in malicious and non-malicious email grouping. A separate dataset "SPAM-Email" has also been generated for these type of research works and experiments. Naive Bayes, C4.5, KNN and SVM algorithms have been applied on the dataset and the accuracy of approximately 94% has been claimed in research literature on the proposed work by the authors.

In 2019, Research devoted to "traitor detection" has remained very restricted as compared to "masquerader detection". One of the hypotheses of this deduction could be that masquerader detection is a comparatively more straight forward and unpretentious than traitor detection [89]. Further contended by Salem, et al. [90] stated that "a masquerader is likely to perform actions inconsistent with the victim's typical behavior". Main task of this research is malicious email classification. It can be implemented in different application such as priority-based filtering of messages, separating SPAM and conveying communications to user files. The Single main concern is the representation of messages, and feature selection. Feature collection is very important, one should choose which attributes to practice and how to share those structures for improved outcomes. Manco, et al. [91] proposed the following three sorts of features to study in email:

categorical transcript, unstructured writing and numeric statistics. Association of these features and data is another approach for classification which can be used. In categorical text "to" and "from" fields are included. In unstructured text very well-defined type of data can be used. It is quite different from categorical data. However, these fields are generally preserved identical as the unstructured writing pitches, with the modules supplementary to the basket of words [91] [92]. Same grounds have been set up to be appropriate for automated email grouping, perhaps not as suitable as the unstructured data statistics [92].

In 2021, Zhangdong, et al. [93] presents a content-based picture retrieval technique for traitor tracking. The DenseNet network is used to reinforce statistical features. To safeguard copyright and user information, a one-way hash method and XOR operation are utilised, and a reversible information concealment scheme is used for traitor tracking. Farhan, et al. [94] examined the violations and subsequent moderating of the 2020 US Presidential Election debate on Twitter, a popular micro-blogging site. They focus on quantifying plausible causes for suspension by identifying suspended users (Case) and comparing their behaviours and properties to (yet) non-suspended (Control) users, drawing on Twitter's rules and policies. Dataset of 240M election-related tweets made by 21 users was used for experimentation. Experiments results revealed that Suspended users breach Twitter's rules at a higher rate than Control users across all elements considered i-e hate speech, offensiveness, spamming, and civic integrity. Lakshit [95] presents a survery report focusing on AI and cyber security and applications of AI in various businesses. First, the purpose of integrating AI with cyber security is defined, which ensures that it can defend against numerous attacks and does not allow a single attack to effectively circumvent safety standards into systems. Following that, deep learning, one of the approaches for fighting against cyber-attacks, is discussed, along with its models (Deep Belief Network, Recurrent Neural Network, and Convolutional Neural Network) and datasets (CERT, TWOs). Finally, it is demonstrated that AI is not only confined to cyber security but also being used in a variety of areas, including education, robotics, automation, and health informatics.

Summary of literature review is tabulated in Table 2.1.

Table 2.1: Literature Review Summary

Ref	**Dataset**	**Model/ Classifier**	**Detection Rate** (%)
[34]	Enron Dataset	Back Propagation Neural Network (BPNN), TF-IDF Algorithm	98
[66]	TWO's Dataset	Deep Learning Techniques	97
[64]	CERT v6.2	PCA, LSTM, Recurrent Neural Network (RNN)	95
[67]	CERT v6.2	Gaussian Density Estimation, Parzen Window Density Estimation, PCA, K-Means	94.79
[52]	CERT v6.2	Convolutional Neural Network (CNN), Long Short Term Memory (LSTM)	94.4
[88]	Enron Dataset	Naïve Bayes (NB), Support Vector Machine (SVM), KNN	94
[62]	CERT v6.2	LSTM	92.7
[68]	TWO's Dataset	Multi State Long Short-Term Memory (MSLSTM)	90.4
[51]	CERT v6.2	Ensemble of LSTM, CNN	90
[31]	Enron Dataset	Granular Machine Learning Analysis using Realistic Conditions	85
[96]	CERT v5.2	Random Forest, Linear Regression, Artificial Neural Network	75
[43]	CERT v1	Time-Based Features, Frequency-Base Feature	
[28]	Enron Dataset	Linguistic Inquiry Word, Count (LIWC)	
[29]	CERT v4.2, Enron Dataset	Sentiment Analysis, Network Browsing	

[30]	Enron Dataset	Gated Recurrent Unit (GRU), Skip Gram	
[32]	Enron Dataset	Spread Subsample	
[33]	Enron Dataset	Natural Language Processing (NLP), Recurrent Neural Network (RNN)	
[35]	Phishing Dataset, Enron Dataset	Natural Language Processing (NLP)	
[86]	Enron Dataset	Big Data Analysis	
[89]	Enron Dataset	Feature Selection	
[92]	Enron Dataset	Support Vector Machine (SVM), Naïve Bayes (NB)	
[79]	TWO's Dataset, Enron Dataset	Support Vector Machine (SVM), Naïve Bayes (NB)	
[60]	TWO's Dataset	Feature Engineering, Deep Learning Techniques	
[63]	CERT v4.2, TWO's Dataset	Hidden Markov method	
[65]	TWO's Dataset	Long Short-Term Memory (LSTM)	
[70]	TWO's Dataset	Sysmon Parser	

2.4.6 Research Gaps and Challenges

From the literature, following research gaps have been identified:

- Insider Threat detection process has major class imbalance issue where the insider threat (red activities) ratio is much less as compared to normal (green) class. Class imbalance created by anomaly-based and un-supervised outlier approaches, is a challenge which needs to be addressed. Handling such problems requires a comprehensive framework encompassing ensemble learning techniques.
- The testing is performed on small portions of the CERT datasets and not the complete data set is analysed which results in low classification percentage.
- Lack of feature engineering techniques for assessing cyber defense solutions to

achieve higher classification rate. In many cases, the data collected does not accompany sufficient background information that is necessary for feature extraction. Most of the existing analysis methods have solely used timestamps or log key in their training process, which limit the ability of Machine Learning models.

- There is a dearth of real world datasets for research and analysis, whereas the available datasets are unlabeled.
- The log data available is unstructured and log files are recorded in diverse formats. Through pre-processing techniques unstructured data needs to be normalized and data redundancies be reduced. There are limited models applying pre-processing techniques to address the data normalization issue in Insider Threats datasets.
- Most studies have used a split ratio of 85:15 for training and testing data sets only; whereas the portion of validation test bed is missing.
- The number of insider class detected is not mentioned in the related research work and the performance metrics used are neither inclusive nor suitable for assessment of such problems. Recall perimeter and Confusion Matrices for the applied models is mostly missing in many studies.
- Problem of increasing the expressiveness of selected trained model to be way high and too specific though the model ends up accommodating stochastic behavior for training data perfectly but is unable to generalize unseen (test) data. Hence a very common Machine Learning problem is faced i.e Over Fitting Challenge especially in smaller data sets.

2.4.7 Problem Statement

Insider Threat activities pose a severe challenge to the repute, business secrets and well-being of the organizations; which are financially depleting to guard against such events. Signature based cyber security solutions are unlikely to deliver the requisite performance for new and multiple attack vectors. Whereas, Cyber analysts are finding it increasingly difficult to effectively monitor current levels of data volume, velocity and huge flux of threat dynamics. Strong trend has been observed toward anomaly-based and unsupervised outlier approaches, which can be attributed to class imbalance in datasets and fear of zero-day malicious attacks. However, it is believed that an effective and robust

Insider Threat defense program should contain a combination of several independent solutions. Firstly, strong and fool proof procedures including prevention and mitigation techniques be implemented. Secondly, misuse based detection be segregated and third line of defence which is Machine Learning techniques encompassing anomaly-based and unsupervised outlier detection should be deployed.

Existing work is mainly focused on (third line of defence) applying multiple Machine Learning techniques for big data analysis to handle the challenge of class imbalance, address over fitting issues and achieve higher rate of classification. This work requires feature engineering techniques and data parsing process which is a complex and time-consuming activity.

Therefore, there is a need to develop an Intelligent framework encompassing multitude of parameters which can proactively identify behavioral anomalies, implement outlier detection for likely Insider Threats in a given environment. The proposed framework should incorporate the pre-processing and feature engineering techniques and should be able to carryout predictive analysis of diverse insider attacks through ensemble learners. These analysis are performed on the basis of multiple patterns of activities that are not typically identified by conventional detection tools.

2.4.8 Research Objectives

This research work focuses on the development of comprehensive framework for Insider Threat detection by using multiple data sets involving user information, computer handling activities, email logs and psychometric data. To handle the issues and challenges identified; we have proposed different hybrid approaches and ensemble learners for the identification of Insider Threats through optimally accurate and efficient way. Main objectives of the research work are:

- To identify and address the class imbalance issue by combination of efficacious pre-processing techniques and ensemble learners
- Perform outlier detection and anomalies of diverse attacks for proactively identifying and effectively monitoring Insider Threats through incorporating supervised learning algorithms
- Apply multitude of data parsing and feature engineering techniques to handle un-

structured data and identify the most effective feature vectors by considering all possible combinations and then ranking them based on the accuracy

- Comparative Performance Analysis of selected (combinations) feature vectors and Machine Learning techniques against the accuracy matrix
- Apply effective Machine Learning (semi supervised) techniques for categorization of un-labelled traitor-based dataset and handling over fitting issues
- To accomplish an Intelligent Cyber Framework based on effective Machine Learning algorithms encompassing preprocessing and feature engineering techniques, for achieving higher classification (True Positive) with effective reduction in False Negative rate
- Acquire the authentic multiple datasets and process these in structured formats. Train, validate and test the proposed Cyber Framework on acquired data sets from various data sources as benchmarks

Chapter 3

S2M: Supervised Stacked Model for Insider Threat Detection

This chapter includes the exploration of the concept; that is, use of ensembles to handle class imbalance in the insider threat detection domain. For this purpose, experiments have been performed using ensemble practices comprising Bagging, Boosting, Stacking and Random Forest. The remaining dataset was pre-processed by transformation of attributes into numerical form and after that low variance filter was applied for feature selection after testing multiple data filtering techniques.

For experimentation CERT 4.2 is used for the training and testing purpose, which consists of normal and malicious activities of 1000 users recorded during the year 2010 to 2011. The dataset is publicly available on the following link for researchers [97]. The dataset include log on, log off, device connected or disconnected, visitation of a website, psychometric data, emails, file open or close events, organizational structure and user information. All events are present in separate csv files. For emails and files, sentiments were extracted from the content and classified as positive or negative activities. The psychometric data was included as five (OCEAN) individual attributes.

Moreover, this work proposes a stacked model trained and tested on labeled dataset thus leading to following contributions: -

1. A complete framework named S2M is designed and developed by combining pre-processing techniques and stacked ensemble learners to tackle class imbalance in large datasets.

2. We employ numerous pre-processing techniques such as Low Variance Filter (LVF), Correlation Filter to transform the datasets for analysis.

3. Proficiency of ML technique- Ensemble Learners are analyzed with sundry base learners including Decision Tree (DT), Gradient Boosting(GB) and Random Forest (RF) and the product is pushed through selected meta learner to achieve the optimized results.

4. Comprehensive results of the proposed model are presented with different performance matrices including Confusion Matrix.

5. Evaluated on publicly available dataset CERT 4.2; the proposed model showed the ability to generalize on a large dataset.

3.1 Methodology

In this section, the details of the proposed ensemble framework are explained. The activities (normal or malicious) are appended together and later pre-processed. Then the pre-processed dataset is fed to an ensemble learner. The learners are trained on 60% of the CERT dataset which are first evaluated on 20% of validation set and the best model is tested on 20% test set. Below we give an overview of the CERT 4.2 dataset and subsequently explain the steps involved in the prediction of insider activities.

3.1.1 System Overview

The proposed framework named as Supervised Stacked Model (S2M) for insider threat detection is demonstrated in Figure 3.1. The framework process is as follows:

1. **Data Collection:** Dataset is obtained and remodeled in standardized formats from various sources. Sources include:

    - User activities like logon details, device logs, emails receipts, file logs and http URLs
    - Profile information and user behavior
    - Organizational structure

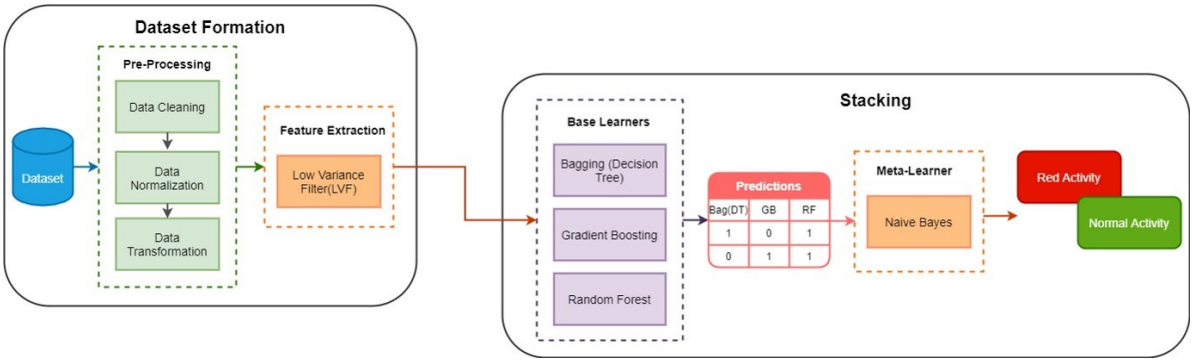


Figure 3.1: S2M: Insider Threat Detection Framework

2. **Data Pre-Processing:**
    - Data cleaning and normalization performed using Correlation Filter
    - Data Transformation from nominal to numerical values
    - Feature extraction using LVF
3. **Stacking** of multiple learners
4. Results are formulated into different performance metrics

The majority of machine learning based systems previously developed usually assume that training sets are well-balanced, but insider threat dataset are highly imbalance in nature because of rare occurrence of malicious activities. This issue cause hindrance in performance of algorithms such as DT. Our model named as Supervised Stacked Model is designed and developed to mainly address class imbalance. The model is a combination of pre-processing techniques and a stacked meta model with Naive Bayes (NB) base learner at level-1 and a combination of Bagged Decision Tree (Bag(DT)), Gradient Boosting (GB) and Random Forest (RF) at level-0. Pre-processing steps such as data cleaning, transformation and feature extraction prepare the dataset for training and testing. Following these steps, dataset is trained and tested on a stack of learners.

3.1.2 Dataset

The Computer Emergency Response Team (CERT) 4.2 dataset [97] has normal and malicious activities of 1000 users recorded for the year 2010 to 2011. The activities recorded include log on, log off, device connected or disconnected, visitation of a web-site, psychometric data, emails, file open or close events, organizational structure and

user information. These events are present in separate csv files. For emails and files, sentiments were extracted from the content and classified as positive or negative activities. The psychometric data was included as five (OCEAN) individual attributes. Then for experimentation, the dataset was next prepared by appending all these separate files into a single activities dataset. 9 types of activities were appended which include Log on/off , Device connect/disconnect , Site Visited, Email Sentiment Positive/ Negative. File Sentiment Positive/ Negative. The final prepared CERT 4.2 dataset had 31M normal samples and 6876 insider activities with 15 attributes (O, C, E, A, N, New_Role, seconds, month, year, date_of_month, activity_date, New Activity, URL Length, URL_TLD, class). The prepared dataset was sorted on the basis of timestamp and subsequently analyzed. The Figure 3.2 show that almost 30% of the data has no red (insider) activities. In order to handle this imbalance, the dataset was divided in three chunks (Chunk 1: 1-35, Chunk 2: 30-70, Chunk 3: 65-100). Table 3.1 shows the details of the chunk (normal and red activities/users).

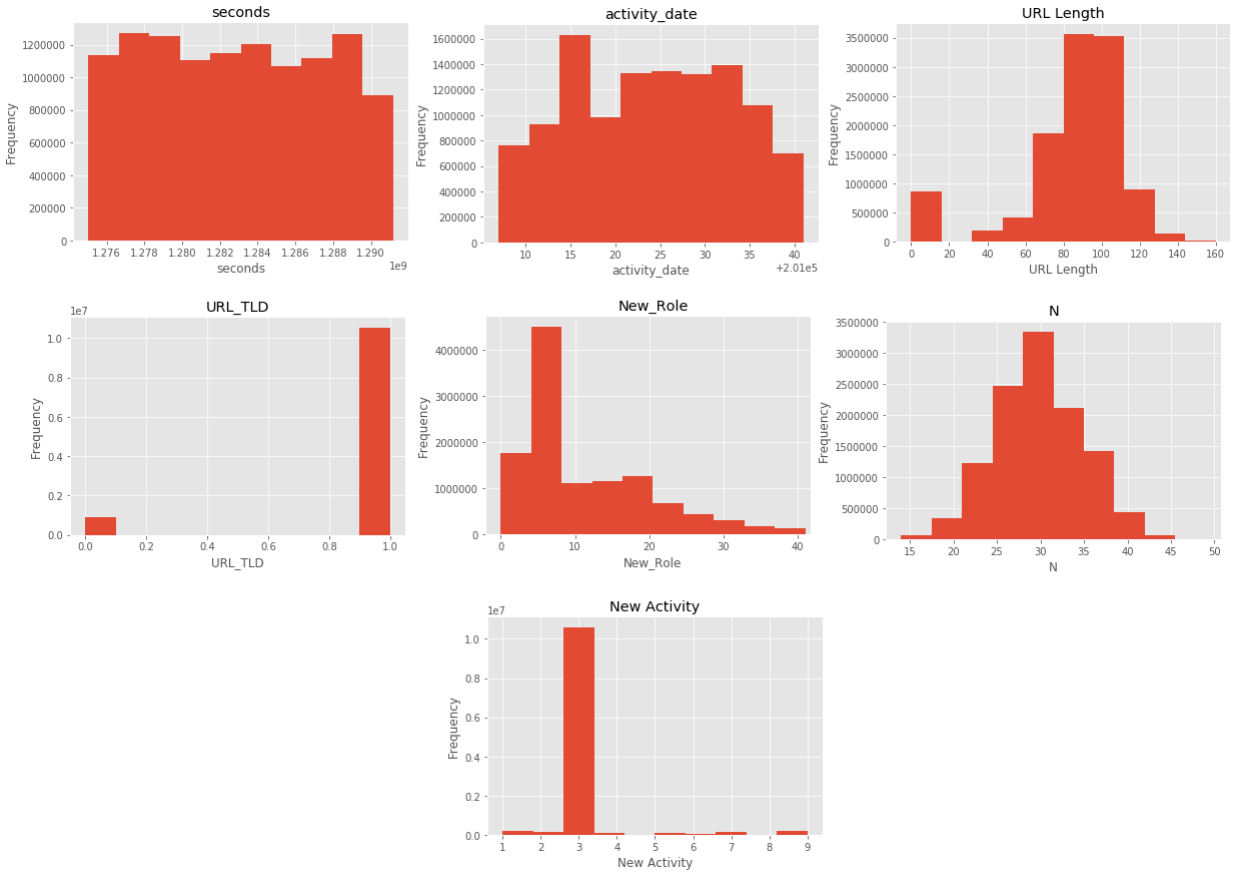


Figure 3.2: Selected Attributes from Chunk 2

Table 3.1: Activities count in Chunk 1: 1-35, Chunk 2: 30-70, Chunk 3: 65-100

Month	Chunk 1		Chunk 2		Chunk 3	
	Normal	**Red**	**Normal**	**Red**	**Normal**	**Red**
Jan-10	2036803					
Feb-10	2013162					
Mar-10	2294041					
Apr-10	2092948					
May-10	1933328		58981			
Jun-10			2151041	254		
Jul-10			1929199	979		
Aug-10			1943188	1110		
Sep-10			1835610	880		
Oct-10			1813564	720		
Nov-10			1712188	674	1545335	605
Dec-10			13495	7	1688531	517
Jan-11					1732527	455
Feb-11					1628677	672
Mar-11					1846681	497
Apr-11					1601439	198
May-11					870022	
Total	**10,370,282**		**11457266**	**4624**	**10913212**	**2873**
User	**1000**		**917**	**54**	**917**	**31**

Class 1 is labeled as malicious while class 0 is normal. Experiments were performed on Chunk 2 as it had the most class 1 samples. Later the final ensemble model was tested on Chunk 3 and thereafter on the whole dataset. Figure 3.2 shows the selected attributes for Chunk 2. In the Figure, we can see that most of the Class 1 samples lie in Chunk 2.

3.1.3 Data Preprocessing

3.1.3.1 Data Cleaning and Normalization

To prepare the dataset for training purposes, first step is data cleaning in which null values and outliers are removed. Following this, normalization is performed using Correlation Filter [98] to cut the sharp edges. Equation 3.1.1 explains the working of correlation filter.

$$G[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u,v]F[i+u, j+v] \tag{3.1.1}$$

This is called cross correlation, denoted as $G = H \otimes F$ The correlation coefficient has a range of values from -1 to 1.

1. A value that is closer to 0 indicates a lesser correlation, whereas a value that is exactly 0 indicates no correlation.
2. A number closer to one indicates positive correlation between features.
3. A number closer to one indicates negative correlation between features.

Figure 3.3 represents heat map of features. with features as row and column headers and feature versus itself on the diagonal. The horizontal and vertical axis shows the features name; the yellow color represents feature with correlation coefficient = 1 while the blue color shows value 0. While Figure 3.4 represents the concentration of values in numeric form, we removed the features where correlation between a pair of variables is between the 0.5-0.6.

3.1.3.2 Data Transformation

Dataset was preprocessed by transformation of attributes into numerical form. User Roles and the 9 activities were assigned numeric values according to a defined dictionary. The timestamp was transformed into date, month, year and seconds while also including a new derived variable that was a combination of the date, month and year. Further, two more derived variables were extracted from the website URLs: (1) URL length and (2) URL top level domain (TLD). The data transformation is summarized as follows:

1. Role is transformed by making a dictionary that is 'IT Admin' is assigned 0 and 'Technician' is assigned 3.

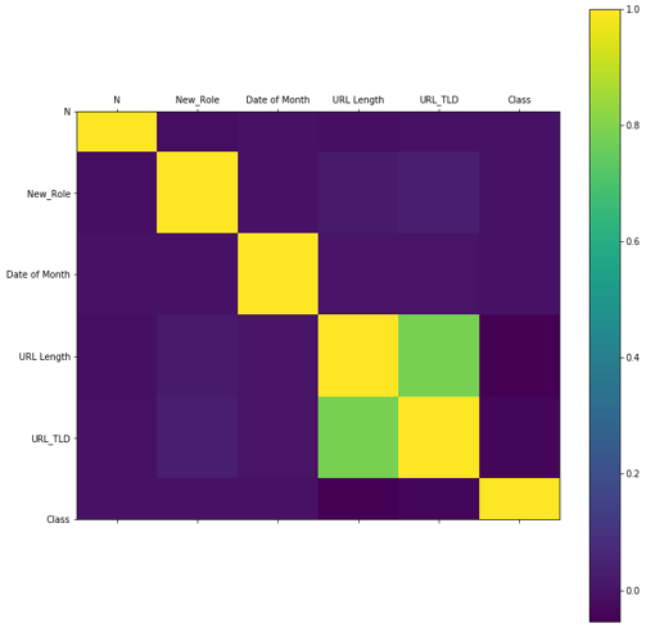


Figure 3.3: Heat Map of Dataset After Performing Correlation Filter

2. Date time is used to create 5 new features (seconds, year, month, date of month and hybrid activity date).

3. URLs are handled by getting 2 new features the length and the top level domain (TLD) of the URL.

4. Activities are mapped to integers by making a dictionary: Logon is assigned 0, Visited URL is assigned 3.

Normally, to handle class imbalance techniques like resampling techniques are employed. Synthetic Minority Over-sampling Technique (SMOTE) [99] was used in our experiments for this purpose. However, for the CERT datasets such techniques did not produce good results because SMOTE is not very practical for high dimensional data as it does not consider neighboring examples from other classes, in turn introducing additional noise.

3.1.4 Feature Extraction

The final step of data preprocessing is to reduce the number of features in large dataset. We followed the Principle Component Analysis (PCA) [100] technique for dimensionality reduction by opting component=4. New four features were constructed as linear combination of initial features. Although maximum variance among features was covered, but still important information was lost. To avoid information loss, Low Variance Filter

	New Activity	O	C	E	A	N	New_Role	Seconds	Year	Month	Date of Month	Activity Date	URL Length	URL_TLD	Class
New Activity	1.000000	0.001740	0.002487	0.001188	-0.006025	0.007054	-0.002118	-0.176690	-0.102265	-0.092952	-0.003200	-0.112460	-0.411676	-0.522520	0.013377
O	0.001740	1.000000	0.334622	-0.092350	0.078358	-0.047525	-0.030615	-0.021628	-0.013039	-0.010565	-0.000768	-0.014310	0.002712	0.005754	0.005087
C	0.002487	0.334622	1.000000	-0.102940	0.143396	-0.032994	0.027664	-0.024410	-0.019145	-0.005345	0.001999	-0.019549	0.013318	0.013973	0.007127
E	0.001188	-0.092350	-0.102940	1.000000	-0.168617	-0.003648	-0.214396	-0.019263	-0.015702	-0.003609	0.003964	-0.015490	-0.007546	-0.015259	-0.008174
A	-0.006025	0.078358	0.143396	-0.168617	1.000000	0.049467	-0.087662	0.014550	0.014216	-0.000745	-0.004972	0.013415	-0.010424	0.002184	-0.002855
N	0.007054	-0.047525	-0.032994	-0.003648	0.049467	1.000000	-0.075119	0.004208	0.001824	0.003073	0.000832	0.002279	-0.020352	-0.011578	-0.003956
New_Role	-0.002118	-0.030615	0.027664	-0.214396	-0.087662	-0.075119	1.000000	0.026489	0.019265	0.007771	0.002519	0.020801	0.042353	0.047953	-0.006727
Seconds	-0.176690	-0.021628	-0.024410	-0.019263	0.014550	0.004208	0.026489	1.000000	0.770101	0.236420	0.005631	0.805864	0.070763	0.115183	0.015855
Year	-0.102265	-0.013039	-0.019145	-0.015702	0.014216	0.001824	0.019265	0.770101	1.000000	-0.435404	-0.044359	0.976193	0.039669	0.067528	0.001311
Month	-0.092952	-0.010565	-0.005345	-0.003609	-0.000745	0.003073	0.007771	0.236420	-0.435404	1.000000	-0.008978	-0.365892	0.039240	0.059315	0.020415
Date of Month	-0.003200	-0.000768	0.001999	0.003964	-0.004972	0.000832	0.002519	0.005631	-0.044359	-0.008978	1.000000	0.161038	0.000460	0.001690	-0.000893
Activity Date	-0.112460	-0.014310	-0.019549	-0.015490	0.013415	0.002279	0.020801	0.805864	0.976193	-0.365892	0.161038	1.000000	0.043717	0.074006	0.002790
URL Length	-0.411676	0.002712	0.013318	-0.007546	-0.010424	-0.020352	0.042353	0.070763	0.039669	0.039240	0.000460	0.043717	1.000000	0.787332	-0.074103
URL_TLD	-0.522520	0.005754	0.013973	-0.015259	0.002184	-0.011578	0.047953	0.115183	0.067528	0.059315	0.001690	0.074006	0.787332	1.000000	-0.046465
Class	0.013377	0.005087	0.007127	-0.008174	-0.002855	-0.003956	-0.006727	0.015855	0.001311	0.020415	-0.000893	0.002790	-0.074103	-0.046465	1.000000

Figure 3.4: Correlation Matrix

(LVF) technique [101] was used for feature selection. Depending on the increase or decrease in variance, LVF selects a set of features and filters out the meaningful attributes in the dataset. The nine attributes (N, New_Role, seconds, month, date_of_month, activity_date, New Activity, URL Length, URL_TLD) that displayed a high variance (greater than 1) were included in the dataset. While the remaining four were discarded. Summary of features is displayed in Figure 3.5

O	1.108539e+02
C	1.192728e+02
E	1.228381e+02
A	1.219971e+02
N	2.490564e+01
Role	8.569982e+01
Activity	2.854457e+00
Year	0.000000e+00
Activity Date	7.762343e+01
Class	1.633339e-05
id	4.415983e+13
PC	1.408856e+06
User	1.162447e+06

Figure 3.5: Final Features Extracted from Low Variance Filter

3.1.5 Machine Learning Techniques

In this section we present the details of our proposed framework Supervised Stacked Model (S2M). Stacked generalization [102] in S2M is deployed to form an ensemble of various machine learning classifiers that altogether can be viewed as using classifiers based on their competency level with respect to their learning parameters set. Stacked generalization framework is categorized into main categories: Level-0 Classifiers (Base Classifiers) and Level-1 Classifier (Meta Classifier). Predictions of base classifiers are used for the training of meta- classifier. In general, more accurate predictions are obtained from stacked generalization framework that base classifiers.

One of the main point is to prepare the training dataset for level-0 classifiers by applying cross-validation technique. The original dataset is presented as $T = (X_n, y_n), n = 1, \ldots, N$, here y_n represents the target malicious instances, X_n is the features vectors of nth instances, randomly split data into $k = 10$ equal folds $T_1, T_2, T_3, ..., T_k$. Here T_k is defined as test and $T_{(-k)} = T - T_k$ as training dataset for k_th fold of a k-fold cross validation. Now for level-0 machine learning algorithms is denoted by $(C_1, C_2, \ldots, C_i), i = 3$, each C_i is trained on data $T(-k)$ and identified each malicious instance X in T_k. Prediction of the model C_i on X is represented as $pk^{(-i)}(X)$.

$$m_{kn} = pk^{(-i)}(X) \tag{3.1.2}$$

The data ensembles at the end of cross-validation of process of each C_i output, and presented as:

$$T_P = (y_n, m_{1n}, ...m_{in}), n = 1, 2, \ldots, N \tag{3.1.3}$$

T_P is the training set of level-1 model C_{meta}. To complete the training process, level-0 models $C_i(i = 1, 2, 3)$ are trained using original dataset T, and C_{meta} is trained by T_P.

Now we consider the prediction process, which uses the models $C_i(i = 1, 2, 3)$ in conjunction with C_{meta}. Given a new instance, models C_i produce a vector $(m_1, \ldots, m_i)$. This vector is input to the level-1 model C_{meta}, whose output is the final prediction result for that instance.

An abstract description of the proposed framework is described in Algorithm 1 which is composed of two steps i.e Training and Prediction. In the Training step, base classifiers that are included in an ensemble are trained on training portion of the dataset. After

that, in Prediction step, class label is assigned to the unlabeled test instance on the basis of our proposed supervised stacked model.

Algorithm 1 Supervised Stacked Model (S2M)

procedure SupervisedStackedModel(X, y)
    X = CorelationFilter(X)
    X = LowVarianceFilter(X)
    TrainX, Trainy, TestX, Testy = TestTrain(X,y)
    Step 1: Training of base classifiers
    while $TrainX \neq 0$ **do** ▷ Learn Level-0 classifiers
        dt = BaggingDT($TrainX$)
        gb = GB($TrainX$)
        rf = RF($TrainX$)
    end while
    nb = NB($TrainX$) ▷ Learn Level-1 classifiers
    Step 2 : Prediction
    for i = 1 to n **do**

$$y = \sum_{x=1}^{n} (Bag(DT) + GB + RF) + NB \tag{3.1.4}$$

    end for
    return y
end procedure

3.2 Experimental Environment setup

Python is a high level, object-oriented scripting language, compared to other languages. For our model construction, python libraries that are used are mentioned below:

3.2.1 Pandas

Pandas is a python bundle and a tool to analyze big data. An open source public library with high-performance, used as a data analysis tool.

3.2.2 NLTK

To work with human language, Natural language tool kit (NLTK) [103] act as a platform to build programs. It provides libraries for text processing tasks like tokenization, stemming, parsing and tagging. For dealing with natural language it is a widely used library.

3.2.3 Scikit-Learn

It exists as a tool for data excavating and to examine the findings. It is used for different purposes like classification, clustering and Regression etc [104].

3.2.4 Keras

Keras is a high-level neural networks API, carved in Python and proficient of running on top of TensorFlow, CNTK, or Theano. Its primary focus was on empowering fast research. Actuality able to leap after notion to outcome through the minimum conceivable interruption is crucial to achieve worthy study [104].

3.2.5 Matplotlib

It is a library of python. It is used to make graphs, histograms, pie charts, tables, scatter plots and bar charts etc. Matplotlib [105] python is used to make metrics.

3.3 Resampling Techniques

Resampling is a practice of carefully using a data sample to increase the accurateness and enumerate the vagueness of a populace constraint. Following are the two collective means of Resampling:

3.3.1 Cross Validation

Cross-Validation is an appraisal to check errors related with the ideal to assess its presentation. This is the utmost elementary line of attack. It merely comprises arbitrarily isolating the dataset in two portions: leading a training set and an additional validation

set or hold-out set. The model is fit on the training set and the fitted model is then used to make forecasts on the validation set.

- **Leave-One-Out Cross-Validation** A far more superior choice than the validation set method. As an alternative of piercing the entire dataset into two splits only one reflection is used for validation and the rest is used to fit the model [106].
- **K-Fold Cross-Validation** It includes arbitrarily separating the set of explanations into k folds of about equivalent proportions. The first fold is canned as a validation set and the rest model is to fit on the residual folds. The process is then repetitive k times, where a dissimilar assembly each period is preserved as the validation set [107].

3.3.2 Bootstrapping

Bootstrap [108] is an influential numerical tool used to enumerate the ambiguity of a given model. Nonetheless, the actual influence of bootstrap is that it could get functional to a wide variety of models where the inconsistency is hard to obtain or no output requisite.

- **Random Over-sampling**. This technique aims to stabilize class dissemination by arbitrarily aggregating minority class examples by redoing them.
- **SMOTE (Synthetic Minority Oversampling Technique)**. It combines new minority occurrences between existing minority cases. It arbitrarily picks up the minority class and computes the K-nearest neighbor for that specific point. Lastly, the artificial points are added among the neighbor's and the selected spot.
- **Random Under-Sampling**. It targets to equilibrium of class distribution by arbitrarily eradicating majority class samples. When occurrences of two different classes are very close to each other, we eradicate the occurrences of the majority class to escalate the spaces between the two classes. This helps in the classification procedure.
- **Cluster-based Over Sampling**. It is autonomously applied to both the class occurrences such as to categorize clusters in the datasets. All clusters are oversampled such that clusters of the same class have the same size.

3.4 Feature Selection Techniques

Feature selection is the procedure of picking the utmost significant types of a dataset. It is anticipated to decrease the figure of input variables to both decreases the computational fee of exhibiting and, in particular circumstances, to increase the performance of the model. Frequently in a high dimensional feature set, there persist several features which are dismissed, meaning these features are nil but extensions of the other vital features. These redundant features do not meritoriously subsidize to the model training as well. So, obviously, there is a need to abstract the most imperative and the most appropriate features for a dataset in mandate to get the optimum prognostic modelling performance.

Feature assortment can be prepared in various ways, nonetheless two focal categories are:

- Filter Method
- Wrapper Method

3.4.1 Filter Method

In filter technique [109] lone subset of the significant features is taken. The model is assembled after choosing the features. This is completed by means of association matrix and it is most frequently prepared using Pearson correlation in which correlation of self-governing variables with the output variable is calculated. Features having correlation of above 0.5 with the production variable are nominated.

The correlation coefficient has values ranging between -1 to 1:

- Value nearer to 0 implies feebler correlation (exact 0 implying no correlation)
- Value nearer to 1 implies robust positive correlation
- Value nearer to -1 implies sturdier negative correlation

3.4.2 Wrapper Method

In wrapper technique [110] machine learning algorithm is desirable and its performance is used as appraisal standards. Features are incorporated or detached based on the

performance of that machine learning algorithm.

Different types of wrapper methods are discussed below:

- **Backward Elimination** All the promising features are supplementary to the model at first and the performance of the model is squared, then iteratively eliminates the worst execution features one by one till the overall performance of the model comes in satisfactory range.
- **Recursive Feature Elimination (RFE)** This Technique works by recursively eradicating attributes and constructing a model on those attributes that linger. It uses accuracy metric to rank the feature according to their standing. The RFE technique takes the model to be used and the number of essential features as input. It then gives the standing of all the variables, 1 being most significant. It also gives its backing, true being a germane feature and False being irrelevant feature.

3.5 Training and Testing

For exercise and analysis we preferred the technique of 10-fold cross validation. As shown in Figure 3.6, it consists of 10 trials, every single while taking diverse sets for exercise and analysis from the input dataset. In this process:

- The input dataset is distributed into 10 equal subsets.
- From these 10 subsets, 9 are used for exercise and 1 is used for analysis.
- Procedure is recurring 10 times, each time taking dissimilar subset for testing.
- Concluding performance is appraised by captivating an average of outcomes.

3.6 Evaluation Measures

In direction to evaluate the performance of suggested methodology, we have selected some typical procedures that include precision, exactness, recollection and F-measure. Here are the mathematical formulations for these factors.

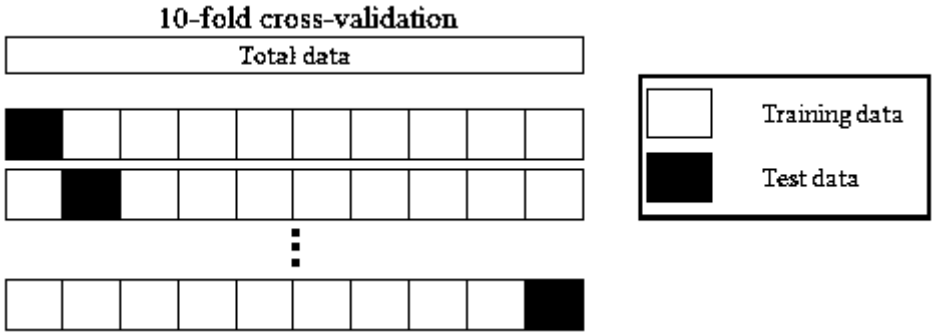


Figure 3.6: 10-Fold Cross Validation

3.6.1 Confusion Matrix

The results predicted by the classifiers are presented in a tabular form that splits the precise prediction of class from unfitting predictions. This is called confusion matrix [111]. It tells the correct and incorrect predictions. Other routine measures like accurateness, exactness, remembrance and F-measure can be calculated by means of this matrix. Confusion matrix is represented in below. The four cells of matrix show true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN).

TP = Amount of predictions that are properly classified by the classifier as confident.
TN = Amount of predictions that are properly classified by the classifier as undesirable.
FP = Amount of predictions that are wrongly classified by the classifier as confident.
FN = Amount of predictions that are wrongly classified by the classifier as undesirable.

		Prediction outcome		
		p	**n**	**total**
actual value	**p′**	True Positive	False Negative	P′
	n′	False Positive	True Negative	N′
	total	P	N	

3.6.2 Accuracy

Accuracy is the element of occurrences that are properly categorized separated by the aggregate amount of occurrences. It can be given as:

$$Accuracy = \frac{No.\ of\ Correctly\ Classified\ Instances}{Total\ No.\ of\ Instances} \times 100 \tag{3.6.1}$$

Confusion matrix can be used to find accuracy by using TP and TN that defines correctly classified instances and sum of all cells of confusion matrix that defines the total instances. It can be given as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.6.2}$$

3.6.3 Precision

Precision is the fraction of number of precise calculations by the complete predictions. It calculates the fraction of instances that are truly positive. In relations of likelihood, precision is the possibility that an instance is correctly classified. In terms of confusion matrix, it can be measured as:

$$Precision = \frac{TP}{TP + FP} \tag{3.6.3}$$

3.6.4 Recall

Recall is the degree of the segment of optimistic occurrences that were properly categorized. In terms of confusion matrix, it can be measured as:

$$Rec = \frac{TP}{TP + FN} \tag{3.6.4}$$

3.6.5 F-Measure

F-Measure is the vocal mean of accuracy and recall. It delivers a equilibrium amongst precision and recall and uses mutually to compute a performance degree. Its formulation is specified as:

$$F - Measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{3.6.5}$$

For measuring the classifier performance, correctness single-handily is not a suitable measure. E.g think through a dataset having a complete of 100 instances out of which

95 are undesirable and 5 are optimistic. If a classifier categorizes all the occurrences as undesirable, the correctness of the classifier will be 95%, although that no optimistic instance is correctly categorized. Hence, the other performance procedures overwhelmed this restraint by computing a segment to segment portion TP.

3.7 Results and Discussion

The prepared dataset was sorted on the basis of timestamp and analyzed. In order to handle this imbalance, the dataset was divided in three chunks (Chunk 1: 1-35, Chunk 2: 30-70, Chunk 3:65-100) presented in Figure3.7. Chunk 2 contains maximum red (insider) activities. Experiments were performed on Chunk 2 with different combinations of traditional and ensemble learners on the preprocessed subset. Figure 3.8 displays the normal and red activities distribution of chunks. Multiple performance metrics are used to assess the proposed model including Area under the Curve (AUC), Accuracy (Acc) and Recall (Rec) as the cost of false negative is high, given in Eqs. 5.4.1 and 5.4.2.

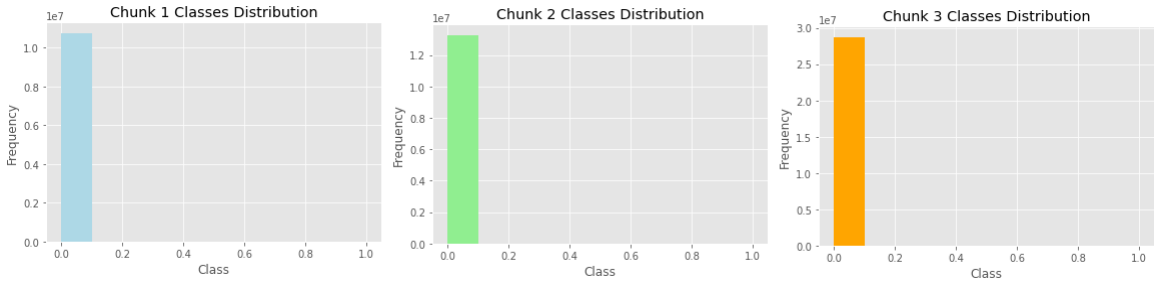


Figure 3.7: Dataset Representation in Chunks

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.7.1}$$

$$Rec = \frac{TP}{TP + FN} \tag{3.7.2}$$

Here,

TP = True Positive (Correct Predicted Red Activities)

TN = True Negative (Correct Predicted Normal Activities)

FP = False Positive (Normal Activities Predicted as Red)

FN = False Negative (Red Activities Predicted as Normal)

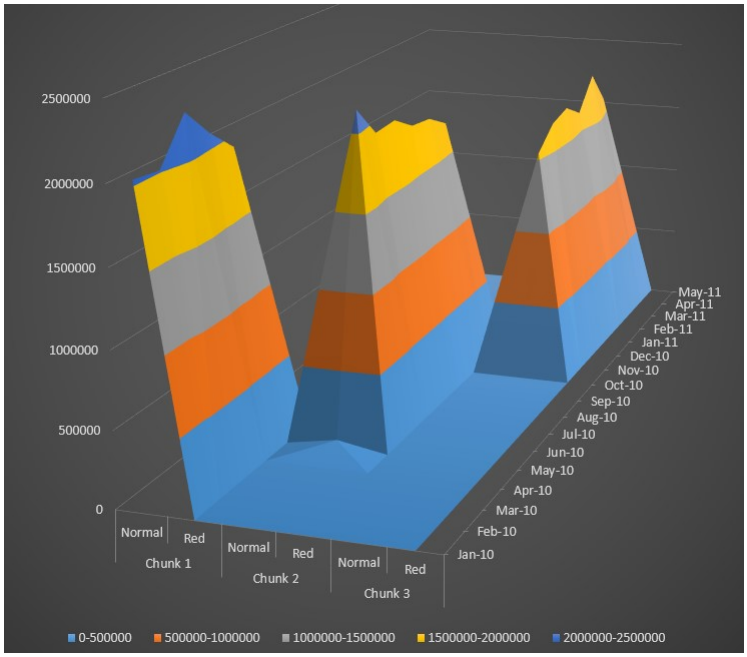


Figure 3.8: Chunk1, 2 and 3

To begin with, we performed experiments using single classifiers on chunk 2. Namely, Multilayer Perceptron (MLP), K Nearest Neighbor (KNN), Naive Bayes (NB), Decision Tree (DT), Random Forest(RF) and Gradient Boosting (GB). Single classifiers displayed highest performance with AUC of 0.83 and 0.779 with DT and NB respectively. Whereas, detection rate of DT touched only 65% of the red class (malicious) samples. Hence, to attain the best results, experiments were re-conducted on the different combinations of learners to build a stack. Stacking technique was chosen to elevate the results of single learners used in stacking. Table 3.2 exhibits the results of our proposed (S2M) Model. Confusion Matrix also displays the correct number of class 1 samples predicted with the Proposed Model.

On Chunk 2, the final stacked bagged ensemble technique gave an AUC of 0.972 with 832 out of 912 class 1 samples predicted correctly in the test set.On Chunk 3, an AUC of 0.990 with 526 out of 571 class 1 samples predicted in the test set. While 1254 class red samples were correctly predicted out of 1413 samples on the whole CERT 4.2 dataset, giving an AUC of 0.982, which can be seen in Table 3.2.

The AUC and ROC for chunk2 (having maximum red activities) was observed for only class 1 samples which can be seen in Figure 3.9 and subsequently AUC and ROC details for whole dataset are shown in Figure 3.10.

Table 3.2: Results of S2M on Test Set

Data	Acc (%)	Rec (%)	Confusion Matrix (%)			AUC
Chunk2	99.68	91.23		***Red***	***Normal***	0.972
			Red	***832 (TP)***	***80 (FN)***	
			Normal	***7087 (FP)***	***2284379 (TN)***	
Chunk 3	99.73	92.11		***Red***	***Normal***	0.990
			Red	***526 (TP)***	***45(FN)***	
			Normal	***5791 (FP)***	***2176855 (TN)***	
All	99.88	88.74		***Red***	***Normal***	0.982
			Red	***1254 (TP)***	***159 (FN)***	
			Normal	***7227 (FP)***	***6229124 (TN)***	

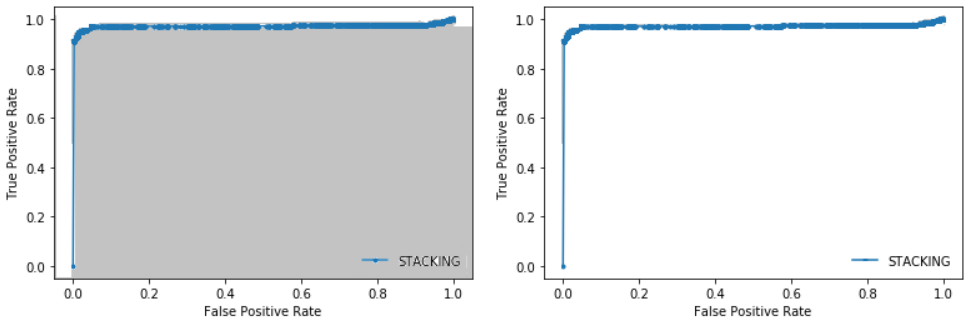


Figure 3.9: AUC and ROC of S2M for Chunk2

The TDR is around 90% in each dataset which is the ultimate goal for the model. The FAR is around 7% to 8% which is appropriate for the CERT datasets, as it has class imbalance issues. Here, the main goal of the model is to predict the red class activities and therefore the False Positives in this case can be ignored. Considering the confusion matrix alone of our (proposed framework) S2M, we are able to achieve better results than cited literature. Results are compared in Table 3.3 below.

S2M is a comprehensive framework comprises of the pre-processing techniques and

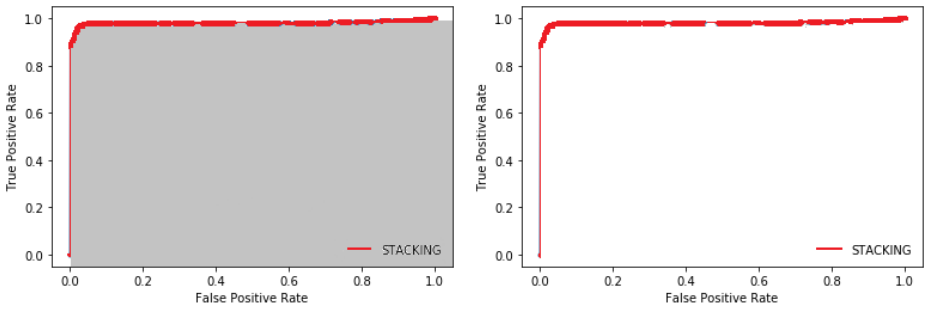


Figure 3.10: AUC and ROC of S2M for entire dataset

stacked learners. We conducted the experiments on unprocessed/ unstructured dataset initially. Results on raw dataset highlighted the class imbalance issue with AUC 0.59 and ACC 61%. To tackle these issues, we processed the dataset using S2M pre-processing techniques. To evaluate the effect of each learner in stacked model, we performed the ablation experiments. Results (Table 3.4) show that combination of two learners give maximum accuracy of 97% and AUC 0.975. Hence, we combined three learners to build a stack and passed through the meta learner as demonstrated in our threat detection framework Figure 3.1.

3.8 Research Contributions

The acme of this chapter is to use a hybrid ensemble model for the detection of an Insider Threat. For that instance, experiments were performed on a huge CERT 4.2 dataset consisting of more than 31 Million records of 1000 users with 15 Cols (feature parameters). For better simulation environment the data was fragmented in three major segments which is training, validation and test of 60%, 20% and 20% records respectively. From previous researches, number of models can be found for analyzing and predicting Insider Threats and also to deal with class imbalance. Among those models, few perform better results than others. The importance of predicting and identifying insider threats has directed to progress of a fresh model which needs to be more accurate. For this reason, a well-defined framework is proposed which provides better results than existing models. Moreover, models involving deep learning are already in use, but they involve individual models such as CNN. Better models showing improved performance is still the need and this research has tried to address the class imbalance problem by proposing a hybrid ensemble Supervised Stacked Model (S2M).

Table 3.3: Comparison with Literature on CERT Datasets

Model/Classifiers	**Detection Rate (%)**
Random Forest, Linear Regression, Artificial Neural Network [64]	75
HMM, Damerau Levenshtein (DL) Distance, Jaccard, and Cosine Distance [63]	80
LSTM [65]	90
Ensemble of LSTM, CNN [68]	90
Convolutional Neural Network (CNN), Long Short Term Memory (LSTM) [62]	94.49
Gaussian density estimation, Parzen window density estimation, PCA, K-Means [67]	94.79
PCA, LSTM, Recurrent Neural Network (RNN) [60]	95
Recurrent Neural Network (RNN) [112]	96.6
Majority Voting Ensembles [113]	97
Proposed Framework (S2M)	**98.2**

Consequently, experiments show that S2M can achieve the best classification accuracy for red activities as well as normal activities for Whole Dataset with 99% Accuracy and 0.982 AUC. Combination of feature engineering techniques like low variance filter (LVF) and ensembles to produce a nested learner S2M that correctly classifies 88.74% of the

Table 3.4: Ablation Experiments on Processed Dataset

Model/Classifiers	**Acc (%)**	**AUC**	**Rec (%)**
Bag(DT) + GB	96	0.962	85.31
Bag(DT) + RF	96.7	0.97	86.2
GB + RF	97.2	0.975	87.5
Proposed Framework (S2M)	99.88	0.982	88.74

insider samples and almost 100% of the normal activities. The publicly available CERT 4.2 dataset is used for experimentation with an AUC of 0.982 that outperformed the existing techniques applied on this dataset. Moreover, the proposed S2 model encompasses the complete framework which handles the class imbalance issue previously not addressed in literature.

Chapter 4

A Dynamic Weighted-Voting Ensemble Framework for Insider Threat Detection

This chapter presents a ML-based framework for detection of Insider Threat activity.The focus is on application of a weighted-voting approach along with pre-processing techniques for identifying malicious insider activities from normal user behaviors. Using various data sources, such as network logs and user telemetry, the scheme trains an ensemble of ML classifiers and infers threats via adaptive weighting. The research addresses the numerous traditional data issues and model engineering challenges, such as clean-up, suitable featurization, and class imbalance. This work is focused on empirical results, including an end-to-end implementation and quantitative results of applying its techniques on popular CERT datasets. Thereby, the research aims to make the following contributions:

1. Employ different pre-processing techniques such as Low Variance Filter (LVF), Correlation Filter to transform datasets for analysis.

2. Proficiency of ML technique -Ensemble Learners are analyzed with multiple base (Naive Bayes (NB), Gradient Boosting(GB) and Random Forest (RF)) learners.

3. In a weighted voting approach; weights of base learners are calculated dynamically, based on their respective competence level.

4. Comprehensive results of the proposed model are presented with different performance matrices such as Rec, Acc, AUC, and Confusion Matrix.

5. Evaluated on publicly available dataset CERT (4.2 and 6.2); the proposed model showed the ability to effectively generalize on any large dataset.

4.1 Methodology

The primary motivation of this study is to evaluate the capability of machine learning techniques in an organizational network to identify insider threats. In this section, the framework for Insider Threat detection is presented in 4.1.1. The framework is designed to be flexible and readily expandable for all types of organizational environments, data collection and analytical techniques. Subsequently, 4.1.2 explains data collection and processing steps, where features are extracted and data granularity are described. Finally, 4.1.3 presents the description of our proposed framework.

4.1.1 System Overview

The proposed framework for Insider Threat detection is demonstrated in Figure 4.1. The framework process is as follows:

1. Data Collection: Data is obtained and stored in standardized formats from various sources. Sources include:

    - User Activities like Logon details, device logs, emails receipts, file logs and web uploads/downloads
    - User Behavior and profile information
    - Organizational Hierarchy

2. Data Pre-Processing: Feature vectors containing user activities and profile information are constructed from aggregated data which is transformed from nominal to numerical form

3. Machine learning techniques are applied on pre-processed dataset

4. Results are formulated into numerous performance metrics

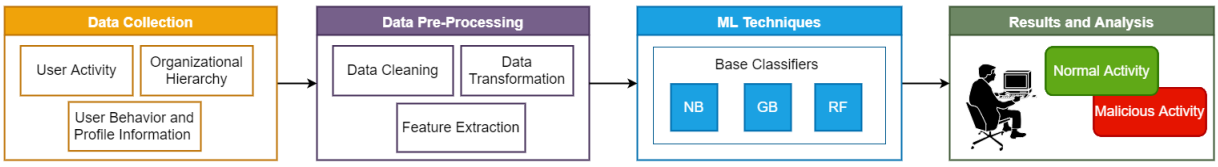


Figure 4.1: System Overview

The framework is configured to work with the participation of security analysts in many phases, precisely in early detection where the signs of malicious and abnormal user activities are being monitored.

In the research work, we have used CERT dataset, which is a de facto standard dataset in Insider Threat detection domain. Specifically, we are interested in analyzing machine learning algorithms that are trained with a small amount of ground reality for detection of an anonymous malicious insider. For this purpose, ensemble machine learning technique along with dynamic weighted voting approach is employed to learn from the information gained on the malicious and normal activities of the recipient. Then we have explored how our proposed framework helps identifying insider threats (malicious activities). The benefit of using ensemble technique with dynamic weighted voting approach is that, each classifier is assigned weights on its classification accuracy. These assigned weights differentiate the classifiers with "better accuracy" from the one with "average accuracy" which will have better impact on final prediction.

Our framework will help to analyze the malicious and normal user behaviors by dynamically assigning weights to base classifiers and aggregating results using weighted voting approach. In previous research work, weights are assigned either by using static technique or dynamic technique, but in our proposed framework the combination of dynamic weight assigning technique with voting approach help us to achieve good results. The main reason of using dynamic weighted voting approach is that each classifier behaves differently on different dataset, hence accuracy and AUC varies accordingly. We assign weights based on classification accuracy of classifier on respective dataset. This approach solves the limitations of static weighted-voting approach. Our model not only tackles the class imbalance but also have ability to generalize on unseen/real world dataset. The role of user in an organization will have a great effect on the type of actions performed, in both malicious and normal activities. In certain instances, user behavior can differ over time. Many user activities need to be considered in order to process a malicious

activity. Thus, high detection rate of malicious activity does not mean that all malicious activities are detected. In fact, small false positive rate also require attention as normal activity can also be malicious. Conclusively, we infer that outcomes that highlight malicious users rather than activities are more important indicator of systems performance. Moreover, multiple scenarios are used for the assessment of Insider Threat detection.

4.1.2 Data Collection

Data collection is the most important step for Insider Threat detection as it helps in successful implementation of machine learning techniques and assists in making correct predictions. Data is collected from various sources and is divided into three main categories:

1. Users' activity data
2. Users' behavior and profile information
3. Organizational hierarchy

Data of the first category comes from network details, device logs, emails' receipts, file logs and web uploads/downloads. There are the real-time data sources that are mostly used in order to be collected and processed in a timely manner to quickly detect and respond to malicious and/or anomaly conduct. The second category of data represents user profile information and user behavior that involves their personality and psychometric traits (O: Openness, C: conscientiousness, E: Extraversion, A: Agreeableness, N: Neuroticism). While the third category of data undertakes user role in an organization along with organizational hierarchy. Data collected from all the sources is then aggregated on the basis of user id and passed to the pre-processing step.

4.1.3 Data Pre-Processing

The aggregated data is in raw form which need to be processed before being fed to the machine learning classifiers. The pre-processing is further divided into three steps i-e

- Data Cleaning
- Data Transformation

- Feature Extraction

4.1.3.1 Data Cleaning

Dataset set is analyzed for uniformity and any discrepancy that require fixation. Rows containing null values are removed.

4.1.3.2 Data Transformation

Data is transformed from nominal to numerical form.

1. Date time is used to create 5 new features (seconds, year, month, date of month and hybrid activity date)
2. Ids, PC, Role, User and Activity are mapped to integers by making a dictionary

4.1.3.3 Feature Extraction

The transformed data contain large number of features that need to be reduced to get the important features. Following feature extraction techniques were used:

1. **Extra Trees Classifier** [114] [115] is an ensemble learning technique that combines the results of numerous de-correlated decision trees to predict final output. For feature selection, Gini index of each feature is computed that is regarded as the importance of that feature. Equation 4.1.1 shows the formula for the calculation of Gini index. On the basis of importance all the features having value less than 0.01 are eliminated and others are selected. Figure 4.2 represents the result of Extra Trees Classifier.

$$Entropy(S) = \sum_{i=1}^{c}(-p_i \log_2(p_i)) \tag{4.1.1}$$

   c is measure of total variance across the c classes and p_i is the probability of an element which is being classified for a distinct class.

2. **Low Variance Filter** [116] computes the variance of each feature. Equation 4.1.2 shows the formula for the calculation of variance. If the variance of feature is low,

ID	0.03596675
PC	0.04711147
User	0.03818047
O	0.02983395
C	0.0179675
E	0.03371427
A	0.11805255
N	0.2651364
Activity	0.36892597
Seconds	0.00241978
Year	0.03721979
Month	0.00405455
Day	0.005054645
Activity Date	0.21111396
Class	0.10555277

Figure 4.2: Result of Extra Tree Classifier

it means that it is not changing much and can be eliminated. Figure 4.3 represents the result of low variance filter.

$$s^2 = \left(\frac{\sum_{i=1}^{n} x_i^2}{n} - \left(\frac{\sum_{i=1}^{n} x_i}{n} \right)^2 \right) \times \frac{n}{n-1} \tag{4.1.2}$$

n is the total number of instances, the variable x_i is the one on which variance is measured.

O	1.108539e+02
C	1.192728e+02
E	1.228381e+02
A	1.219971e+02
N	2.490564e+01
Role	8.569982e+01
Activity	2.854457e+00
Year	0.000000e+00
Activity Date	7.762343e+01
Class	1.633339e-05
id	4.415983e+13
PC	1.408856e+06
User	1.162447e+06

Figure 4.3: Low Variance Filter Results

After feature extraction, vectors of fixed length are created that summarizes user activities. Each vector contains user id, role, type of activity performed, activity date and pc information. Figure 4.4 represents the feature details of CERT dataset.

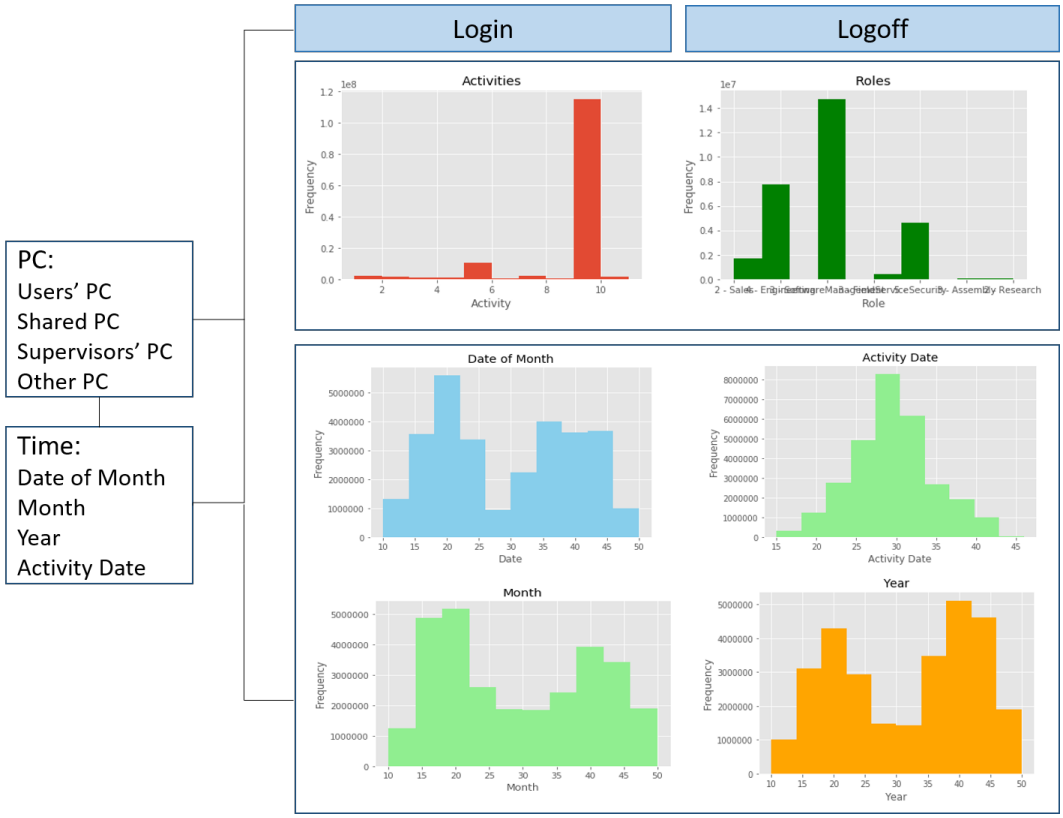


Figure 4.4: Dataset Representation in Chunks

4.1.4 Machine Learning Techniques

In this section we present the details of our proposed framework named as Dynamic Weighted-Voting Ensemble Learning Framework (DWvEn). Generally, the creation of an ensemble [117] is primarily concerned with two phases: Selection and Combination. The selection of competent classifiers based on the accuracy and diversity is the key point for the efficiency of an ensemble. While different techniques can be used to combine the predictions of individual classifiers.

By considering these, the proposed framework selects a set $C = (C_1, C_2, ...C_M)$ of M different classifiers and combine their predictions through weighted voting approach, It is worth noting that weighted voting [118] is a widely used strategy for combining predictions in pairwise classification, in which classifiers are not treated equally. Each classifier is assessed on evaluation set T and assigned a weight, usually proportional to its classification accuracy.

A dataset T with N classes is used to compute the competence of each classifier C_i, with $i = 1, 2, ...M$ and a $M \times N$ matrix W which is defined as follows:-

$$\mathbf{W} = \begin{bmatrix} w_{1,NB} & w_{1,GB} & w_{1,RF} \\ w_{2,NB} & w_{2,GB} & w_{2,RF} \\ w_{3,NB} & w_{3,GB} & w_{3,RF} \\ . & . & . \\ . & . & . \\ w_{N,NB} & w_{N,GB} & w_{N,RF} \end{bmatrix}. \tag{4.1.3}$$

where each element $w_{i,n}$ is defined by

$$\mathbf{w_{i,n}} = \frac{2g_n^{C_i}}{|T_n| + g_n^{C_i} + h_n^{C_i}} \tag{4.1.4}$$

where T_n is the total instances of the dataset belonging to the class n, $g(C_i)_n$ are the number of correct predictions of classifier C_i on T_n and $h(C_i)_n$ are the number of incorrect predictions of C_i that an instance belongs to class n. Clearly, each weight $w_{i,n}$ is the F1-score of classifier C_i for n class. The rationale behind this is to measure the efficiency of each classifier, relative to each class n of the evaluation set T.

Eventually, the prediction of test instance is calculated through equation 4.1.5. Where max function returns the maximum value and assigns a class label to test instance.

$$y = \sum_{x=1}^{n} (w_{i,n} \times (C_i(x) = n)) \tag{4.1.5}$$

An abstract description of the proposed framework is described in Algorithm 2 which is composed of three steps i.e Training, Evaluation and Prediction. In the Training step, base classifiers that are included in an ensemble are trained on the training dataset. After that, in the Evaluation step, the competency of each classifier is computed, and weights are assigned to each base classifier. Finally, in Prediction step, class label is assigned to the unlabeled test instance on the basis of our proposed weighted voting approach. An overview of the proposed framework is presented in Figure 4.5.

Algorithm 2 Dynamic Weighted Voting Approach

procedure Dynamic Weighted Voting(X, y)
  X = ExtraTreesClassifier(X)
  X = LowVarianceFilter(X)
  TrainX, Trainy, TestX, Testy = TestTrain(X,y)
  Step 1: Training of base classifiers
  while $TrainX \neq 0$ **do**
    nb = NB($TrainX$)
    gb = GB($TrainX$)
    rf = RF($TrainX$)
  end while
  Step 2 : Evaluation
  for i = 1 to m **do**
    Apply each base classifier on $TestX$
    for n = 1 to m **do**
      Calculate weight of each classifier for **n** class

$$\mathbf{w_{i,n}} = \frac{2g_n^{C_i}}{|T_n| + g_n^{C_i} + h_n^{C_i}} \tag{4.1.6}$$

    end for
  end for
  Step 3: Prediction
  Predict label of test instance z by using

$$z = \sum_{x=1}^{n} (w_{i,n} \times (C_i(x) = n)) \tag{4.1.7}$$

  return z
end procedure

4.2 Experimental Evaluation

In this section, we describe the evaluation results of our proposed framework on CERT dataset. Section 4.2.1 presents the details of dataset along with data processing. Section 4.2.2 introduces experimental settings and performance metrics. While section 4.2.3 presents the results of experiments.

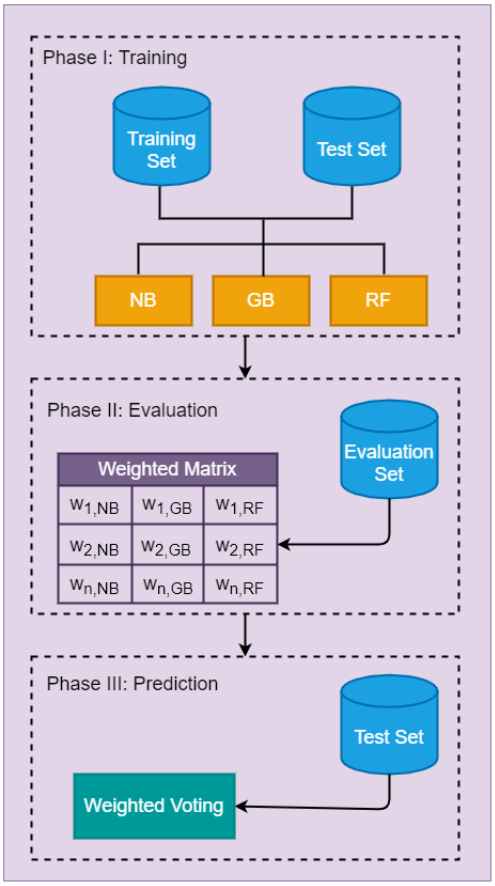


Figure 4.5: Framework Diagram

4.2.1 Dataset

The CERT 6.2 [119] dataset has normal and malicious activities of 4000 users recorded for the year 2010 to 2011. The activities recorded include log on, log off, device connected or disconnected, visitation of a website, psychometric data, emails, file open or close events, organizational structure and user information. These events are present in separate csv files. For emails and files, sentiments were extracted from the content and classified as positive or negative activities. The psychometric data was included as five (OCEAN) individual attributes. Then for experimentation, the dataset was next prepared by appending all these separate files into a single activities dataset. 11 types of activities were appended which include Log on/off, Device connect/disconnect, Web Upload/Download/Visit, Email Sentiment Positive/ Negative and File Sentiment Positive/ Negative. The final prepared CERT 6.2 dataset had 135M normal samples and 470 insider activities with 15 attributes. Analysis of dataset by time and malicious users

emails is demonstrated in Figure 4.6.

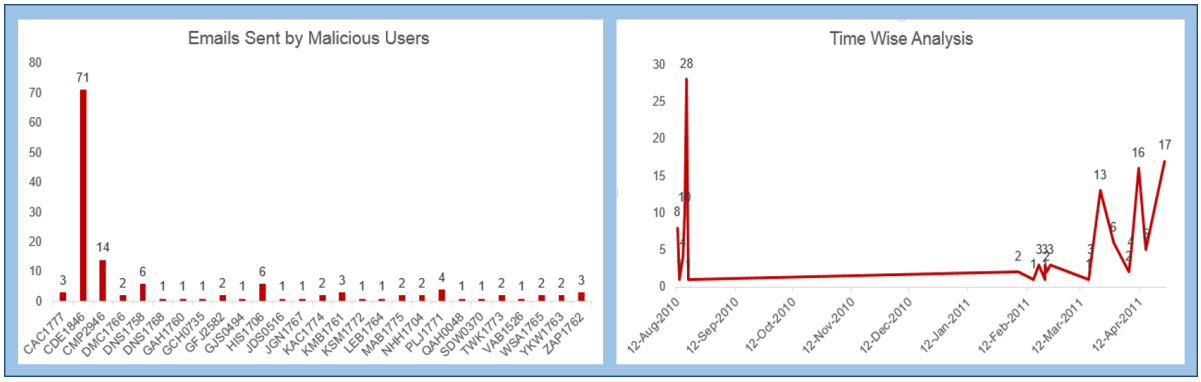


Figure 4.6: Dataset Analysis

4.2.2 Experiment Settings

Our objective in this work is to obtain a reasonable estimate of the performance of the proposed framework on organizational networks, based on scenarios characterized by constraints on the amount of data available for machine learning algorithms training. Specifically, labeled data for training detection systems is scarce in real-world environments.

The prepared dataset was sorted on the basis of timestamp and analyzed. In order to handle this imbalance, the dataset was divided in three chunks (Chunk 1: 1-35, Chunk 2: 30-70, Chunk 3:65-100) presented in Figure 4.7. Chunk 3 contains maximum red (insider) activities. Chunk 3 was further divided into month wise chunks as most of the red activities were found from February to April 2011. Data of month February, March and April was selected that contain 23M normal and 376 red activities. Insider information scenario wise is described in Table 4.1.

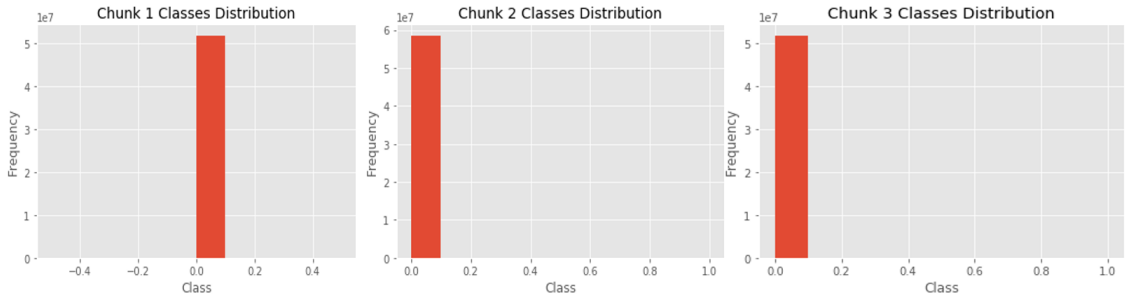


Figure 4.7: Dataset Representation in Chunks

Table 4.1: CERT Dataset Scenario Information

Scenarios	**No. of Users**	**No. of Activities**
1- Data Breach	1	22
2- Intellectual Property Theft	1	242
3- Forgery	25	68
4- Infiltration	1	134
5- Miscellaneous	1	4

The results of the evaluation are obtained from a series of experiments, where each setting is randomly replicated 10 times by a machine learning algorithm on a dataset. In the first set of experiments, we have compared the framework results with single classifiers. While in the second set of experiments, results of our framework are compared with other ensemble learning techniques. For each Insider Threat scenario detailed analysis is performed. Moreover, models trained on CERT 6.2 are also tested against other versions of CERT i.e. 4.2 insider data for exploring the performance of framework.

4.2.2.1 ML Training Configuration and Parameterization

In this research, Python 3.7 is used for data pre-processing and sklearn [120] library is used for the implementation of machine learning algorithms. The training data is cleaned and transformed. The algorithms are trained on binary data composed of two class i-e normal (positive) and malicious (negative).

Naive Bayes [121] assumes to perform best under default parameters. While for other two algorithms, experiments were performed with parameter settings and parameters with best results are selected. For Gradient Boosting [122] number of estimators are tuned from 50 to 100 and depth of each tree is from 5 to 10. While for Random Forest [123] number of features selected are from 3 to 10 and number of estimators ranges from

5 to 15.

4.2.2.2 Performance Metrics

In Insider Threat applications, Recall also called Detection Rate (DR) and false positive rate (FPR) are mostly used [124].

$$\textbf{Recall} = \frac{TP}{TP + FN} \tag{4.2.1}$$

$$\textbf{FPR} = \frac{FP}{FP + TN} \tag{4.2.2}$$

Where FP, FN, TP, TN are False Positive, False Negative, True Positive and True Negative respectively. TP represents correctly predicted malicious activities and TN represents correctly predicted normal activities. FP represents normal activities predicted as malicious and FN represents malicious activities predicted as normal.

Along with these, other performance metrics Precision, F1-Score, Area Under the Curve (AoC) and Region Under the curve (RoC) are also used for analyzing the performance of proposed framework.

$$\textbf{Precision} = \frac{TP}{TP + FP} \tag{4.2.3}$$

$$\textbf{F1} - \textbf{Score} = \frac{2}{Precision^{-}1 + Recall^{-}1} \tag{4.2.4}$$

4.2.3 Results by Single Classifiers

After preprocessing, machine learning algorithms are trained to classify the possible malicious activities. Classification algorithms namely, Gradient Boosting, Naive Bayes, Random Forest, Decision Tree, K Nearest Neighbour (KNN), Logistic Regression (LR), AdaBoost and Support Vector Machine (SVM) are trained and tested. Table 4.2 shows the results of single classifiers. Naive Bayes(NB) and Random Forest(RF) have higher accuracy than other classifiers. Result analysis shows that KNN, SVM, LR and AdaBoost showed an average AUC of 50% which is inadequate. While GB, NB, RF and

DT show better results. Based on individual accuracy, NB, RF and GB are chosen as base classifiers. Figure 4.8 shows the AUC, recall and F1-Score of single classifiers.

Table 4.2: Results on Test Set

Classifiers	**Acc (%)**	**Rec (%)**	**Confusion Matrix (%)**			**AUC**
Naive Bayes	74.56	75		***Malicious***	***Normal***	0.815
			Malicious	***62 (TP)***	***8 (FN)***	
			Normal	***936886 (FP)***	***2746236 (TN)***	
Random Forest	99.99	74		***Malicious***	***Normal***	0.728
			Malicious	***32 (TP)***	***38 (FN)***	
			Normal	***11 (FP)***	***3683111 (TN)***	
DWvEn (GB,NB,RF)	99.98	100		***Malicious***	***Normal***	0.99
			Malicious	***70 (TP)***	***0 (FN)***	
			Normal	***1368 (FP)***	***3681754 (TN)***	

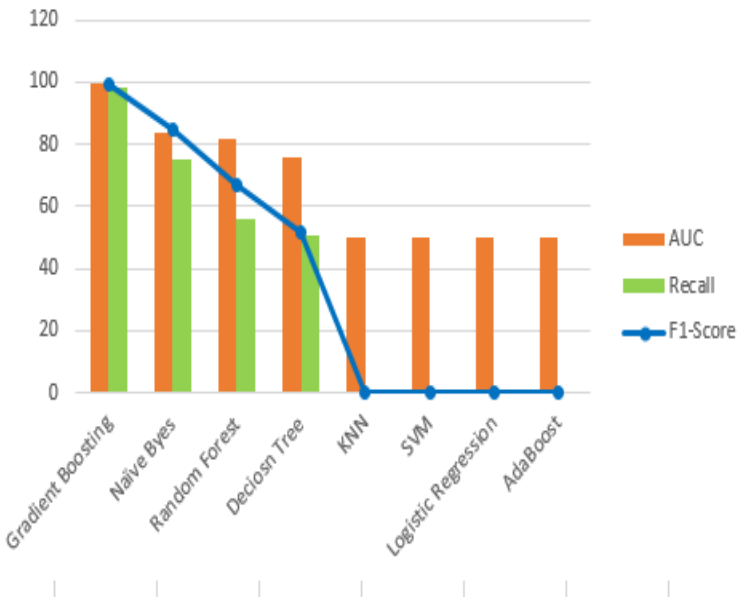


Figure 4.8: Results of Single Classifiers on Test Set

4.2.4 Results by Other Ensemble Learning Techniques

On the basis of results from single classifiers; GB, NB, RF and DT were used for the construction of an ensemble. Different combinations of these classifiers are used for ensemble and weighted voting approach was used for the aggregation of results. Main objective is to evaluate the efficiency of proposed weighted voting approach over static voting approach by using similar set of base classifiers for an ensemble. Thus, the disparity in accuracy can be traced solely with respect to the voting methodologies used. Base learner classifiers used in ensemble are the most popular and effective machine learning algorithms. Parameter setting plays vital role in performance of ML model. To achieve the optimized model, we performed experiments using multiple hyper-parameters settings. The optimal parameters configuration for base learners are presented in Table 4.3.

Results of different ensemble combinations are presented in Figure 4.9. Ensemble composed of NB, GB and RF outperformed other combinations by attaining an accuracy of 99.98%, AUC 0.99 and detection rate 99.63%. The AUC for the dataset was observed for only class 1 samples which can be seen in Figure 4.10.

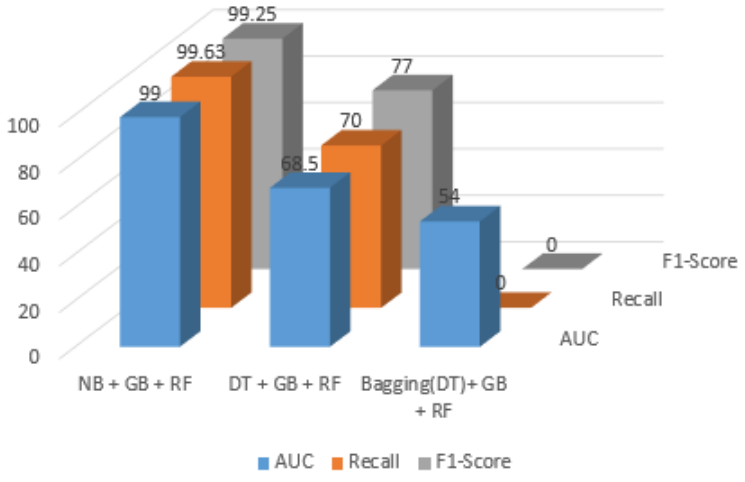


Figure 4.9: Results of Ensemble Learning Techniques on Test Set

To evaluate the performance of our proposed framework, we have performed the experiments on CERT 4.2 version. The result of our proposed framework on both versions of CERT are presented in Table 4.4.

Table 4.5 and 4.6 represent performance evaluation of ensemble learning techniques on

Table 4.3: Configuration Parameters for Base Learners

Algorithm	Parameter Setting
GB	• Learning Rate = 0.5 • N Estimators = 150 • Criterion = 'friedman_mse'
NB	• Var Smoothing = $1e^{-9}$ • Epsilon_ = Float
RF	• N Estimators = 200 • Criterion = 'gini' • Min Samples Split = 2 • Max Features = 'auto'
DT	• Splitter = 'best' • Criterion = 'entropy' • Min Samples Split = 4 • Max Features = 'auto'

CERT 4.2 and 6.2 datasets respectively. The results show that our dynamic weighted-voting technique attained the predictions of each classifier more accurately than the traditional ensembles with voting approaches. DWvEn presents the best performance with the highest recall and AUC, followed by Ensl(DT,GB,RF). In details, DWvEn demonstrates 97.1–98% and 98.7–99% classification accuracy for CERT 4.2 and 6.2 datasets respectively; while Ensl(DT, GB,RF) reports 66.3–68% and 68.5–70.14% under

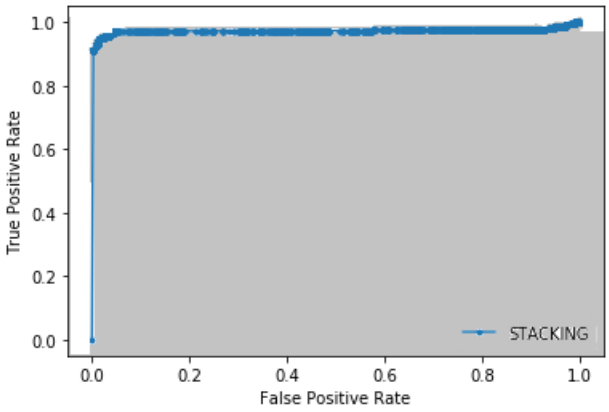


Figure 4.10: AUC of Proposed Framework

Table 4.4: Results of DWvEn on CERT Dataset

Dataset Version	Acc (%)	Rec (%)	Confusion Matrix (%)			AUC
CERT r6.2	99.98	99.76		Malicious	Normal	0.99
			Malicious	70 (TP)	0 (FN)	
			Normal	1368 (FP)	3681754 (TN)	
CERT r4.2	98.70	97.2		Malicious	Normal	0.98
			Malicious	1047 (TP)	30 (FN)	
			Normal	77 (FP)	4917764 (TN)	

similar conditions. Figure 4.11 shows comparison of different ensembles performance on distributed dataset.

The comparison of our proposed framework with state-of-art techniques is presented in Table 4.7.

4.3 Research Contributions

This research work presents a Machine Learning based framework DWvEn to assist cybersecurity analysts in detection of Insider Threats. The significant advantage of DWvEn is that weights assigned on each component classifier of the ensemble are based

Table 4.5: Performance Evaluation of Ensemble Learning Techniques on CERT 4.2 Dataset

Algorithm	**Ratio=10%**		**Ratio =20%**		**Ratio=30%**		**Ratio=40%**	
	AUC	**Recall**	**AUC**	**Recall**	**AUC**	**Recall**	**AUC**	**Recall**
Ensl(DT, GB,RF)	66.3	67	66.7	67.43	66.98	67.77	68	68.5
Ensl(Bag(DT), GB, RF)	55.2	57.1	55.88	57.5	56.12	57.32	56.22	57.57
DWvEn	**97.1**	**96.2**	**97.6**	**96.8**	**97.9**	**97.1**	**98**	**97.2**

Table 4.6: Performance Evaluation of Ensemble Learning Techniques on CERT 6.2 Dataset

Algorithm	**Ratio=10%**		**Ratio =20%**		**Ratio=30%**		**Ratio=40%**	
	AUC	**Recall**	**AUC**	**Recall**	**AUC**	**Recall**	**AUC**	**Recall**
Ensl(DT, GB,RF)	68.5	70	69.23	70.01	69.82	70.21	70.14	70.5
Ensl(Bag(DT), GB, RF)	54.3	56.2	54.67	56.4	54.87	57	55.5	57.64
DWvEn	**98.7**	**99.01**	**98.82**	**99.2**	**98.9**	**99.39**	**99**	**99.63**

on its accuracy on each class of the dataset. The research benchmarks an ensemble learner with three different ML algorithms GB, NB, and RF – on publically available CERT (6.2 and 4.2) data sets. Among the single classifier ML algorithms, GB, NB and RF achieve the high AUC, F1-Score and Accuracy with the lowest false positive rate. Based on the results of single classifiers, we built an ensemble learner and weighted voting approach is applied to identify malicious activities.

Our research results show that the proposed framework is able to effectively learn from the limited training data and generalize to identify new users with malicious activities. Proposed framework achieves a high detection rate of 99.76%, AUC 0.99 and accuracy 99.98% with False Negative to zero. Therefore, we can conclude that the new weighted voting strategy had a significant impact on the performance of all ensembles of self-

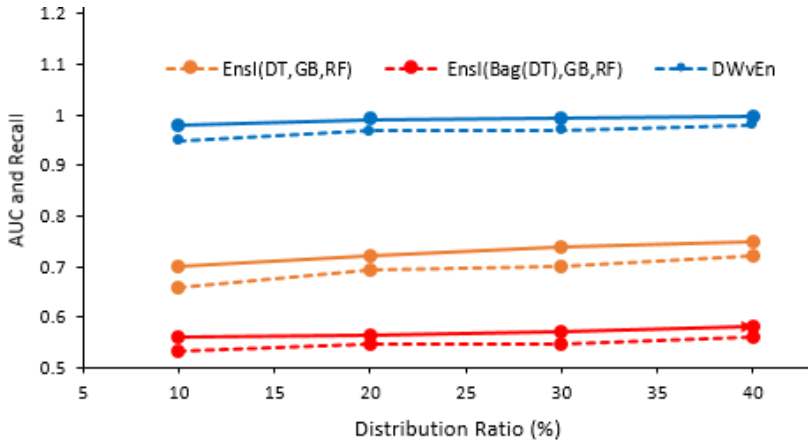


Figure 4.11: Comparative Analysis of Different Ensembles

Table 4.7: Comparison of DWvEn with State-of-Art Techniques on CERT r6.2 Dataset

Model/Techniques	**Acc (%)**
Support Vector Machine (SVM) [96]	80.1
Gaussian density estimation, Parzen window density estimation, PCA, K-Means [125]	90
Ensemble of LSTM, CNN [68]	90.42
PCA, LSTM, Recurrent Neural Network (RNN) [60]	95.53
LSTM [65]	98
Proposed Framework (DWvEn)	**99.98**

labeled algorithms, exploiting the individual predictions of each component classifier more efficiently than the simple voting schemes.

Additionally, it ended up to be more generalized when employed to a different organization's data. Future work will study other ML techniques, such as Deep Learning and Semi Supervised Learning, data availability and data representations for anomaly de-

tection. Moreover, informed attackers' actions can also be introduced to further inspect the performance under more adverse settings.

Chapter 5

Textual Analysis of Traitor-Based Dataset through Semi Supervised Machine Learning

This chapter uses textual analysis and weighs the performance of semi-supervised algorithms to detect the Insider Threats. Textual analysis has been performed using machine learning algorithms including Decision Tree (DT), Naive Bayes (NB), Logistic Regression (LR), KNN, Random Forest (RD), Support Vector Machine (SVM) and Neural Networks (NN). Term Frequency– Inverse Term Frequency (TF-ITF) method used to quantify and evaluate the relevance of words for pre-processing of the dataset.

Enron corpus dataset is used for the exercise and analysis purpose, which consists of personal and official emails of the organization. It is publicly available on the reffered link for researchers [126]. This version of the dataset comprises around 517,431 emails distributed in 3500 folders and contains the information of each of the 151 employees. These emails do not include attachments. Each message contains the emails address of the dispatcher and the receiver, subject, day and period, frame and additional technical niceties.

Since Insider Threat to employers and companies is a complex and growing challenge, detection of Insider Threats has become tedious tasks, consequently traitor-based analysis using modern technologies is essential. Mounting a system to achieve such assignment shams a sum of trials. Firstly, dearth of publicly available datasets for training and

evaluation purpose. Furthermore, the foremost training can be performed through the use of already accessible datasets, which are naturally unlabeled and fail to provide the required outcome.

Consequently, this research work proposes a semi-supervised model, trained and tested on labeled dataset. The major contributions include:

1. Pre-processing the unlabeled dataset and prepare it for training and testing. Insider Threat detection performed through Textual analysis, big data and email logs are worthwhile.

2. Class label identification done through clustering algorithm and prediction of malicious emails by using multiple Machine Learning Classifiers.

3. Applying the supervised and unsupervised algorithms on dataset. Results generated by these algorithms have been compared and analyzed against the authentic acquired datasets.

5.1 Methodology

Machine Learning (ML) is considered Artificial Intelligence (AI)'s subset that stretches machines ability to perform tasks without explicit instructions. By using ML algorithms, computer systems can do many tasks on its own for example, predictions, clustering, pattern recognition, classification and so on. The algorithm takes samples as input called training set, which are described by measurable characteristics called features, and information from training phase is used to check the pattern and link between input and output. Contingent upon the learning style, ML algorithms can be assembled into three principle classifications as shown in Fig 5.1:

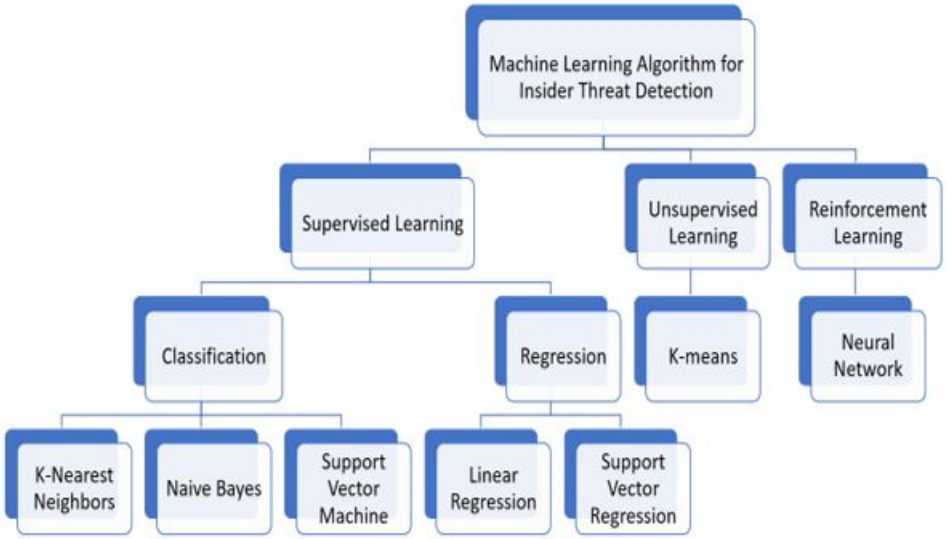


Figure 5.1: Taxonomy of Machine Learning Algorithms

5.1.1 Supervised Learning

In supervised learning category, data set is available and prior knowledge about output is known. There is an association between feedback and response. Supervised learning problems are distributed into two categories "Regression" and "Classification".

5.1.1.1 Classification

In classification method, results are predicted in a discrete output. Input variables are divided into discrete categories. Classifications algorithms are discussed below:

1. **K-Nearest Neighbors:** In this model, unseen and concealed data point is classified by observing the value of K, if these points are closest to it or otherwise. We use distance metrics i.e. Eucliden distance, Mahalanobis distance, L norm to find K nearest neighbor.

   This model needs the whole training data to be deleted, rendering it unsalable to huge data sets which is its limitation. Authors have proposed an answer to this subject by developing a tree-based hunt [127]. Additionally, there is likewise an online adaptation of the KNN characterization, it can similarly be utilized for critical regression tasks [128] yet it is not much of the time utilized for smart data.

2. **Naive Bayes:** It is grounded on the Bayes Theorem. It predicts the unseen data point $z = (z1, z2, z3....zM)$ using bayes formula with the "naıve" supposition of liberation of structures. Naive Bayes classifiers need a small numeral of data topics to be proficient and can contract with high-dimensional data points, and still are speedy and extremely expandable [129].

3. **Support Vector Machine:** These are conventionally binary classifiers which are non-probabilistic. This isolates the training classes set by discovering maximum edge hyperplane. SVMs are also capable of handling multi-dimensional data sets and proficient in optimum memory utilization; which brings them in the category of one of the top supervised learning frameworks. One critical disadvantage of this prototypical is that it does not give probability estimates [130].

5.1.1.2 Regression

In a regression problem, results are predicted within an uninterrupted output. Input variables are plotted to an unremitting function. Regression algorithms are discussed below:

1. **Linear Regression:** It has demonstrated to be a linear model that has a linear linking amongst the feedback variables (x) and the single response variable (y). Approaches for example Ordinary Least Square (OLS), Least Mean Square (LMS), Regularized Least Squares (RLS) are utilized for training the model. LMS is material for smart data since its quick, adaptable to huge data sets and learn parameters online by utilizing stochastic gradient descent [131].

2. **Support Vector Regression:** Support Vector Regression (SVR) is an all-encompassing rendition of the SVM model talked about in the classification section that is utilized to tackle regression issues [132].

5.1.2 Unsupervised Learning

Unsupervised learning permits the line of attack to problems without labeled dataset and no prior knowledge about result is known. Clustering is used to derive the structure-based relationship among the variables.

5.1.2.1 K-means

K-means procedure group the untagged data set into K clusters (groups), in which if data points be appropriate to same bunch have likeness. Similarity is considered by calculating distance between data points [133].

The K-means technique is precisely effective and exceedingly accessible but it has many limitations because it uses Euclidean distance as similarity measures. Moreover, against the outliers, cluster centers not being vigorous. In accumulating, the K-means algorithm allocates only one of the clusters to every one data point, which in certain cases may result in improper clusters.

5.1.3 Reinforcement Learning

A study includes the algorithms that forecast the feedback for a problem conditional to many parameters of modification. At that point, the determined feedback becomes an input constraint and until the ideal output is found, new output is determined. This learning style is used by Artificial Neural Networks (ANN) and Deep Learning.

5.1.3.1 Neural Network

There occurs numerous sorts of neural networks, but we are only focusing on models used for smart data analysis. Usually, a procedure called a neural network entails of discrete, interlocked entities, usually called neurons, nodes, or units. Fig 5.2 demonstrates the assembly of a lone non-natural neuron that obtains feedback from another neuron or data input from one or more sources. The node or artificial neuron makes a weight multiplication of each of these inputs. Then the multiplications are added and the sum is transferred to an activation function. The process leads to a single neuron output [134].

5.2 Proposed Framework

The proposed framework comprises of two core steps.

1. Class label identification through clustering algorithm

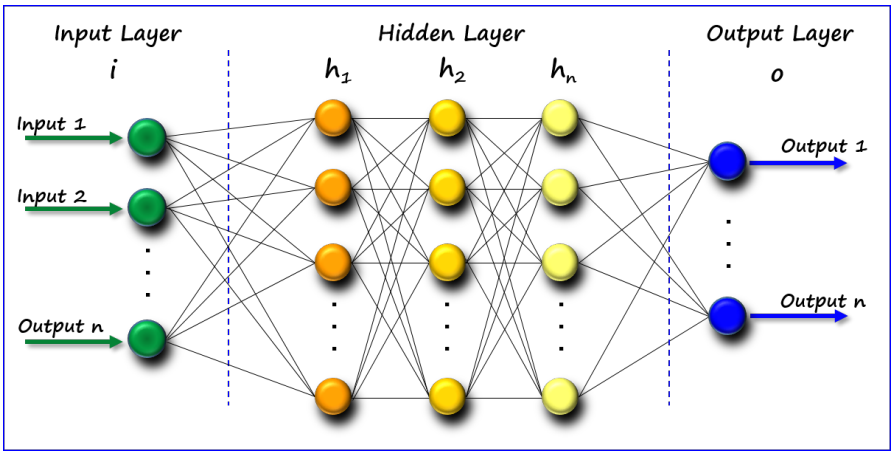


Figure 5.2: Neural Network Architecture

2. Prediction of malicious emails by using machine learning classifiers

Figure 5.3 illustrates the proposed model, which contains four components. Dataset selection and cleaning, Dataset Pre-processing, Transformation and Data labelling respectively. Each component is explained in the following subsections.

The data acquisition method gains data from Enron repository and TWOS research lab. Pre-processing techniques comprise missing value imputation, stop word removal and stemming. Following this, next step is data transformation where vector format is used for the transformation of textual data. After that unsupervised machine learning model K-Means is applied to classify emails based on message content. After te classification, each classifier is trained using the training set. The flowchart of the proposed approach is given in Figure 5.4.

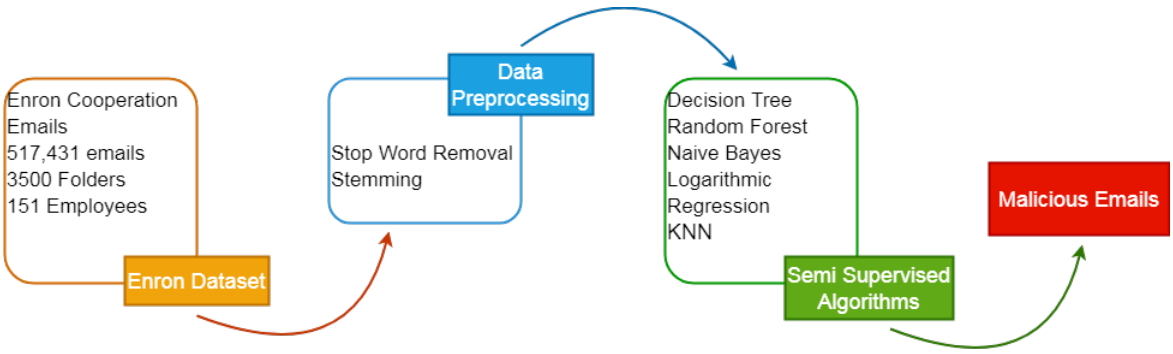


Figure 5.3: High Level System Architecture

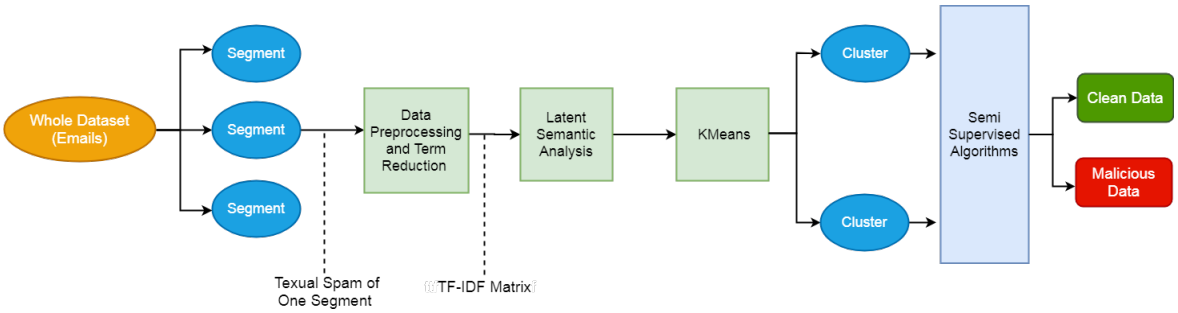


Figure 5.4: Insider Threat Detection using Semi Supervised Learning

5.2.1 Processing Un-Labelled Data

A hypothesis-driven approach, while sometimes useful, spawns risks of missing important information (e.g. the data that do not follow the initial expectation) is prone to user or confirmation bias and makes it more difficult to quantify potential sub-populations in the data. Hence, unsupervised approaches that do not require 'labelled' data and avoid these pitfalls are preferable for those instances [135].

The main objective is to analyze malicious emails comprising sensitive information; for which, there are not many open source email datasets. The best current one is the Enron dataset. Though it is real-world dataset, but available information is unstructured and noisy. However, tagged data are vital for supervised learning. They are frequently obtainable only in small extents, while untagged data may be copious. Prior to analyzing the dataset, we have pre-processed and labeled the dataset. To label the dataset, unsupervised labelling approach K-means has been used. In Fig 5.5, steps of dataset labeling process is shown.

Figure 5.5: Data Labeling Process

5.2.2 Selection of Dataset

5.2.2.1 Enron Dataset

Real-world emails dataset with a user base of 150 has been used with a time window of four years. The dataset encompasses useful information for analysis of email contents directed to detect Insider Threat involving collaborating traitors. The "Enron" email dataset is original and very useful information for traitor-based research on Machine Learning models. This comprehensive dataset can identify the malicious emails from the given features within the data. It contains information about the Corporation; a firm which went bankrupted due to fraudulent business practices in December 2001. Resultantly, Federal Energy Regulatory Commission (FERC) at first released approximately sixteen hundred thousand emails covering the time window between 2000 to 2002 including company's executives but then due to the sensitive nature and contents these were reduced to nearly 0.5 M. Alongside thousands of original emails it also contains the meta data details of sent and received emails. The purpose was to build and provide a platform for artificial intelligence framework which could distinguish the malicious emails symbolizing the Insider Threat. In Fig 5.6, statistical information for Enron dataset has been represented.

No of emails before cleansing	517,431
Period (after cleansing)	01.1999 - 07.2002
No. of removed distinct, bad email addresses	3,769
No. of emails after cleansing	411,869
No. of internal emails (sender and recipient from the Enron domain)	311,438
No. of external emails (sender or recipient outside the Enron domain)	120,180
No. of distinct, cleansed email addresses	74,878
No. of isolated users	9,390
No. of distinct, cleansed email addresses from the Enron domain (social network users) without isolated members the set *IID* in *SNIU*=(*IID*, *R*)	20,750
No. of network users within IID with no activity	15,690 (76%)
Percentage of all possible relationships	5.83%

Figure 5.6: The statistical information for the Enron dataset

5.2.2.2 TWOS Dataset

The Wolf of SUTD (TWOS) dataset was obtained from real host machine user activity which includes both legal user information and suspicious insider activities (masqueraders and traitors). The dataset was collected during the Singapore University of Tech-

nology and Design competition in March 2017 and includes data obtained from six data sources (keystrokes, cursor, host display, network traffic, emails, and login) along with additional findings from the questionnaire on psychological personality. The dataset contains the actions of 24 users collected over a 5-day period. This involves twelve occurrences of the masquerader, each 90 minutes long and five possible instances of the traitor, each 120 minutes long. Emails have been defined as an essential feature for the purpose of detecting insider attacks. In the order of e-mails received by users, the e-mail behaviour of all users is stored within a single file. It includes details such as timestamp, header, sender, receiver, LIWC features extracted from an email message (anonymity message body on particular request).

5.2.3 Data Cleaning

Stacking and reviewing the dataset for uniformity and any anomalies that need fixation in the full dataset. In addition, as aptly suggested in the specified scheme, rows that do not contain email messages are removed.

5.2.4 Data Pre-Processing

Dataset was initially pre-processed by extracting email content from full message as represented in Figure 5.7. After the extraction of email contents, following steps are performed on the dataset.

- Stopword Removal
- Stemming

1. **Stopword Removal** The English words which do not show any impact in a sentence are called stop words. These can easily be ignored without losing the sentence meaning [136].
2. **Stemming** is the method of minimizing a word to its word stem that affixes to suffixes and prefixes or to the roots of words known as a lemma. In natural language understanding (NLU) and natural language processing (NLP), Stemming is vital [137].

Message-ID:
<18782981.1075855378110.JavaMail.evans@thyme>
Date: Mon, 14 May 2001 16:39:00 -0700 (PDT)
From: phillip.allen@enron.com
To: tim.belden@enron.com
Subject:
Mime-Version: 1.0
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit
X-From: Phillip K Allen
X-To: Tim Belden <Tim Belden/Enron@EnronXGate>
X-cc:
X-bcc:
X-Folder:
$\backslash Phillip\_Allen\_Jan2002\_1 \backslash Allen, PhillipK. \backslash' SentMail$
X-Origin: Allen-P
X-FileName: pallen (Non-Privileged).pst
Here is our forecast

Figure 5.7: Extracting Email Content

5.2.5 Data Transformation

In data transformation technique, data is altered from one structure/format into another. Hence, our both datasets were in textual form, we converted them into vector format which is represented in Figure 5.8. Our main goal is to find frequent terms. Therefore, we encoded the altered dataset using TF-IDF as explained in equation 5.2.1. Frequent terms are represented in Figure 5.9.

$$TF - IDF = TF(t, d) \times IDF(t) \tag{5.2.1}$$

Where,

TF = Term Frequency = Number of times term, t appears in doc, d

IDF = Inverse Document Frequency = $log\frac{1+n}{1+df(d,t)} + 1$

5.2.6 Data Labeling

The dataset contains only inputs known as features and no outcomes. In supervised machine learning we work with inputs and concerned end results. In this scenario unsupervised machine learning model K-Means is used to classify emails based on message

['travel', 'to' , 'have', 'a', 'busi', 'meet', 'take', 'the', 'fun',
'out', 'to', 'prepare', 'a', 'present', 'I', 'would', 'suggest',
'hold', 'the', 'a', 'trip', 'without', 'ani', 'formal', busi', 'meet',
'i', 'would', 'even', 'ion', 'om', 'weather', 'a', 'trip', 'is',
'even', 'desire', 'or', 'necessary', meet', 'i', 'think', 'it',
'would', 'be', 'more', 'product', 'to', 'tri', 'offer', 'group',
'about', 'what', 'is', 'work', 'and', 'what', 'is', 'not', 'the',
'other', 'are', 'quiet', 'just', 'wait', 'for', 'their', 'turn', 'held',
'in', 'a', 'round', 'table', discuss', 'format', 'content', 'my',
'play', 'golf', 'and', 'rent', 'a', 'ski', 'boat', 'and', 'jet', 'ski']

Figure 5.8: Vector Representation of Textual Data

content. KMeans is a popular machine learning clustering algorithm, where K stands for the number of clusters. In our research methodology, KMeans classifier with 2 clusters and 100 iterations is used. The emails are classified into 2 clusters as represented in Figure 5.10. After that top terms are extracted from each cluster to find out the cluster that contains malicious emails. Figure 5.11 shows the frequent terms of each cluster. Cluster 1 has weird terms like 'hou' and 'ect' so it is considered as malicious cluster. Hence, all the emails assigned to this cluster are considered as malicious emails.

After this step our dataset get labeled and now supervised machine learning models are applied on the dataset to find out the results.

5.3 EXPERIMENTATION AND RESULTS

Tensor flow library using python language in Anaconda IDE setup is used for the development of the framework. The proposed framework is represented in Algorithm 3. We tuned our model with different parameters for obtaining the best results.

A semi-supervised technique is used by combining unsupervised clustering algorithm with supervised learning classifiers. We have used multiple classifiers for attaining wider spectrum of diversity. The range to which each distinct classifier disagrees about the Area under Curve (AUC); fixes the parameter of diversity. Irrespective of the existing relation between given parameters, each attribute is considered separately by Naive

1.25 (0, 68203)	0.052839156766044
(0, 49356)	0.066108851235679
(0, 64092)	0.101064851560689
(0, 33910)	0.101526406936526
(0, 42003)	0.117719413737451
(0, 19398)	0.106824327240435
(0, 63410)	0.205147790405665
(0, 58960)	0.101351130740114
(0, 36179)	0.091370921025778
(0, 54860)	0.085358617162426
(0, 17275)	0.092704138838047
(0, 36091)	0.059446832795107
(0, 72700)	0.067389089105987
(0, 65946)	0.094786604961902
(0, 49594)	0.046362628417132

Figure 5.9: Data Encoding Through TF-IDF

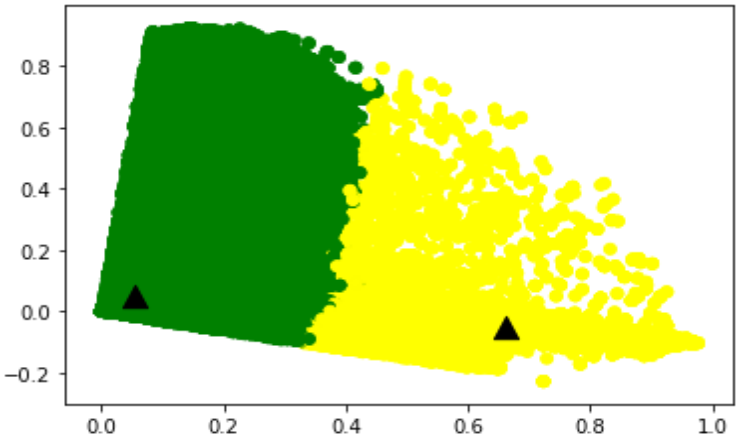


Figure 5.10: Classification of Emails using K-Means

Bayes classifier. However, for achieving optimal results a statistical linkage is determined through Linear Regression model between the given dependent and independent variables. The prediction of numeric output remains confined through Regression model whereas Naive Bayes Classifier asphyxiates this issue. On the other hand, to determine the cost rate and output class; several dependent variables can be effectively handled by

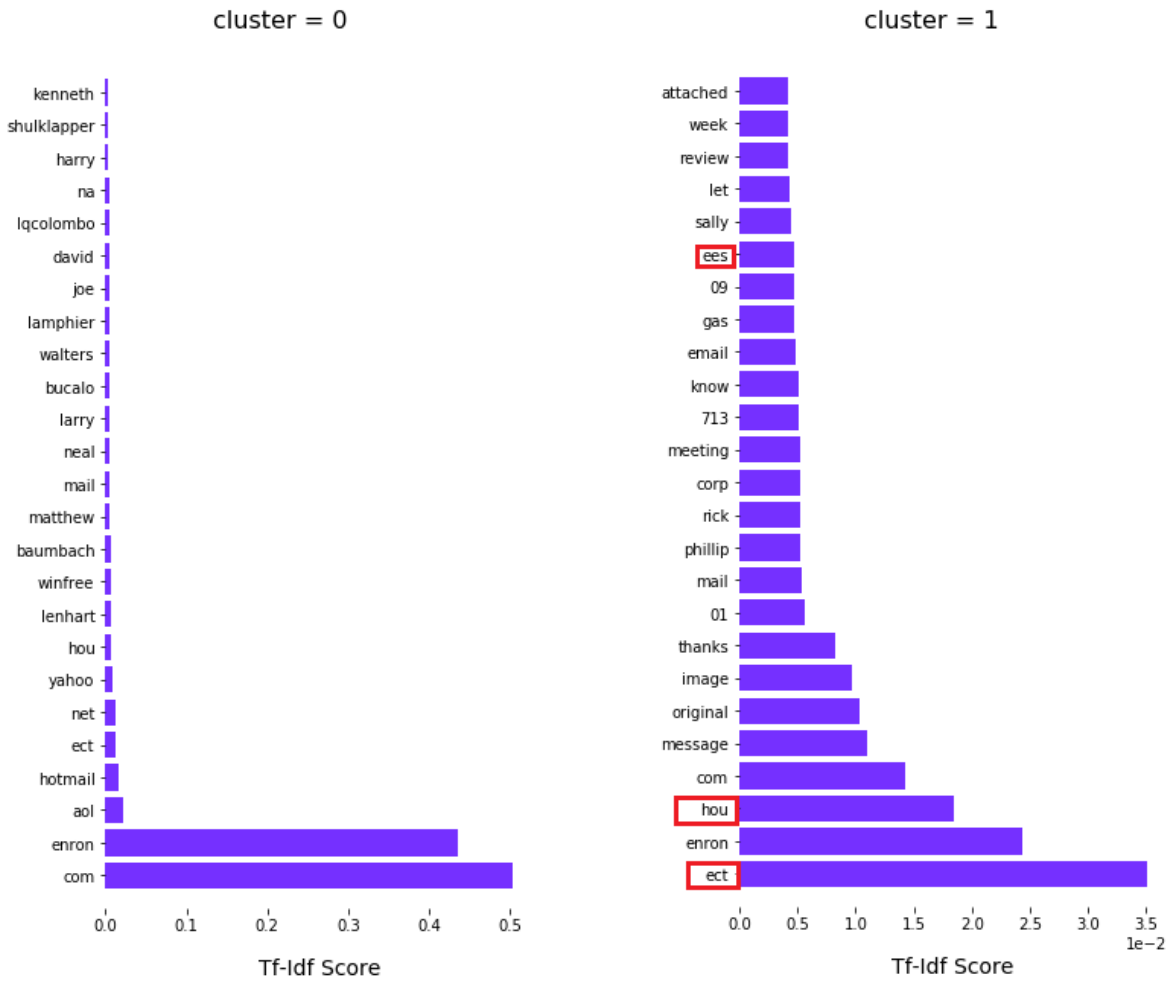


Figure 5.11: Frequent Terms from Each Cluster

Decision tree. The Random Forest Classifier uses subset of data, basing on "information gain" for its feature selection and the K Nearest Neighbour Classifier used distance measures between the instances.

Avoiding complex models with many parameters, limited data concerns have been addressed. By using existing pivotal models, we have restricted their normalization and capacity to overfit.

5.4 COMPARATIVE ANALYSIS

Initial experiments were carried out on unstructured/ raw data set using multiple single classifiers. Namely; Decision Tree (DT), K Nearest Neighbor (KNN), Neural Network (NN), Logistic Regression (LR), Random Forest (RF), Naive Bayes (NB), Support Vec-

Algorithm 3 Semi-Supervised Algorithm

procedure SemiSeupervised(X, y)
    Term reduction and Data Pre-Processing techniques
    X = TF-IDF(X)
    KMeans(X, y)
    $TrainX, Trainy, TestX, Testy = TestTrain(X, y)$
    DT($TrainX, Trainy$)
    while $TrainX \neq 0$ **do**
        z = DT($TrainX$)
    end while
    return z
end procedure

tor Machine (SVM). Results showed an average accuracy of 73% and AUC 0.72, which were inadequate. Hence, to attain the best results, experiments were re-conducted on the proposed model (explained in Section 5.2). The classifiers Decision Tree (DT) and Logistic Regression (LR) with the highest AUC of 0.994 and 0.992 respectively showed worth-mentioning result. However, Decision Tree (DT) was the only classifier to detect 99% of malicious emails.

To evaluate the performance of proposed model, different performance metrics are used. Such as Accuracy (Acc), Recall (Rec) as the false positive are of more significance and Area Under the Curve (AUC), demonstrated in Eqs. 5.4.1 and 5.4.2.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.4.1}$$

$$Rec = \frac{TP}{TP + FN} \tag{5.4.2}$$

Here,

TP = True Positive (Correct Predicted Normal Emails)

TN = True Negative (Correct Predicted Malicious Emails)

FP = False Positive (Malicious Emails Predicted as Normal)

FN = False Negative (Normal Emails Predicted as Malicious)

Experiment's results generated by our proposed framework using multiple classifiers are

tabulated in Table 5.1 and 5.2. For better understanding of model, classifiers' average accuracy is illustrated in Figure 5.12.

Table 5.1: Results of Proposed Model on Test Set of Enron Dataset

Classifier	**Acc (%)**	**Rec (%)**	**Confusion Matrix (%)**			**AUC**
Decision Tree	99.96	99		***Normal***	***Malicious***	0.994
			Normal	***112894 (TP)***	***27 (FN)***	
			Malicious	***14 (FP)***	***15316 (TN)***	
LR	99.92	99.8		***Normal***	***Malicious***	0.989
			Normal	***112905 (TP)***	***16 (FN)***	
			Malicious	***33 (FP)***	***15297 (TN)***	
Random Forest	99.57	97		***Normal***	***Malicious***	0.983
			Normal	***112854 (TP)***	***67 (FN)***	
			Malicious	***484 (FP)***	***14846 (TN)***	
KNN	99.03	95		***Normal***	***Malicious***	0.974
			Normal	***112393 (TP)***	***528 (FN)***	
			Malicious	***708 (FP)***	***14622 (TN)***	
Naive Bayes	95.13	0.61		***Normal***	***Malicious***	0.805
			Normal	***112588 (TP)***	***333 (FN)***	
			Malicious	***5909 (FP)***	***9421 (TN)***	

It is evident from the confusion matrix of Decision Tree (DT) that it predicted the accurate number of malicious emails [138]. Comparison of our model's results with state-of-art techniques is showed in Table 5.3. We are able to achieve better results in term of accuracy and prediction than previous techniques. Not only accuracy is

Table 5.2: Results of Proposed Model on Test Set of TWOS Dataset

Classifier	**Acc (%)**	**Rec (%)**	**Confusion Matrix (%)**			**AUC**
Decision Tree	99.3	99.8		***Normal***	***Malicious***	0.995
			Normal	***129 (TP)***	***5 (FN)***	
			Malicious	***2 (FP)***	***420 (TN)***	
LR	94.8	95		***Normal***	***Malicious***	0.947
			Normal	***100 (TP)***	***34 (FN)***	
			Malicious	***21 (FP)***	***401 (TN)***	
Random Forest	97.5	97		***Normal***	***Malicious***	0.978
			Normal	***128 (TP)***	***6 (FN)***	
			Malicious	***10 (FP)***	***412 (TN)***	
KNN	98.38	99		***Normal***	***Malicious***	0.981
			Normal	***131 (TP)***	***3 (FN)***	
			Malicious	***6 (FP)***	***416 (TN)***	
Naive Bayes	98.02	99		***Normal***	***Malicious***	0.966
			Normal	***126 (TP)***	***8 (FN)***	
			Malicious	***3 (FP)***	***419 (TN)***	

enhanced, but also overfitting issues have been catered for. If we look at the AUC, our proposed model resulted an AUC of 0.994 and 15316/15330 malicious samples were marked accurately in the test set. We observed the AUC and ROC of malicious samples only in the datasets, showed in Figure 5.13. The main purpose of the model is to predict the red class activities; therefore, the False Negatives have been disregarded.

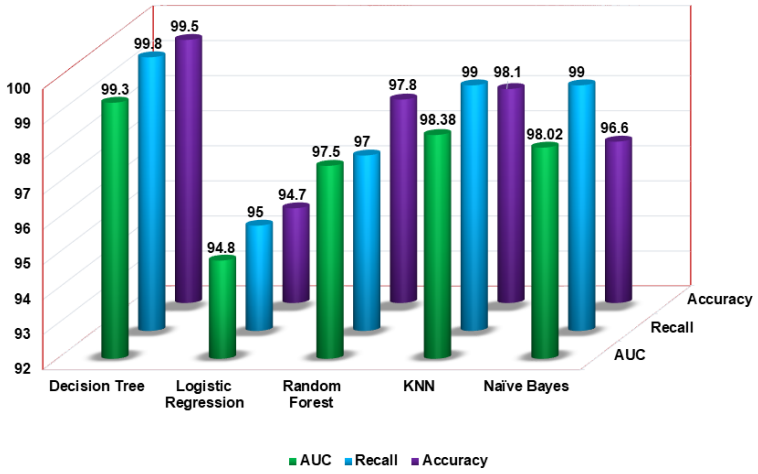


Figure 5.12: Graph Representing Results of Proposed Semi-Supervised Model

Table 5.3: Comparison of Proposed Model with State-of-Art Techniques

Model/Techniques	**Accuracy (%)**
User-Centric Model [31]	85
Artificial Intelligence [139]	88.57
Back Propagation Neural Network (BPNN) [34]	98
Proposed Model (Semi-Supervised Model)	**99.96**

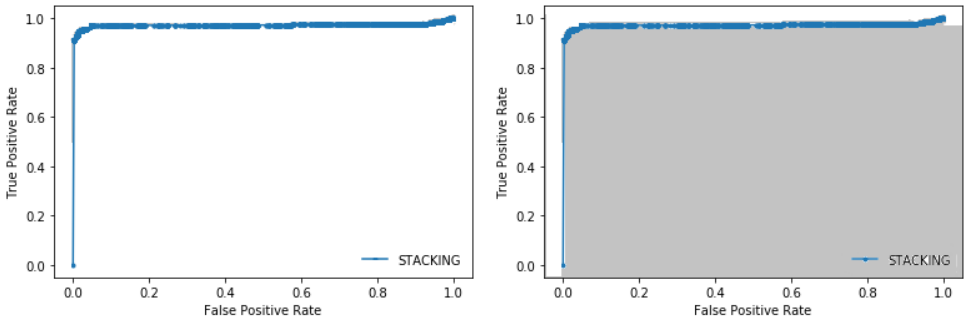


Figure 5.13: AUC and ROC of Decision Tree

5.4.1 Research Limitations

Even though, we have trained and tested the model on multiple datasets but still original labeled data is relatively limited. Model's performance can be improved on a large and

labeled dataset. Though, Enron is a real world dataset but has one attribute (emails) only, which is not sufficient to analyze complete user behaviour and detect malicious activities. Other attributes such as user psychological features, organizational hierarchy and web activities should also be incorporated to assist cybersecurity analysts. Granted that Enron is real world dataset, there is still a gap between the available attributes of real-world data (scenarios) and synthetic data.

5.5 Research Contributions

In the malicious datasets the research dedicated to traitor detection has been very limited. This difference can be explained by the assumption that masquerader detection is simpler and more straightforward than traitor detection, as also argued by Salem et al. [90] who mentioned that "a masquerader is likely to perform actions inconsistent with the victim's typical behavior".

The highlight of this chapter is to use an unlabeled dataset to train and subsequently test a semi-supervised model for detection of an Insider Threat. Class label identification done through clustering algorithm and prediction of malicious emails carried out by using multiple Machine Learning Classifiers. For that purpose, experiments were conducted on the widely used Enron dataset. A preprocessing stage that includes feature extraction and feature reduction processes in the field of machine learning is a vital role for speeding up computation and improving classification accuracy. The problem addressed in this study is associated with data transformation, prior to machine learning classifiers. The unstructured email documents were preprocessed by removing header information, HTML tags, subject content, attachments and leaving only the messages in body to be processed by the proposed approach.

For experiments and performance evaluation, publicly available Enron and TWOS datasets are acquired and processed. After dataset formation, semi supervised model comprised of pre-processing methods like TF-IDF and data labeling technique K-Means combined with machine learning algorithms are harnessed.

Experiments show that Decision Tree combined with the adopted pre-processing technique can achieve the best classification accuracy for malicious emails as well as normal emails with 99.96% Accuracy and 0.994 AUC. Combination of pre-processing techniques

like TF-IDF and decision tree model that correctly classifies 99% of the malicious emails and almost 100% of the normal emails.

Experiments performed on Enron dataset delivered an AUC of 0.994, which outperformed the existing techniques applied on this dataset. Insider Threat detection performed through preprocessing techniques with textual analysis, big data and email logs are worthwhile.

Chapter 6

Handling Insider Threat Through Supervised Machine Learning Techniques

Information technology systems faced cyber security threats, mostly from insiders. Network security mechanism for insiders are not as strict as for rest. Also, insider can easily bypass security or have legitimate access to confidential documents, therefore, to detect and prevent insider threat is a growing challenge. The aim of this chapter is to implement predictive models that are using verbal investigation to conclude an operative's threat level computer-mediated messages, particularly electronic mail. The emails log part of the TWOS dataset has been analyzed using supervised machine learning practices.

The statistical set comprise behavior traces of 24 users observed over 5 day's spam. The outcomes are collated and contrasted for the following algorithms: Adaboost, Naive Bayes (NB), Logistic Regression (LR), KNN, Linear Regression (LR) and Support Vector Machine (SVM).

A Single technique to identify these coercions is supervised learning, which forms models from training data and testing data. Nevertheless, supervised learning necessitates a possibly a costly training development, and is consequently inhibited by a normally small amount of insider threat data obtainable for such exercises. As TWOS dataset size is small (approx. 2000 emails), we have handled this issue by using simple mod-

els. Complex models can make irrational curves that will almost perfectly explain the training data, but possibly will perform poorly over the test data. By avoiding the complex models with many parameters, we achieved the limiting their generalization and possibility of over fitting.

The core phases comprise of following:

1. Pre-processing the labeled dataset and prepare it for training/testing.
2. Choosing the best fitted model among other Supervised Techniques.
3. Training and validation.
4. Discussion besides relative analysis of the research outcomes.

6.1 Methodology

We will implement some methods in this section which have been adapted to analyze the data. This covers dataset definition, methodologies such as machine learning techniques and detection of anomalous emails. Each unit is detailed in the following subsections.

6.1.1 Supervised Learning

Supervised learning is fruitful when the confirm reply is known. All the emails of the dataset were classified as 'Normal Email' or Anomalous Email'. There are six supervised algorithms that are used in classification.

1. **Adaboost** [140]. Also known as 'Adaptive Boosting' is a meta-learning algorithm that integrates frail classifiers into a unique strong classifier. Decision stumps is a frail learner in AdaBoost with decision trees and a single chop. To categorize instances by inserting more weight on tough and less on those already handled well are working of AdaBoost. For classification and regression problem AdaBoost procedures can be used. Adaboost algorithm is explained in Fig 6.1.
2. **KNN** [141]. KNN algorithm explores the complete statistical set for the k amount of utmost identical circumstances, or head-to-head, that specify the matching exemplary as the row with lost figures. The item is cast by the popular poll of its

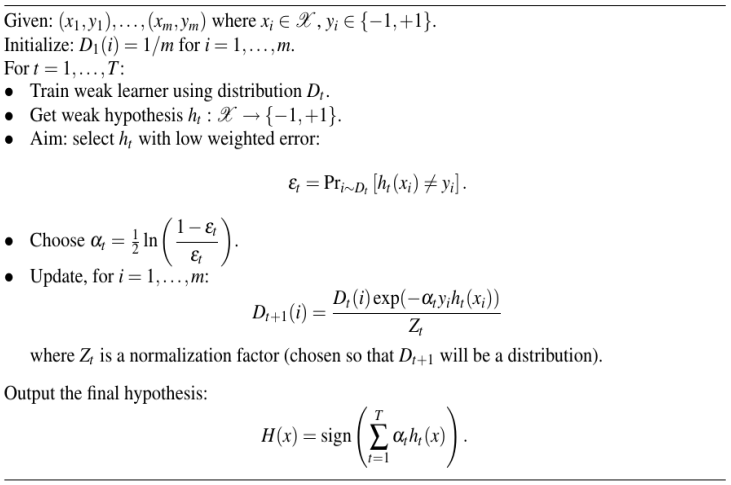


Fig. 1 The boosting algorithm AdaBoost.

Figure 6.1: The boosting Algorithm AdaBoost

nationals, the item is assigned to the most collective class amongst the adjacent neighbors k.

To articulate this delinquent, let us represent the new feedback vector (data point) by x, its K bordering neighbors by Nk (x), the foretold class tag for x by y, and the class variable by a distinct arbitrary variable t. Furthermore, 6.1.1 signifies the indicator function: 1(s)=1 if s is true and 1(s)=0 otherwise. The formula of the ordering task is

$$p(t = c|x, k) = \frac{1}{k} \sum_{i \epsilon N_k(X)} 1(t_i = c) \tag{6.1.1}$$

$$y = arg_c max p(t = c|x) \tag{6.1.2}$$

i.e., the input vector x will be pigeonholed by the approach of its neighbors' tags.

3. **Support Vector Machine** (SVM) [142] [143]. SVM is a direct ideal for grouping. The algorithm produces a contour or a hyper plane which separates the figures into classes.

   At the Paramount, we discourse the linear SVM that discovers a hyper plane that is a linear function of the input variable. To articulate the problematic, we represent the standard vector to the hyper plane by w and the constraint for regulating the offset of the hyper plane from the source alongside its standard vector by b. In addition, to safeguard that SVMs can pact with outliers in the data, we acquaint

with a variable i, titled as a slack variable, for each training point xi , which stretches the distance by which this training point disrupts the margin in units of |w|. This binary linear classification task is defined using inhibited optimization delinquent of the form

$$f(\omega, b, \epsilon) = \frac{1}{2}\omega^t\omega + \sum_{i=1}^{n} \epsilon_i \tag{6.1.3}$$

Subject to

$$y_i(\omega^T x_i + b) - 1 + \epsilon \geq 0 \; i = 1, \ldots, n \tag{6.1.4}$$

$\epsilon_i \geq 0 \; i = 1, \ldots, n.$

Wherever parameter C > 0 defines how profoundly a violation is punished. It ought to be distinguished that even though we used the L1 norm here for the penalty term $\sum_{i=1}^{n} \epsilon_i$, there exist additional penalty terms, such as the L2 norm, which would be preferred with reverence to the necessities of the solicitation.

4. **Linear Regression** [144]. It comprises on conclusion of the most nominal straight line across the lines. The line that finest suit is called the regression line. The goal is to learn a function $f(x, w)$. This is a mapping $f : \varnothing x \mapsto y$, and is a linear combination of a fixed set of linear or nonlinear functions of the input variable, denoted as $\varnothing_i(x)$ and called elementary functions. The form of f(x,w) are subsequent.

$$f(x, \omega) = \varnothing(x)^T\omega \tag{6.1.5}$$

Where w is the weight vector or matrix $w = (w_1, \ldots, w_D)^T$, and $\varnothing = (\varnothing_1, \ldots, \varnothing_D)^T$. There occurs a comprehensive variety of elementary functions, such as polynomial, gaussian, radial, and sigmoidal basic functions, which ought to be preferred with reverence to the use.

5. **Naïve Bayes** [145]. Naïve Bayes is a plain learning algorithm which uses Bayes rule at the same time with a tough supposition that, given the class, the attributes are readily self-contained.

   Prearranged a new, unobserved data point (input vector) $z = (z1, \ldots, zM)$, naive Bayes classifiers, who are the kinfolk of probabilistic classifiers, classify z based on applying Bayes' theorem with the "naive" supposition of independence amid the features (attributes) of z given the class variable t. By applying Bayes' theorem.

$$p(t = cz_1, \ldots . z_m) = \frac{p(z_1, \ldots . z_m t = c)t(t = c)}{p(z_1, \ldots . z_m)} \tag{6.1.6}$$

And by smearing the naive independence supposition and some generalizations, we have

$$p(t = cz_1, \ldots . z_m) \infty \; p(t = c) \prod_{j=1}^{M} p(z_j | t = c) \tag{6.1.7}$$

Consequently, the practice of the classification assignment is

$$y = arg_c \; maxp \; (t = c) \prod_{j=1}^{M} p(z_j | t = c) \tag{6.1.8}$$

Wherever y denotes the predicted class label for z. Diverse naive Bayes classifiers use different tactics and dissemination to estimate p(t=c) and $p(z_j|t = c)$.

Even though this assumption of freedom is often abused in routine, nevertheless, naïve Bayes still delivers accuracy for reasonable classification. The calculating performance and many other attractive features are mixed.

6.1.1.1 Comparative Analysis of Supervised Learning Algorithms

Supervised learning algorithms deals with labeled data. There are two distinct types of methods in this category: classification and regression.

SVM can model the boundaries of non-linear decisions. Nevertheless, SVM remains characteristically memory-intensive, and it is tough for a SVM to choose on an appropriate kernel, making it problematic to model with large data sets. SVM performs better when dealing with multi-dimensional and continuous data.

Although Naive Bayes (NB) is used to model practical problems such as grouping of text and identification of spam. Being simple and autonomous of each other all input features makes arbitrary forest algorithms seamless for modeling real-world concerns. Random forest algorithms are easier to appliance and adjust to the dimensions of the data set accessible. Random Forest is more predictive and takes less time to predict and gives high accuracy result.

Famous regression algorithms are the K Nearest Neighbors and logistic regression. Such algorithms exist also referred to as "instance-based," which anticipate every one of the new reflection by observing for the maximum identical training data. Conversely, these algorithms are memory-intensive and execute poorly for large-dimensional data.

In our scenario, Adaboost outperformed the other algorithms of supervised machine learning.

6.2 Detection of Anomalous Emails

Insider can cause massive harm in the current IT environment, and we cannot thwart or diminish damage from mischievous insiders. To resolve the limit of compliance with automation information security, human elements should also be examined. We re-structured the email as an important factor for insider threat detection [146]. The recommended framework comprises of two core steps.

6.2.1 Proposed Framework

The proposed model contains four units: data collection and pre-processing unit, data transformation unit, supervised learning unit and classification unit respectively. Fig 6.2 shows the shows the summarized form of our framework.

Stage 1: Data Collection and Preprocessing.

- Acquired the dataset from authors and studied it thoroughly.
- Separated the useful information i.e. emails and merged all CSVs.
- Cleaned and pre-processed the data using multiple feature engineering techniques.

Stage 2: Data Transformation.

- After applying pre-processing techniques, analyzed the available dataset.
- Converted the textual data into vector form using TF-IDF.

Stage 3: Supervised Learning.

- As the dataset is labelled, supervised learning is the best fit as per our knowledge.
- Firstly, Dataset was trained/tested on KNN, Naïve Bayes, Linear Regression and SVM.
- Results of above-mentioned algorithms were not satisfactory.

- After performing further RD, Adaboost was applied on dataset and much improved results were achieved.

Stage 4: Validation of the Prediction Model.

- Choose the proper validation method to validate the result from prediction model.
- Analyze the findings.

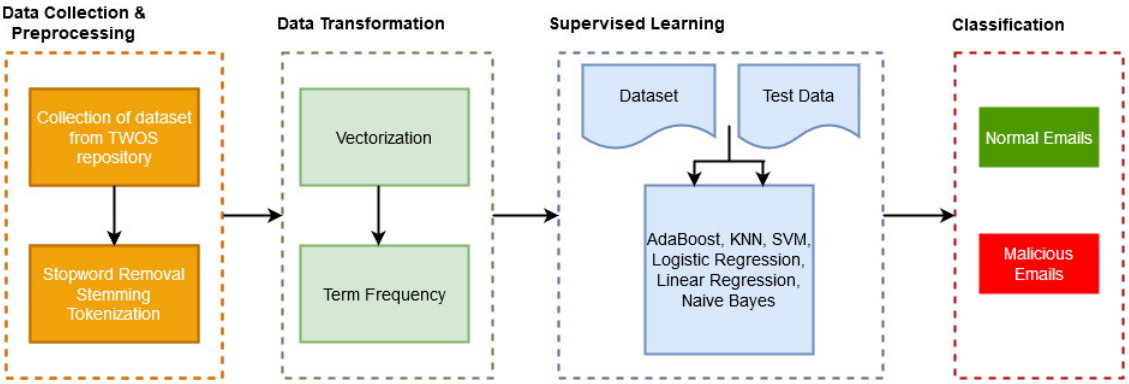


Figure 6.2: System Overview

6.3 Processing Labelled Data

There are following five steps to evaluate feasible anomalous emails by inspecting composed emails data. In the first step, data collected from TWOS repository. Pre-processing steps involve missing value accusation, removal of stop words, stemming and tokenization. Then it is followed by transformation of data where textual data is converted into vector form. After pre-processing, machine learning algorithms required have been functional to organize the emails. Steps of data processing are shown below 6.3.

6.3.1 Dataset

The TWOS dataset [147] was gathered from actual user interaction with the proposed machine, which comprehends both authentic consumer information and malevolent insider occurrences. The figures were coined together during the March 2017 Singapore University of Technology and Design competition and comprise data acquired from six

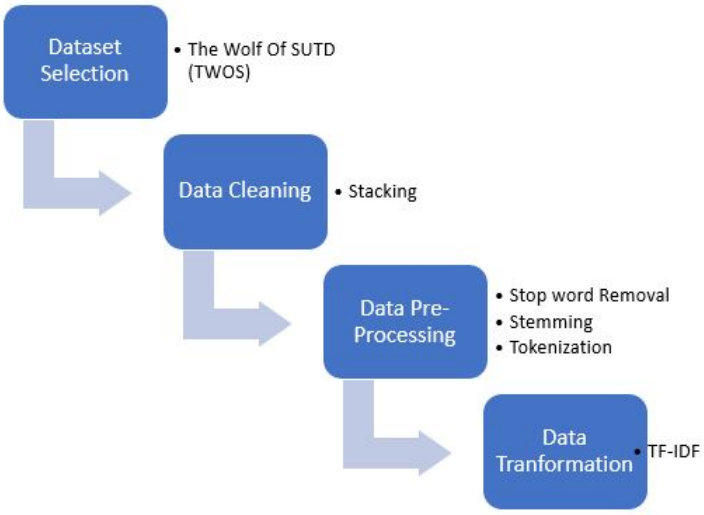


Figure 6.3: Steps of Data Processing

data foundations (keystrokes, mouse, host monitor, network traffic, emails, and login) alongside supplementary psychological behavior survey results. The dataset contains behavior of 24 users that were collected over a 5-days span. This contains twelve instances of the masquerader, each 90 minutes long and five potential instances of the traitor, each one 120 minutes long.

Dataset that we acquired, consisted on 6 CSV files. Each file has separate information, which is described in detail here:-

1. **Keystrokes.csv:** Keystroke activity of each user was found within one or more files named according to user ID followed by an optional timestamp. Depending on the amount of interaction of the user, authors have created multiple files as a result of log rotation. It contains information pertaining to timestamp, key press / release event, key value and username.

2. **Mouse.csv:** Similar to keystroke logs, mouse activity of each user is present within one or more files named according user ID followed by an optional timestamp. Depending on the amount of interaction of the user, multiple files have been created as a result of log rotation. It encloses information pertaining to timestamps, cursor movement / click ratio /, coordinates of mouse pointer and username.

3. **Emails.csv:** Email activity of all users are within several DSV files, where each file represents a particular user - rows with the label NORMAL are emails sent

by that user while rows with label ANOMALY were built as random mixtures of other users' emails with emphasis to include samples from all remaining users - due to this fact, the ratio of anomaly rows varies among particular users. Each file contains information such anonym zed body of message, and structures extracted by Linguistic Inquiry and Word Count (LIWC) apparatus. We have used body of the emails and LIWC features for some classification task related to authorship verification/user identification.

4. **Host Monitor.csv:** The system calls of each user can be found within multiple files that contain the name of the user. It contains file system, registry, process and network related information. Specifically, it contains information such as timestamp, process name, PID, Parent Process name, Parent PID, system call operation.

5. **Personality Tests.csv:** All users were asked to fill in personality test containing 50 questions. Writer's castoff the opinion poll encouraged by dark triad philosophy. This questionnaire enabled investigators to associate participants' behavior with emotional gauges.

6. **Network Traffic.csv:** Network activity of all users are logged into several pcap files that are consecutive in time (they were captured one by one). The capture of the pcap files started 1 day before the competition and finished 30 minutes after official end of the competition. They contain information such as HTTP Request (e.g, GET, POST) / Response, status cipher, content span, and content category.

Emails were found to be an important attribute aimed at the intention of Insider Threat exposure. In the order of emails sent by the users, email activity of all users is contained inside a single file. It includes data such as timestamp, header, sender, recipient, LIWC features mined from an electronic message body (on specific request anonymized message body). We have taken emails.csv and performed data processing techniques (Explained in the following sections) on it for better results.

6.3.2 Data Cleaning

Developing assessment from AI- and machine learning-based expertise critically is subject to the value of the underlying statistics. Study in data cleaning has delivered a

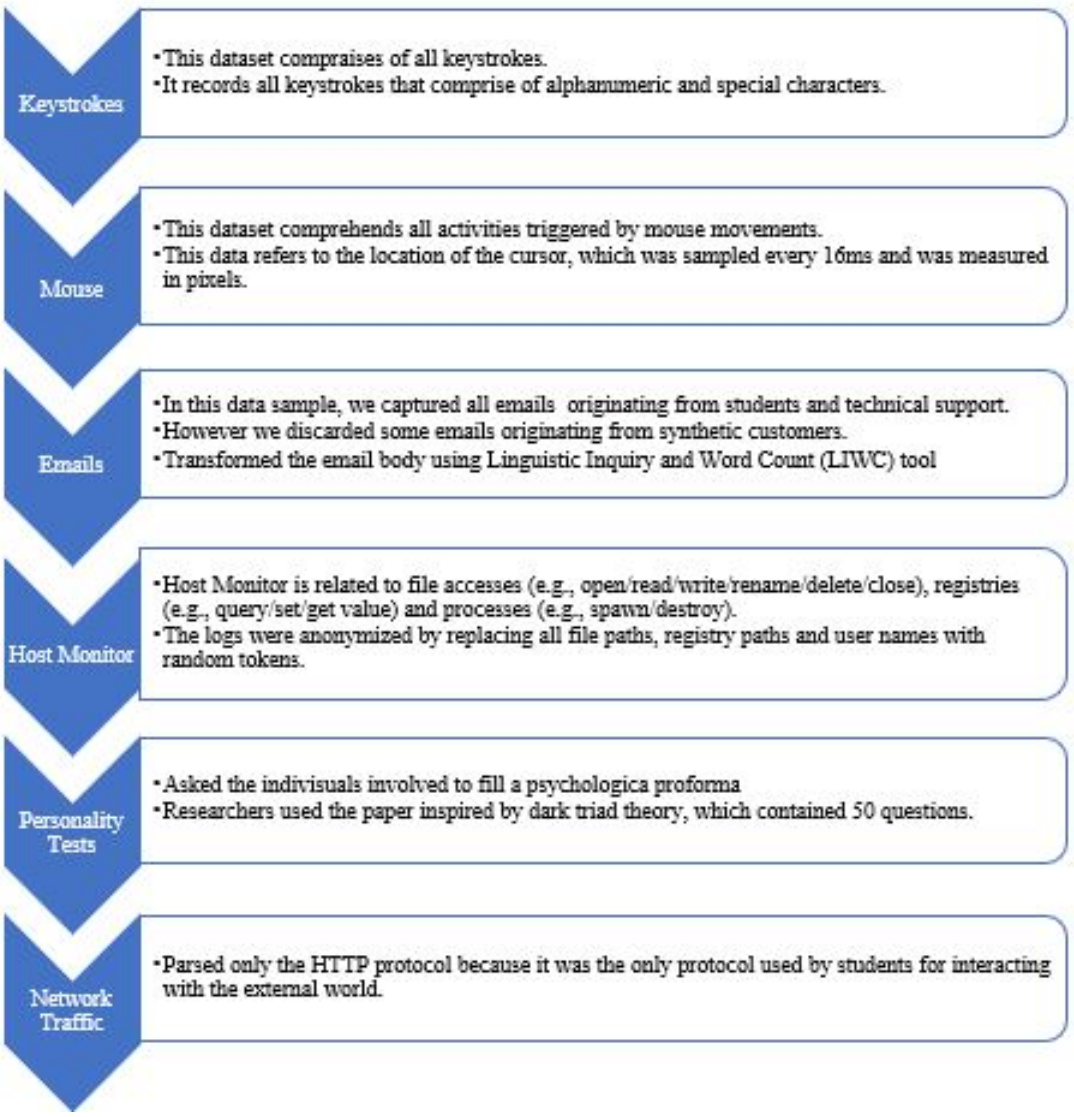


Figure 6.4: Summary of TWOs Dataset

variation of methodologies to report diverse and multiple data complications. We have implemented the Stacking in the entire dataset to analyze the whole dataset for consistency and for any discrepancies that require fixation.

In addition to stacking, all rows containing messages, text and null values were removed or filled in with suggested/appropriate values in the defined scheme.

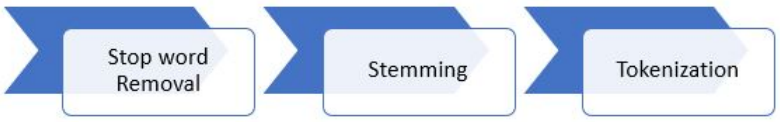
6.3.3 Data Pre-Processing

The dataset consists of .csv files that emails of different users. Firstly, we have merged all emails files are merged into a single heading. It has consumer ID, Email Content,

tweets, and label of email in the file.

Secondly, pre-processing is performed on the dataset that consists of following steps:

- Stopword Removal
- Stemming
- Tokenization



1. **Stopword Removal** Stop words are the English verses which do not show any impact in a sentence. They can easily be ignored without losing the sentence meaning [148].
2. **Stemming** is a technique of lessening a word to its word stem that attaches to suffixes and prefixes or to the origins of words known as a lemma. In natural language understanding (NLU) and natural language processing (NLP), Stemming is a vigorous activity [149].
3. **Tokenization** is the way to riven the given writing into parts called tokens. Tokens can be solitary words, phrases or even whole sentences. In the technique of tokenization, some types such as punctuation marks may be unrestrained. The tokens usually turnout to be input for the processes like vectorization [150].

6.3.4 Data Transformation

Data transformation shows a significant part in data mining and machine learning. Data transformation is the technique of altering data from one format into a different form as explained in equation 6.3.1. The influence of different transformations varies. The email data consist of textual data that is converted into vector layout as shown in Figure 6.5. After the conversion of textual data to vector form by using TF-IDF vectorize. TF-IDF vectorize change a cluster of raw catalogues to a matrix of TF-IDF features as shown in Figure 6.6.

$$TF - IDF = TF(t, d) \times IDF(t) \tag{6.3.1}$$

Where,

TF = Term Frequency = Number of times term, t appears in doc, d

IDF = Inverse Document Frequency = $log\frac{1+n}{1+df(d,t)} + 1$

```
['the', 'follow', 'item', 'were', 'present', 'in', 'this', 'week', 'corpor', 'chang', 'control', 'jan', '28',
'begin', 'at', '6:00', 'p', 'm', 'the', 'telephoni', 'team', 'will', 'remov', 'the', 'telephon', 'switch', 'fr
om', '3ac', 'and', 'migrat', 'phone', 'number', 'the', 'mainten', 'is', 'expect', 'to', 'last', 'approxim', 'o
ne', 'hour', 'each', 'block', 'of', 'number', 'will', 'experi', 'an', 'outag', 'of', 'a', 'few', 'minut', 'dur
e', 'which', 'call', 'made', 'to', 'an', 'affect', 'number', 'will', 'not', 'go', 'through', 'and', 'will', 'n
'omaha', 'at', '402-398-7454']
```

Figure 6.5: Vector Representation of Textual Data

```
(0, 67767)    0.06937179449416819
(0, 47504)    0.3181875690631305
(0, 54789)    0.06792312041150975
(0, 38522)    0.09171483341147117
(0, 65942)    0.08369842513125694
(0, 73477)    0.14727563973076654
(0, 55895)    0.07271542548196704
(0, 55834)    0.082591831756838
(0, 74440)    0.03001431013948936
(0, 39721)    0.08716918875502584
(0, 31817)    0.09028669441540148
(0, 69267)    0.24174194244617922
(0, 51728)    0.032439071476644
(0, 52589)    0.05118235333859011
(0, 34813)    0.08022860520331387
(0, 67707)    0.18247858140507917
```

Figure 6.6: Data Encoding Through TF-IDF

6.4 Experimentation and Results

We implemented the framework in Python with Tensor flow in backend. Anaconda IDE was used for development. The algorithm for the proposed framework is demonstrated in Algorithm 4.

Initial experiments were carried out on unstructured/ raw data set using multiple single classifiers. Namely, Decision Tree (DT), K Nearest Neighbor (KNN), Neural Network (NN), Logistic Regression (LR), Random Forest (RF), Naive Bayes (NB), Support Vector Machine (SVM). Results showed an average accuracy of 73% and AUC 0.72, which were inadequate. Hence, to attain the best results, experiments were re-conducted on the proposed model (explained in Section 3). The classifiers Decision Tree (DT) and

Algorithm 4 Supervised Learning Algorithm

1: **procedure** SupervisedLearningAlgorithm(X, y)
2: Data Pre-Processing and Term Reduction
3: Formation of TF-IDF Matrix
4: AdaBoost(X, y)
5: St(X) = h(AdaBoost)
6: **end procedure**

Logistic Regression (LR) with the highest AUC of 0.994 and 0.992 respectively showed worth-mentioning result. However, Decision Tree (DT) was the only classifier to detect 99% of malicious emails.

A number of single classifiers were run on dataset. Namely, AdaBoost, K Nearest Neighbor (KNN), Support Vector Machine (SVM), Logistic Regression, Linear Regression and Naive Bayes (NB). The single classifiers give a satisfactory performance, with highest AUC of 0.983 and 0.95 with AdaBoost and KNN respectively. However, only AdaBoost detected 98% of the malicious emails. The performance of the model was measured with performance metrics including Accuracy (Acc), Recall (Rec) as the cost of false positive is high and Area under the Curve (AUC) given in Eqs. 6.4.1 and 6.4.2. The results produced by traditional models are reported in Table 6.1 and Figure 6.7.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{6.4.1}$$

$$Rec = \frac{TP}{TP + FN} \tag{6.4.2}$$

Here,

TP = True Positive (Correct Predicted Normal Emails)

TN = True Negative (Correct Predicted Malicious Emails)

FP = False Positive (Malicious Emails Predicted as Normal)

FN = False Negative (Normal Emails Predicted as Malicious)

Confusion Matrix for the applied AdaBoost model is shown which displays the correct number of malicious samples predicted with the model. Considering this matrix alone we have achieved much better results. The proposed technique gave an AUC of 0.983 with 495 out of 504 anomalous samples predicted correctly in the test set. The ROC

Table 6.1: Results of Single Classifiers on Test Dataset

Classifier	Acc (%)	Rec (%)	AUC	Confusion Matrix (%)		
AdaBoost	98.3	98	0.983		Malicious	*Normal*
				Malicious	495 (TP)	9 (FN)
				Normal	8 (FP)	488 (TN)
KNN	96	97	0.95		Malicious	*Normal*
				Malicious	488 (TP)	16 (FN)
				Normal	24 (FP)	472 (TN)
SVM	95	95	0.95		Malicious	*Normal*
				Malicious	478 (TP)	16 (FN)
				Normal	24 (FP)	472 (TN)
LR	94.8	93	0.947		Malicious	*Normal*
				Malicious	479 (TP)	25 (FN)
				Normal	27 (FP)	469 (TN)
LR	93.2	91	0.975		Malicious	*Normal*
				Malicious	450 (TP)	52 (FN)
				Normal	18 (FP)	478 (TN)
NB	73.7	76	0.48		Malicious	*Normal*
				Malicious	374 (TP)	130(FN)
				Normal	166 (FP)	336 (TN)

and AUC for the dataset was observed for only malicious samples which can be seen in Figure 6.8 respectively. Here, the key purpose of the model is to predict the anomalous

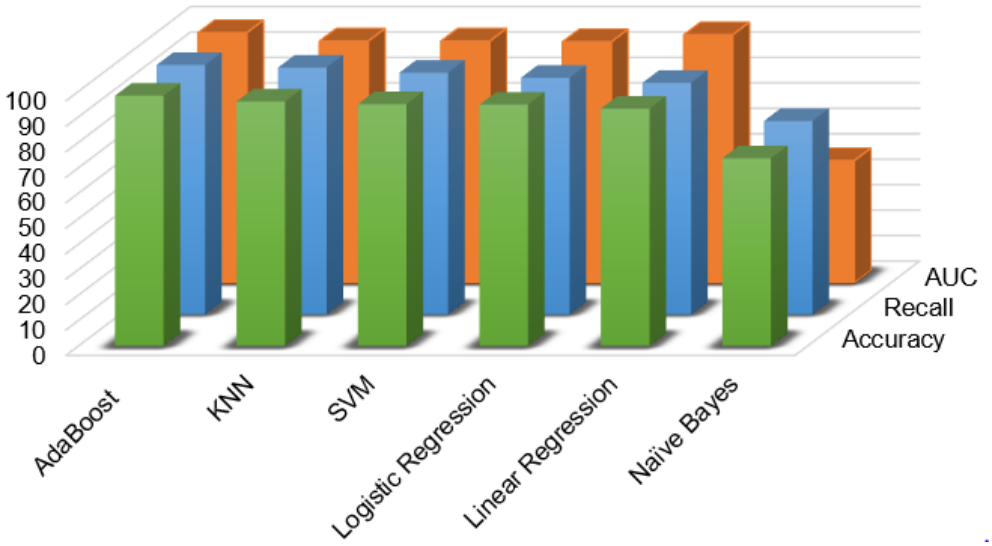


Figure 6.7: Graph Representing Results of Supervised Learning Algorithms

emails, so the False Negatives in this case have been disregarded.

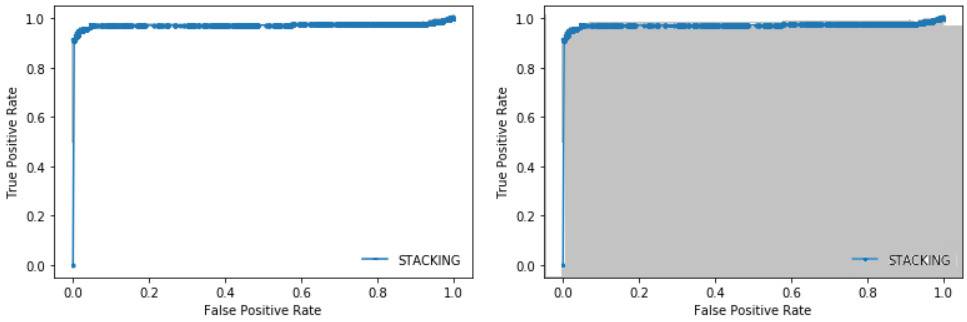


Figure 6.8: ROC and AUC of AdaBoost

6.5 Research Contributions

This chapter proposed a technique for the identification of anomalous emails. Experiments show that AdaBoost can achieve the best classification accuracy for malicious emails as well as normal emails with 98.3% Accuracy and 0.983 AUC. Combination of pre-processing techniques like TF-IDF and AdaBoost model that correctly classifies 98% of the malicious emails and almost 98.3% of the normal emails. TWOS dataset was acquired through email communication with researchers of Cyberlab of Singapore University of Technology and Design (SUTD) after signing the given contract. Experiments results show our framework achieved an AUC of 0.983 that outperformed the existing

techniques applied on this dataset.

Our main contributions are as follows:

1. Presented a perspective that supervised machine learning and data mining can be applied proficiently to detect Insider Threats.

2. Proposed a supervised learning framework that tackles with evolving concepts using the algorithm Adaboost.

3. Effectively addressed the trial of limited labeled training data and handled the overfitting issue.

4. Finally, compared our methodology with existing supervised learning models and established its efficiency by using real-world Insider Threat data.

Chapter 7

CONCLUSION AND FUTURE WORK

7.1 Conclusion

Concluding this research with swift sight; four complementary frameworks were contributed. These frameworks focused on handling of class imbalance in datasets, detection of insider threats, differentiate malignant / normal emails and identify the anomalies. The sole aim is mitigating and containing Insider Threats causing intentional and unintentional damages in systems, that comprises multitude of parameters consisting email logs, metadata, user contours and psychometric data. The entire knowledge discovery process was done by laborious work, data gathering, data preprocessing techniques, picking appropriate Machine Learning techniques to find patterns among the data and interpreting them.

Primarily, a framework to report the class imbalance problem and proposed a hybrid ensemble technique StackBagNB. Our approach combines feature engineering techniques like low variance filter (LVF) and ensembles to produce a nested learner StackBagNB that decorously classifies the insider samples and normal activities on publicly available CERT 4.2 dataset. The contributions made in this area are not limited to the nested ensemble technique but also include the introduction of new derived variables produced. We are quite certain and confidant in the fallouts presented by our framework that made it tough for insider attackers in exploiting their privileges.

Secondly, a Weighted Voting based framework DWvEn to assist Cyber Security Analysts in detection of insider threats. The research benchmarks an ensemble learner with three different ML algorithms – GB, NB, and RF – on publicly available CERT (6.2 and 4.2) data sets. Among the single classifier ML algorithms, GB, NB and RF achieve the high AUC, Recall and Accuracy with the lowest false negative rate. Based on the results of single classifiers, we built an ensemble learner and a weighted voting approach is applied to identify malicious activities. Results show that the proposed framework is able to effectively learn from the limited training data and generalize to identify new users with malicious activities. The system achieves a high detection rate, AUC and accuracy. Additionally, it seems to be more generalized when employed to a different organization's data.

Thirdly, a semi-supervised framework to regulate the risk exposure of email logs initiated by an insider. As a fragment of this framework, we offered and assessed an approach to correctly identify malignant and normal emails. Our methodology applies semi semi-supervised machine learning taxonomy on a valuable collection of Enron corpus for the identification of malicious emails. In addition, we demonstrated the experiment outcomes to calculate the projected framework. We have faith in that as a consequence of our recommended research, administrations with security necessities will more probably clasp the benefits that AI framework offer.

Finally, our fourth proposed framework is designed to detect anomalies. Few approaches have incorporated supervised learning into the insider threat detection and none of them have considered the intricacies of incorporating limited labeled training data. We performed analysis and pre-processing on TWOS dataset. To capture these new threats, we proposed a model built on supervised machine learning techniques. Imposing these limitations assists in reducing the threat of insider attacks. The contributions made in this paper are not limited to the supervised learning techniques but also include the introduction of new derived variables produced. We have successfully catered the over fitting in a small dataset by using simple models.

7.2 Future Work

As it is the situation with any thought-provoking problem, there is a great deal of potential work to be done in this field. With admiration to data collection, our frameworks require observing users' behavior. This suggestion is fed to a probabilistic system to determine the probability of an access request that can result in an attack. In this report, we believed that this material could be accessed and evaluated without restrictions. However, in order to implement this program in specific situations, it is important to address a variety of legal, privacy and ethical problems. Potential work may involve the creation of technical solutions that offer privacy protections for consumers. In fact, strategies are required to ensure that details on internal attacks are not spoofed. In addition, explainable Deep Learning for Insider Threat detection to make prediction results understandable to human is key toward a trustworthy and reliable Insider Threat detection model.

So far as policy specification is concerned, future research involves developing digital interfaces to define risk-and-trust policies as well as undertaking usability tests to help select interfaces that reduce policy specification errors. Moreover, Multi-model Learning based Insider Threat Detection to combine the user activity data with user profile data and user relationship data is under-exploited and worthy to explore.

Ultimately, we conclude that the methodologies, methods and research provided in this dissertation are essential for understanding and avoiding Insider Threats. Our suggested work is of interest to many groups and organizations.

Bibliography

[1] J. Bush. Survey suggests economy could lead to cybercrime increase. *Purdue University News Service*, 2009.

[2] Song Han, Miao Xie, Hsiao-Hwa Chen, and Yun Ling. Intrusion detection in cyber-physical systems: Techniques and challenges. *IEEE systems journal*, 8(4): 1052–1062, 2014.

[3] Pallabi Parveen, Zackary R Weger, Bhavani Thuraisingham, Kevin Hamlen, and Latifur Khan. Supervised learning for insider threat detection using stream mining. In *2011 IEEE 23rd international conference on tools with artificial intelligence*, pages 1032–1039. IEEE, 2011.

[4] Robert Mitchell and Ing-Ray Chen. A survey of intrusion detection techniques for cyber-physical systems. *ACM Computing Surveys (CSUR)*, 46(4):1–29, 2014.

[5] Martin Naedele. Addressing it security for critical control systems. In *2007 40th Annual Hawaii International Conference on System Sciences (HICSS'07)*, pages 115–115. IEEE, 2007.

[6] CMS collaboration et al. A deep neural network to search for new long-lived particles decaying to jets. *Machine Learning: Science and Technology*, 1(3):035012, 2020.

[7] John C Russ, James R Matey, A John Mallinckrodt, and Susan McKay. The image processing handbook. *Computers in Physics*, 8(2):177–178, 1994.

[8] Michael I Jordan and Tom M Mitchell. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260, 2015.

[9] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[10] Bernard Adam and Ian F Smith. Reinforcement learning for structural control. *Journal of Computing in Civil Engineering*, 22(2):133–139, 2008.

[11] Adam J Barker, Harry Style, Kathrin Luksch, Shinichi Sunami, David Garrick, Felix Hill, Christopher J Foot, and Elliot Bentine. Applying machine learning optimization methods to the production of a quantum gas. *Machine Learning: Science and Technology*, 1(1):015007, 2020.

[12] Karel van den Bosch and Adelbert Bronkhorst. Human-ai cooperation to benefit military decision making. NATO, 2018.

[13] Fei Jiang, Yong Jiang, Hui Zhi, Yi Dong, Hao Li, Sufeng Ma, Yilong Wang, Qiang Dong, Haipeng Shen, and Yongjun Wang. Artificial intelligence in healthcare: past, present and future. *Stroke and vascular neurology*, 2(4), 2017.

[14] David McArthur, Matthew Lewis, and Miriam Bishary. The roles of artificial intelligence in education: current progress and future prospects. *Journal of Educational Technology*, 1(4):42–80, 2005.

[15] Amelie Gyrard and Amit Sheth. Iamhappy: Towards an iot knowledge-based cross-domain well-being recommendation system for everyday happiness. *Smart Health*, 15:100083, 2020.

[16] Andrea Zanella, Nicola Bui, Angelo Castellani, Lorenzo Vangelista, and Michele Zorzi. Internet of things for smart cities. *IEEE Internet of Things journal*, 1(1): 22–32, 2014.

[17] Razvan Pascanu, Jack W Stokes, Hermineh Sanossian, Mady Marinescu, and Anil Thomas. Malware classification with recurrent networks. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1916–1920. IEEE, 2015.

[18] Bojan Kolosnjaji, Apostolis Zarras, George Webster, and Claudia Eckert. Deep learning for classification of malware system call sequences. In *Australasian Joint Conference on Artificial Intelligence*, pages 137–149. Springer, 2016.

[19] Leandros A Maglaras, Jianmin Jiang, and Tiago J Cruz. Combining ensemble methods and social network metrics for improving accuracy of ocsvm on intrusion detection in scada systems. *Journal of Information Security and Applications*, 30: 15–26, 2016.

[20] Salima Omar, Asri Ngadi, and Hamid H Jebur. Machine learning techniques for anomaly detection: an overview. *International Journal of Computer Applications*, 79(2), 2013.

[21] YS Kong, S Abdullah, D Schramm, MZ Omar, and SM Haris. Optimization of spring fatigue life prediction model for vehicle ride using hybrid multi-layer perceptron artificial neural networks. *Mechanical Systems and Signal Processing*, 122:597–621, 2019.

[22] S Virushabadoss and C Bhuvaneswari. Analysis of behavior profiling algorithm to detect usage anomalies in fog computing. In *One Day National Conference On Internet Of Things-The Current Trend In Connected World*, 2018.

[23] Frank L Greitzer. Insider threats: It's the human, stupid! In *Proceedings of the Northwest Cybersecurity Symposium*, pages 1–8, 2019.

[24] Salima Omar, Asri Ngadi, and Hamid H Jebur. An adaptive intrusion detection model based on machine learning techniques. *International Journal of Computer Applications*, 70(7), 2013.

[25] Athul Harilal, Flavio Toffalini, John Castellanos, Juan Guarnizo, Ivan Homoliak, and Martín Ochoa. Twos: A dataset of malicious insider threat behavior based on a gamified competition. In *Proceedings of the 2017 International Workshop on Managing Insider Security Threats*, pages 45–56, 2017.

[26] Matt Bishop and Carrie Gates. Defining the insider threat. In *Proceedings of the 4th annual workshop on Cyber security and information intelligence research: developing strategies to meet the cyber security and information intelligence challenges ahead*, pages 1–3, 2008.

[27] Ameya Sanzgiri and Dipankar Dasgupta. Classification of insider threat detection techniques. In *Proceedings of the 11th annual cyber and information security research conference*, pages 1–4, 2016.

[28] Hongmei Chi, Carol Scarllet, Zornitza Genova Prodanoff, and Dominique Hubbard. Determining predisposition to insider threat activities by using text analysis. In *2016 Future Technologies Conference (FTC)*, pages 985–990. IEEE, 2016.

[29] Jianguo Jiang, Jiuming Chen, Kim-Kwang Raymond Choo, Kunying Liu, Chao Liu, Min Yu, and Prasant Mohapatra. Prediction and detection of malicious insiders' motivation based on sentiment profile on webpages and emails. In *MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM)*, pages 1–6. IEEE, 2018.

[30] Charlie Soh, Sicheng Yu, Annamalai Narayanan, Santhiya Duraisamy, and Lihui Chen. Employee profiling via aspect-based sentiment and network for insider threats detection. *Expert Systems with Applications*, 135:351–361, 2019.

[31] Duc C Le, Nur Zincir-Heywood, and Malcolm I Heywood. Analyzing data granularity levels for insider threat detection using machine learning. *IEEE Transactions on Network and Service Management*, 17(1):30–44, 2020.

[32] Naghmeh Moradpoor Sheykhkanloo and Adam Hall. Insider threat detection using supervised machine learning algorithms on an extremely imbalanced dataset. In *Journal of CyberWarfare and Terrorism (IJCWT)*, pages 1–26, 2020.

[33] Esteban Castillo, Sreekar Dhaduvai, Peng Liu, Kartik-Singh Thakur, Adam Dalton, and Tomek Strzalkowski. Email threat detection using distinct neural network approaches. In *Proceedings for the First International Workshop on Social Threats in Online Conversations: Understanding and Management*, pages 48–55, 2020.

[34] Mohamed Abdulhussain Ali Madan Maki and Suresh Subramanian. Using an artificial neural network to improve email security. In *Implementing Computational Intelligence Techniques for Security Systems Design*, pages 131–145. IGI Global, 2020.

[35] Gaoqing Yu, Wenqing Fan, Wei Huang, and Jing An. An explainable method of phishing emails generation and its application in machine learning. In *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, volume 1, pages 1279–1283. IEEE, 2020.

[36] Shuhan Yuan and Xintao Wu. Deep learning for insider threat detection: Review, challenges and opportunities. *Computers & Security*, page 102221, 2021.

[37] Duc C Le and Nur Zincir-Heywood. Anomaly detection for insider threats using unsupervised ensembles. *IEEE Transactions on Network and Service Management*, 2021.

[38] Ujwala Sav and Ganesh Magar. Insider threat detection based on anomalous behavior of user for cybersecurity. In *Data Science and Security*, pages 17–28. Springer, 2021.

[39] Sergiu Eftimie, Radu Moinescu, and Ciprian Răcuciu. Insider threat detection using natural language processing and personality profiles. In *2020 13th International Conference on Communications (COMM)*, pages 325–330. IEEE, 2020.

[40] Azamat Sultanov and Konstantin Kogos. Insider threat detection based on stress recognition using keystroke dynamics. *arXiv preprint arXiv:2005.02862*, 2020.

[41] Mathieu Garchery and Michael Granitzer. Adsage: Anomaly detection in sequences of attributed graph edges applied to insider threat detection at fine-grained level. *arXiv preprint arXiv:2007.06985*, 2020.

[42] Jari Jääskelä. Anomaly-based insider threat detection with expert feedback and descriptions. 2020.

[43] Malvika Singh, BM Mehtre, and S Sangeetha. Insider threat detection based on user behaviour analysis. In *International Conference on Machine Learning, Image Processing, Network Security and Data Sciences*, pages 559–574. Springer, 2020.

[44] Zahra Nematzadeh, Roliana Ibrahim, and Ali Selamat. Comparative studies on breast cancer classifications with k-fold cross validations using machine learning techniques. In *2015 10th Asian Control Conference (ASCC)*, pages 1–6. IEEE, 2015.

[45] R Ani, Jithu Jose, Manu Wilson, and OS Deepa. Modified rotation forest ensemble classifier for medical diagnosis in decision support systems. In *Progress in Advanced Computing and Intelligent Engineering*, pages 137–146. Springer, 2018.

[46] David A Omondiagbe, Shanmugam Veeramani, and Amandeep S Sidhu. Machine learning classification techniques for breast cancer diagnosis. In *IOP Conference Series: Materials Science and Engineering*, volume 495, page 012033. IOP Publishing, 2019.

[47] Quinlan D Buchlak, Nazanin Esmaili, Jean-Christophe Leveque, Farrokh Farrokhi, Christine Bennett, Massimo Piccardi, and Rajiv K Sethi. Machine learning applications to clinical decision support in neurosurgery: an artificial intelligence augmented systematic review. *Neurosurgical review*, pages 1–19, 2019.

[48] Abdoulaye Diop, Nahid Emad, Thierry Winter, and Mohamed Hilia. Design of an ensemble learning behavior anomaly detection framework. *International Journal of Computer and Information Engineering*, 13(10):551–559, 2019.

[49] Seok-Jun Bu and Sung-Bae Cho. A hybrid system of deep learning and learning classifier system for database intrusion detection. In *International Conference on Hybrid Artificial Intelligence Systems*, pages 615–625. Springer, 2017.

[50] Pratik Chattopadhyay, Lipo Wang, and Yap-Peng Tan. Scenario-based insider threat detection from cyber activities. *IEEE Transactions on Computational Social Systems*, 5(3):660–675, 2018.

[51] Ivan Homoliak, Flavio Toffalini, Juan Guarnizo, Yuval Elovici, and Martín Ochoa. Insight into insiders and it: A survey of insider threat taxonomies, analysis, modeling, and countermeasures. *ACM Computing Surveys (CSUR)*, 52(2):1–40, 2019.

[52] Jahanzaib Malik, Adnan Akhunzada, Iram Bibi, Muhammad Imran, Arslan Musaddiq, and Sung Won Kim. Hybrid deep learning: An efficient reconnaissance and surveillance detection mechanism in sdn. *IEEE Access*, 8:134695–134706, 2020.

[53] Atul Sajjanhar, Yong Xiang, et al. Image-based feature representation for insider threat classification. *arXiv preprint arXiv:1911.05879*, 2019.

[54] D Karthikeyan, V Mohanraj, Y Suresh, and J Senthilkumar. An efficient stacking model with srpf classifier technique for intrusion detection system. *International Journal of Communication Systems*, 34(10):e4737, 2021.

[55] Prabhav Gupta, Yash Ghatole, and Nihal Reddy. Stacked autoencoder based intrusion detection system using one-class classification. In *2021 11th International*

Conference on Cloud Computing, Data Science & Engineering (Confluence), pages 643–648. IEEE, 2021.

[56] Alexander Liu, Cheryl E Martin, Tom Hetherington, and Sara Matzner. Ai lessons learned from experiments in insider threat detection. In *AAAI Spring Symposium: What Went Wrong and Why: Lessons from AI Research and Applications*, pages 49–55, 2006.

[57] Abdeljalil Agnaou, Anas Abou El Kalam, Abdellah Ait Ouahman, and Mina De Montfort. Automated technique to reduce positive and negative false from attacks collected through the deployment of distributed honeypot network. *International Journal of Computer Science and Information Security*, 14(9):494, 2016.

[58] Xianbo Dai, Na Wang, and Wenjuan Wang. Application of machine learning in bgp anomaly detection. In *Journal of Physics: Conference Series*, volume 1176, page 032015. IOP Publishing, 2019.

[59] Hamed HaddadPajouh, Ali Dehghantanha, Raouf Khayami, and Kim-Kwang Raymond Choo. A deep recurrent neural network based approach for internet of things malware threat hunting. *Future Generation Computer Systems*, 85:88–96, 2018.

[60] Aaron Tuor, Samuel Kaplan, Brian Hutchinson, Nicole Nichols, and Sean Robinson. Deep learning for unsupervised insider threat detection in structured cybersecurity data streams. *arXiv preprint arXiv:1710.00811*, 2017.

[61] Syam Akhil Repalle and Venkata Ratnam Kolluru. Intrusion detection system using ai and machine learning algorithm. *International Research Journal of Engineering and Technology (IRJET)*, 4(12):1709–1715, 2017.

[62] Fangfang Yuan, Yanan Cao, Yanmin Shang, Yanbing Liu, Jianlong Tan, and Binxing Fang. Insider threat detection with deep neural network. In *International Conference on Computational Science*, pages 43–54. Springer, 2018.

[63] Owen Lo, William J Buchanan, Paul Griffiths, and Richard Macfarlane. Distance measurement methods for improved insider threat detection. *Security and Communication Networks*, 2018, 2018.

[64] Duc C Le and A Nur Zincir-Heywood. Machine learning based insider threat modelling and detection. In *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pages 1–6. IEEE, 2019.

[65] Jiuming Lu and Raymond K Wong. Insider threat detection with long short-term memory. In *Proceedings of the Australasian Computer Science Week Multiconference*, pages 1–10, 2019.

[66] Teng Hu, Weina Niu, Xiaosong Zhang, Xiaolei Liu, Jiazhong Lu, and Yuan Liu. An insider threat detection approach based on mouse dynamics and deep learning. *Security and Communication Networks*, 2019, 2019.

[67] Junhong Kim, Minsik Park, Haedong Kim, Suhyoun Cho, and Pilsung Kang. Insider threat detection based on user behavior modeling and anomaly detection algorithms. *Applied Sciences*, 9(19):4018, 2019.

[68] Malvika Singh, Babu M Mehtre, and S Sangeetha. User behavior profiling using ensemble approach for insider threat detection. In *2019 IEEE 5th International Conference on Identity, Security, and Behavior Analysis (ISBA)*, pages 1–8. IEEE, 2019.

[69] Lidong Wang and Randy Jones. Big data analytics in cyber security: Network traffic and attacks. *Journal of Computer Information Systems*, pages 1–8, 2020.

[70] Mohammad Rasool Fatemi and Ali A Ghorbani. Threat hunting in windows using big security log data. In *Security, Privacy, and Forensics Issues in Big Data*, pages 168–188. IGI Global, 2020.

[71] Hossein Hassani, Christina Beneki, Stephan Unger, Maedeh Taj Mazinani, and Mohammad Reza Yeganegi. Text mining in big data analytics. *Big Data and Cognitive Computing*, 4(1):1, 2020.

[72] Ravdeep Kour, Adithya Thaduri, and Ramin Karim. Railway defender kill chain to predict and detect cyber-attacks. *Journal of Cyber Security and Mobility*, pages 47–90, 2020.

[73] Jorge Maestre Vidal, Marco Antonio Sotelo Monge, and Sergio Mauricio Martínez Monterrubio. Espada: enhanced payload analyzer for malware detection robust

against adversarial threats. *Future Generation Computer Systems*, 104:159–173, 2020.

[74] Mohammed Nasser Al-Mhiqani, Rabiah Ahmed, Z Zainal Abidin, and SN Isnin. An integrated imbalanced learning and deep neural network model for insider threat detection.

[75] Liu Liu, Olivier De Vel, Qing-Long Han, Jun Zhang, and Yang Xiang. Detecting and preventing cyber insider threats: A survey. *IEEE Communications Surveys & Tutorials*, 20(2):1397–1417, 2018.

[76] Keshnee Padayachee. An assessment of opportunity-reducing techniques in information security: An insider threat perspective. *Decision Support Systems*, 92: 47–56, 2016.

[77] Philip A Legg, Nick Moffat, Jason RC Nurse, Jassim Happa, Ioannis Agrafiotis, Michael Goldsmith, and Sadie Creese. Towards a conceptual model and reasoning structure for insider threat detection. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, 4(4):20–37, 2013.

[78] Monowar H Bhuyan, Dhruba Kumar Bhattacharyya, and Jugal K Kalita. Network anomaly detection: methods, systems and tools. *Ieee communications surveys & tutorials*, 16(1):303–336, 2013.

[79] Gaurang Gavai, Kumar Sricharan, Dave Gunning, Rob Rolleston, John Hanley, and Mudita Singhal. Detecting insider threat from enterprise social and online activity data. In *Proceedings of the 7th ACM CCS international workshop on managing insider security threats*, pages 13–20, 2015.

[80] Weizhi Meng, Kim-Kwang Raymond Choo, Steven Furnell, Athanasios V Vasilakos, and Christian W Probst. Towards bayesian-based trust management for insider attacks in healthcare software-defined networks. *IEEE Transactions on Network and Service Management*, 15(2):761–773, 2018.

[81] Michael Mayhew, Michael Atighetchi, Aaron Adler, and Rachel Greenstadt. Use of machine learning in big data analytics for insider threat detection. In *MILCOM 2015-2015 IEEE Military Communications Conference*, pages 915–922. IEEE, 2015.

[82] Duc C Le and A Nur Zincir-Heywood. Evaluating insider threat detection workflow using supervised and unsupervised learning. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 270–275. IEEE, 2018.

[83] Duc C Le and Nur Zincir-Heywood. Exploring adversarial properties of insider threat detection. In *2020 IEEE Conference on Communications and Network Security (CNS)*, pages 1–9. IEEE, 2020.

[84] Pedro Ferreira, Duc C Le, and Nur Zincir-Heywood. Exploring feature normalization and temporal information for machine learning based insider threat detection. In *2019 15th International Conference on Network and Service Management (CNSM)*, pages 1–7. IEEE, 2019.

[85] Jason Matterer and Daniel LeJeune. Peer group metadata-informed lstm ensembles for insider threat detection. In *The Thirty-First International Flairs Conference*, 2018.

[86] Muhammad Nabeel Asim, Muhammad Wasim, Muhammad Sajid Ali, and Abdur Rehman. Comparison of feature selection methods in text classification on highly skewed datasets. In *2017 First International Conference on Latest trends in Electrical Engineering and Computing Technologies (INTELLECT)*, pages 1–8. IEEE, 2017.

[87] Tarannum Zaki, Md Sami Uddin, Md Mahedi Hasan, and Muhammad Nazrul Islam. Security threats for big data: A study on enron e-mail dataset. In *2017 international conference on research and innovation in information systems (icriis)*, pages 1–6. IEEE, 2017.

[88] Lei Shi, Qiang Wang, Xinming Ma, Mei Weng, and Hongbo Qiao. Spam email classification using decision tree ensemble. *Journal of Computational Information Systems*, 8(3):949–956, 2012.

[89] Nancy Leong. *Identity Capitalists: The Powerful Insiders Who Exploit Diversity to Maintain Inequality*. Stanford University Press, 2021.

[90] Malek Ben Salem, Shlomo Hershkop, and Salvatore J Stolfo. A survey of insider attack detection research. *Insider Attack and Cyber Security*, pages 69–90, 2008.

[91] Giuseppe Manco, Elio Masciari, Massimo Ruffolo, and Andrea Tagarelli. Towards an adaptive mail classifier. In *Proc. of Italian Association for Artificial Intelligence Workshop*, 2002.

[92] William W Cohen et al. Learning rules that classify e-mail. In *AAAI spring symposium on machine learning in information access*, volume 18, page 25. Stanford, CA, 1996.

[93] Zhangdong Wang, Jiaohua Qin, Xuyu Xiang, and Yun Tan. A privacy-preserving and traitor tracking content-based image retrieval scheme in cloud computing. *Multimedia Systems*, pages 1–13, 2021.

[94] Farhan Asif Chowdhury, Dheeman Saha, Md Rashidul Hasan, Koustuv Saha, and Abdullah Mueen. Examining factors associated with twitter account suspension following the 2020 us presidential election. *arXiv preprint arXiv:2101.09575*, 2021.

[95] Lakshit Malhotra, Bharat Bhushan, and Rahul Veer Singh. Artificial intelligence and deep learning-based solutions to enhance cyber security. *Available at SSRN 3833311*, 2021.

[96] Han-Sung Kim and Sung-Deok CHA. Efficient masquerade detection using svm based on common command frequency in sliding windows. *IEICE TRANSACTIONS on Information and Systems*, 87(11):2446–2452, 2004.

[97] Insider threat test dataset. `https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=508099`, . Accessed: 2020-07-30.

[98] Alan Lukezic, Tomas Vojir, Luka ˇCehovin Zajc, Jiri Matas, and Matej Kristan. Discriminative correlation filter with channel and spatial reliability. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6309–6318, 2017.

[99] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[100] C.M. Bishop. Variational principal components. *Journal of Artificial Neural Networks*, 9:509–514, 1999.

[101] Andrea Bommert, Xudong Sun, Bernd Bischl, Jörg Rahnenführer, and Michel Lang. Benchmark for filter methods for feature selection in high-dimensional classification data. *Computational Statistics & Data Analysis*, 143:106839, 2020.

[102] David H Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992.

[103] João Ferreira, Hugo Gonçalo Oliveira, and Ricardo Rodrigues. Improving nltk for processing portuguese. In *8th Symposium on Languages, Applications and Technologies (SLATE 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.

[104] Aurélien Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, 2019.

[105] Ekaba Bisong. Matplotlib and seaborn. In *Building machine learning and deep learning models on google cloud platform*, pages 151–165. Springer, 2019.

[106] Quentin F Gronau and Eric-Jan Wagenmakers. Limitations of bayesian leave-one-out cross-validation for model selection. *Computational brain & behavior*, 2(1): 1–11, 2019.

[107] Tzu-Tsung Wong and Po-Yang Yeh. Reliable accuracy estimates from k-fold cross validation. *IEEE Transactions on Knowledge and Data Engineering*, 32(8):1586–1594, 2019.

[108] Alok Kumar Dwivedi, Indika Mallawaarachchi, and Luis A Alvarado. Analysis of small sample size studies using nonparametric bootstrap test with pooled resampling method. *Statistics in medicine*, 36(14):2187–2205, 2017.

[109] Mahdieh Labani, Parham Moradi, Fardin Ahmadizar, and Mahdi Jalili. A novel multivariate filter method for feature selection in text classification problems. *Engineering Applications of Artificial Intelligence*, 70:25–37, 2018.

[110] Jesús González, Julio Ortega, Miguel Damas, Pedro Martín-Smith, and John Q Gan. A new multi-objective wrapper method for feature selection–accuracy and stability analysis for bci. *Neurocomputing*, 333:407–418, 2019.

[111] Amalia Luque, Alejandro Carrasco, Alejandro Martín, and Ana de las Heras. The impact of class imbalance in classification performance metrics based on the binary confusion matrix. *Pattern Recognition*, 91:216–231, 2019.

[112] Tianyu Du, Shouling Ji, Lujia Shen, Yao Zhang, Jinfeng Li, Jie Shi, Chengfang Fang, Jianwei Yin, Raheem Beyah, and Ting Wang. Cert-rnn: Towards certifying the robustness of recurrent neural networks. 2021.

[113] Filip Wieslaw Bartoszewski, Mike Just, Michael A Lones, and Oleksii Mandrychenko. Anomaly detection for insider threats: An objective comparison of machine learning models and ensembles. In *IFIP International Conference on ICT Systems Security and Privacy Protection*, pages 367–381. Springer, 2021.

[114] Alpesh M Patel and Anil Suthar. Adaboosted extra trees classifier for object-based multispectral image classification of urban fringe area. *International Journal of Image and Graphics*, page 2140006, 2020.

[115] BV Padmaja, V Prasa, and KVN Sunitha. A novel random split point procedure using extremely randomized (extra) trees ensemble method for human activity recognition. *EAI Endorsed Transactions on Pervasive Health and Technology*, 6 (22):e5, 2020.

[116] Nassim Bessaad, Qilian Bao, Shuodong Sun, Yuding Du, Lin Liu, and Mahmood Ul Hassan. Adaptive dual wavelet threshold denoising function combined with allan variance for tuning fog-sins filter. *Journal of Shanghai Jiaotong University (Science)*, 25(4):434–440, 2020.

[117] Xibin Dong, Zhiwen Yu, Wenming Cao, Yifan Shi, and Qianli Ma. A survey on ensemble learning. *Frontiers of Computer Science*, 14(2):241–258, 2020.

[118] Mohanad Abd Shehab and Nihan Kahraman. A weighted voting ensemble of efficient regularized extreme learning machine. *Computers & Electrical Engineering*, 85:106639, 2020.

[119] Cert dataset r6.2. `https://kilthub.cmu.edu/articles/dataset/Insider_Threat_Test_Dataset/12841247/1?file=24844280`, . Accessed: 2020-07-30.

[120] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss,

Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[121] Irina Rish et al. An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46, 2001.

[122] Alexey Natekin and Alois Knoll. Gradient boosting machines, a tutorial. *Frontiers in neurorobotics*, 7:21, 2013.

[123] Mahesh Pal. Random forest classifier for remote sensing classification. *International journal of remote sensing*, 26(1):217–222, 2005.

[124] Anna L Buczak and Erhan Guven. A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications surveys & tutorials*, 18(2):1153–1176, 2015.

[125] Junhong Kim, Minsik Park, Haedong Kim, Suhyoun Cho, and Pilsung Kang. Insider threat detection based on user behavior modeling and anomaly detection algorithms. *Applied Sciences*, 9(19):4018, 2019.

[126] Enron dataset. `http://www-2.cs.cmu.edu/enron`, . Accessed: 2020-08-04.

[127] Hosagrahar V Jagadish, Beng Chin Ooi, Kian-Lee Tan, Cui Yu, and Rui Zhang. idistance: An adaptive b+-tree based indexing method for nearest neighbor search. *ACM Transactions on Database Systems (TODS)*, 30(2):364–397, 2005.

[128] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[129] Shenglei Chen, Geoffrey I Webb, Linyuan Liu, and Xin Ma. A novel selective naïve bayes algorithm. *Knowledge-Based Systems*, 192:105361, 2020.

[130] Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multiclass support vector machines. *IEEE transactions on Neural Networks*, 13(2):415–425, 2002.

[131] George AF Seber and Alan J Lee. *Linear regression analysis*, volume 329. John Wiley & Sons, 2012.

[132] Sai Li, Huajing Fang, and Xiaoyong Liu. Parameter optimization of support vector regression based on sine cosine algorithm. *Expert systems with Applications*, 91: 63–77, 2018.

[133] Adam Coates and Andrew Y Ng. Learning feature representations with k-means. In *Neural networks: Tricks of the trade*, pages 561–580. Springer, 2012.

[134] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521 (7553):436–444, 2015.

[135] Anton Vladyka and Tim Albrecht. Unsupervised classification of single-molecule data with autoencoders and transfer learning. *Machine Learning: Science and Technology*, 1(3):035013, 2020.

[136] Jashanjot Kaur and P Kaur Buttar. A systematic review on stopword removal algorithms. *Int. J. Futur. Revolut. Comput. Sci. Commun. Eng*, 4(4), 2018.

[137] Safaa I Hajeer, Rasha M Ismail, Nagwa L Badr, and Mohamed Fahmy Tolba. A new stemming algorithm for efficient information retrieval systems and web search engines. In *Multimedia Forensics and Security*, pages 117–135. Springer, 2017.

[138] Tianrui Peng, Ian Harris, and Yuki Sawa. Detecting phishing attacks using natural language processing and machine learning. In *2018 ieee 12th international conference on semantic computing (icsc)*, pages 300–301. IEEE, 2018.

[139] Aydin Farrokhi, Farid Shirazi, Nick Hajli, and Mina Tajvidi. Using artificial intelligence to detect crisis related to events: Decision making in b2b by artificial intelligence. *Industrial Marketing Management*, 91:257–273, 2020.

[140] Robert E Schapire. Explaining adaboost. In *Empirical inference*, pages 37–52. Springer, 2013.

[141] Michael W Kenyhercz and Nicholas V Passalacqua. Missing data imputation methods and their performance with biodistance analyses. In *Biological Distance Analysis*, pages 181–194. Elsevier, 2016.

[142] SF Ding, BJ Qi, and HY Tan. An overview on theory and algorithm of support vector machines. *Journal of University of Electronic Science and Technology of China*, 40(1):2–10, 2011.

[143] Cong Zhou, J Geoffrey Chase, and Geoffrey W Rodgers. Support vector machines for automated modelling of nonlinear structures using health monitoring results. *Mechanical Systems and Signal Processing*, 149:107201, 2021.

[144] Chao-Ying Joanne Peng, Kuk Lida Lee, and Gary M Ingersoll. An introduction to logistic regression analysis and reporting. *The journal of educational research*, 96(1):3–14, 2002.

[145] Armin Askari, Alexandre d'Aspremont, and Laurent El Ghaoui. Naive feature selection: Sparsity in naive bayes. In *International Conference on Artificial Intelligence and Statistics*, pages 1813–1822. PMLR, 2020.

[146] Won Park, Youngin You, and Kyungho Lee. Detecting potential insider threat: Analyzing insiders' sentiment exposed in social media. *Journal of Security and Communication Networks*, 2018, 2018.

[147] Athul Harilal, Flavio Toffalini, Ivan Homoliak, John Henry Castellanos, Juan Guarnizo, Soumik Mondal, and Martín Ochoa. The wolf of sutd (twos): A dataset of malicious insider threat behavior based on a gamified competition. *J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl.*, 9(1):54–85, 2018.

[148] Jaideepsinh K Raulji and Jatinderkumar R Saini. Stop-word removal algorithm and its implementation for sanskrit language. *International Journal of Computer Applications*, 150(2):15–17, 2016.

[149] Jasmeet Singh and Vishal Gupta. A systematic review of text stemming techniques. *Artificial Intelligence Review*, 48(2):157–217, 2017.

[150] S Vijayarani and R Janani. Text mining: open source tokenization tools - an analysis. *Advanced Computational Intelligence: An International Journal (ACII)*, 3(1):37–47, 2016.