SUPERVISED LEARNING FOR INSIDER THREAT DETECTION

Phavithra Manoharan

Thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

Victoria University, Australia

Institute for Sustainable Industries and Liveable Cities

April 2024

Abstract

Cyberattacks cause havoc in the digital world, but the most significant threat might be from those who appear to be trustworthy: insiders. Insider threats pose a significant and evolving challenge to organisations, jeopardizing data security, operational processes, and overall well-being. Unlike external threats, these threats stem from individuals with authorized access and deep familiarity with internal systems, making them particularly difficult to detect and potentially causing more substantial damage. Insiders, including employees, contractors, or business partners, possess legitimate access to a company's systems and data. When these insiders act maliciously or negligently, they can cause significant damage through theft, sabotage, or espionage. While robust for detecting and preventing insider threats, machine learning and deep learning techniques face several challenges. This thesis aims to highlight three significant challenges in insider threat detection and prediction.

A significant challenge in evaluating insider threat detection and prediction algorithms is the lack of standardized datasets and problem settings. This variability makes it difficult to compare the effectiveness of different approaches and provide clear recommendations for decision-makers. To address this challenge, this study aims to objectively evaluate the performance of supervised machine learning algorithms within a consistent experimental setting. This will be achieved by implementing supervised algorithms using the balanced CERT r4.2 dataset, employing a uniform feature extraction methodology. The performance of various supervised machine learning algorithms on a balanced dataset using the same feature extraction method is thoroughly evaluated. Additionally, an exploration of the impact of hyperparameter tuning on performance within the balanced dataset

is conducted.

The second challenge is, traditionally, detecting insider threats, which involves analyzing user behaviours recorded in logs and developing a binary classifier to differentiate between malicious and non-malicious individuals. However, existing approaches only consider either standalone activities or sequential activities. A novel approach is proposed to enhance the detection of malicious insiders: a bilateral insider threat detection method that harnesses the power of recurrent neural networks and incorporates both standalone and sequential activities. Initially, behavioural characteristics are extracted from log files, representing the standalone activities. Then, RNN models are utilized to capture the features that represent sequential activities. Subsequently, the features obtained from standalone and sequential activities are merged, and a binary classification model is employed to detect insider threats effectively. The experiment findings using the publicly available CERT r4.2 dataset demonstrate that the proposed bilateral insider threat detection approach significantly improves the performance of insider threat detection.

The third challenge is that previous research has addressed the challenge by pinpointing malicious actions that have already occurred but they have provided limited assistance in preventing these risks. This research introduces a novel approach based on bidirectional long-term memory networks, aiming to effectively capture and analyse the patterns of individual actions and their sequential dependencies. The focus lies in predicting whether an individual will become a malicious insider in the future based on their daily behavioural records over the preceding several days. The performance of the four supervised learning algorithms on manual features, sequential features, and the ground truth of the day with various combinations is analysed. Additionally, the performance of different RNN models, such as RNN, LSTM, and BiLSTM, in incorporating these features is investigated. Moreover, the performance of different predictive lengths on the ground truth of the day and different embedded lengths for the sequential features is explored. All experiments are conducted on the CERT r4.2 dataset, with experiment results indicating that BiLSTM achieves the highest performance in combining these features.

In summary, this research can effectively address three significant challenges in insider threat detection and prediction.

Doctor of Philosophy Declaration

"I, Phavithra Manoharan, declare that the PhD thesis entitled *Supervised Learning for Insider Threat Detection* is no more than 80,000 words in length including quotes and exclusive of tables, figures, appendices, bibliography, references and footnotes. This thesis contains no material that has been submitted previously, in whole or in part, for the award of any other academic degree or diploma. Except where otherwise indicated, this thesis is my own work". "I have conducted my research in alignment with the Australian Code for the Responsible Conduct of Research and Victoria University's Higher Degree by Research Policy and Procedures.

Signature: Date: 22/04/2024

Acknowledgements

This thesis would not have been possible without the invaluable support of many individuals. I am particularly grateful to Professor Hua Wang, whose guidance and mentorship have been instrumental throughout my doctoral studies. Professor Wang, I am deeply appreciative of your willingness to share your knowledge, your unwavering support, and your constant kindness, understanding, and patience. Thank you for granting me the opportunity to delve into my ideas and pursue the specific research topic I have longed to explore. Your trust in me was instrumental in shaping this project. Furthermore, your financial support through the ISILC Fee Sponsorship Offer for my Ph.D. studies significantly eased the burden of this journey. Your guidance also proved invaluable in crafting and submitting my research papers, allowing me to share my findings at conferences and in esteemed journals. Over the years, your willingness to share both excellent professional advice and genuine personal support has been a constant source of strength. Through times of sadness and happiness, failure and success, you've been a steady presence, and your belief in me has played a significant role in shaping who I am today.

I am particularly grateful to my associate supervisor, Dr. Jiao Yin, for your kind support, valuable advice, and encouragement throughout my Ph.D. candidature. Your contributions, even though they came later in my journey, have been immeasurable. The regular weekly meetings and your willingness to correct my work were invaluable for my progress. Thank you for being such a kind mentor – your support has been more than I could have asked for. I am also grateful to Prof. Yanchun Zhang for your support and enlightenment during our group meetings. I would also like to extend my thanks to Dr. Wenjie Ye for your

invaluable advice and encouragement, particularly during the critical stages of my research.

I would like to thank my other PhD friends, Wei Hong and Kejing Du, for sharing their knowledge, suggesting ideas, and offering guidance in the insider threat detection field. I also want to thank my friends Dr Sudha Subramani, Dr Shekha Chenthara, Dr Rubina Sarki, and Samsad for their support.

Without expressing my deepest gratitude to my husband, Dr Sarathkumar Rangarajan, I would be remiss. His unwavering encouragement ignited my desire to pursue this Ph.D. journey. Throughout its course, his steadfast support was a constant source of strength, particularly during the challenging moments that both life and my research presented. With his unwavering belief in me, I was able to persevere and reach this significant milestone. Thank you for everything, my love.

I am incredibly fortunate to have the unwavering support of my family. To my appa, Manoharan, my deepest gratitude goes out to you. Your constant love and encouragement have been a strength throughout my life, especially during this challenging journey. I also want to express my heartfelt thanks to my younger sister, Dr Devasudha Manoharan. I am grateful for her support and guidance throughout this process.

I would also like to extend my heartfelt gratitude to my extended family for their unwavering support throughout my PhD journey. To my uncle, Rangarajan, and my aunt, Thangamani, your constant encouragement and well wishes meant the world to me. A special thank you goes to my sister-in-law, Janu. Your willingness to help care for my baby girl during the pandemic was invaluable. Knowing my daughter was in loving hands allowed me to focus on my research with greater peace of mind.

Last but not least, to my amazing girl Niralya, who is now a whole five years old! Words can't express my love and gratitude. Even though my PhD journey meant missing some cuddles during your toddler years, your bright smile and infectious laughter always brought sunshine into my world. Thank you for being so understanding, even if you couldn't quite grasp it then. This achievement is for both of us, and I promise to make it up to you with all the love, playtime, and adventures you deserve!

Dedication

In loving memory of Padma Manoharan, my mother, whose endless love and support would have fueled every step of this journey. Though you are no longer with me, your essence remains a constant source of inspiration. This thesis is dedicated to you, Amma.

Publications

This thesis includes work by the author that has been published or accepted for publication. These publications are the own work of the author of this thesis, and the author has the permission of the publishers to reproduce the contents of these publications for academic purposes.

In particular, some data, ideas, opinions and figures presented in this thesis have previously appeared or may appear shortly after the submission of this thesis as follows:

Publications:

- **Manoharan, P.,** Yin, J., Wang, H., Zhang, Y., & Ye, W. (2023). Insider threat detection using supervised machine learning algorithms. Telecommunication Systems, 1-17.

- **Manoharan, P.,** Hong, W., Yin, J., Zhang, Y., Ye, W., & Ma, J. (2023, October). Bilateral Insider Threat Detection: Harnessing Standalone and Sequential Activities with Recurrent Neural Networks. In International Conference on Web Information Systems Engineering (pp. 179-188). Singapore: Springer Nature Singapore.

- **Manoharan, P.,**Yin, J., Wang, H., Zhang, Y., Ye, W. (2024, August) Insider Threat Detection: A Review. International Conference on Networking and Network Applications. Yinchuan City, China. (Accepted).

- **Manoharan, P.,** Hong, W., Yin, J., Wang, H., Zhang, Y., Ye, W., Optimising Insider Threat Prediction: Exploring BiLSTM Networks and Sequential Features. Data Science and Engineering. (Under Review)

Contents

List of Figures

List of Tables

Glossary

AdB AdaBoost

AI Artificial Intelligence

ANN Artificial Neural Network

Bi-LSTM Bi-directional Long Short-Term Memory

BRITD Behaviour Rhythm Insider Threat Detection

CERT/CC CERT Coordination Center

CISA Cyber and Infrastructure Security Agency

CNN Convolutional Neural Network

DBN Deep Belief Network

DL Deep Learning

DLP Data Loss Prevention

DNN Deep Neural Network

DoS Denial-of-Service

DT Decision Tree

FERC Federal Energy Regulatory Commission

FNN Feedforward Neural Network

GCN Graph Convolutional Networks

GNB Gaussian Naive Bayes

GNN Graph Neural Network

GRU Gated Recurrent Unit

HMMs Hidden Markov Models

IF Isolation Forest

IHMM Improved Hidden Markov Models

IP Intellectual Property

IT Information Technology

KNN K-Nearest Neighbors

LAC LSTM Autoencoder with Community

LDA Linear Discriminant Analysis

LDAP Lightweight Directory Access Protocol

LR Logistic Regression

LSTM Long Short-Term Memory

ML Machine Learning

MLP multiLayer Perceptron

NB Naive Bayes

NLP Natural Language Processing

OCC One-Class Classification

OCSVM One-Class Support Vector Machines

PAM Privileged Access Management

PCA Principal Component Analysis

PDF Probability Density Function

PP Psychological Profiling

QDA Quadratic Discriminant Analysis

RBAC Role-Based Access Control

RDBMS Relational Database Management Systems

RF Random Forest

RNN Recurrent Neural Network

RUU Are You You

SA Structural Anomaly detection

SGD Stochastic Gradient Descent

SIEM Security Information and Event Management

SVM Support Vector Machines

TWOS The Wolf of SUTD

UAG User Action Graph

UAM User Activity Monitoring

UBA User Behaviour Analysis

UEBA User and Entity Behaviour Analytics

XGBoost Extreme Gradient Boosting

Chapter 1

Introduction

1.1 Background

In today's hyper-connected world, the pervasive spectre of cyber threats casts a formidable shadow across the entire digital landscape. These threats encompass various malicious activities and vulnerabilities, posing an omnipresent and multifaceted risk to individuals, organisations, and nations. Insider threats emerge as a nuanced and distinctive category, casting a shadow within the organisation's walls. As we delve into the broader sphere of cyber threats, we inevitably arrive at insider threats, where the lines between friend and foe blur, and the risks are often concealed in plain sight.

Recent data breaches and system sabotage activities that have seriously affected users worldwide have brought cyber security into greater prominence [1, 2, 3, 4]. For example, Australians lost a staggering A$13,885,099 to threats and extortion scams in 2023, according to Scamwatch [5]. These incidents serve as compelling reminders of the urgent need to prioritize and strengthen cyber security measures to safeguard sensitive information and protect against malicious threats [6, 7, 8, 9].

Insider threats pose a significant challenge to cyber security in contemporary times. Such a threat can manifest in various forms of malicious activity, including exploiting security privileges to pilfer intellectual property, divulging or trading

customer data, or deploying malware and backdoors on corporate computers. This constitutes insider misconduct. Insider threats are more vulnerable than outsider threat attacks, and while rare, they can cause significant damage [10, 11, 12, 13].

A recent report by Ponemon Institute (2022) paints a concerning picture of insider threats [14]. The frequency of these incidents is on the rise, with 67% of organisations experiencing between 21 and more than 40 insider attacks annually. This represents a significant jump from 60% in 2020 and 53% in 2018. Furthermore, the cost associated with each incident has risen dramatically. According to the same report, the average cost per incident now stands at a staggering $15.38 million [14]. This highlights the severe financial impact insider threats can have on organisations. The report also reveals that the time to detect and contain insider attacks remains a challenging task. On average, it takes organisations 77 days to address them, with data loss occurring in 42% of cases before containment [14].

A separate survey found that privileged users pose the most significant insider threat risk, with 55% of organisations identifying them as a concern[1]. This underscores the importance of implementing robust monitoring of privileged accounts closely. Moreover, it confirms insider threats' rise, reporting a 47% increase over the past two years.

The 2023 Insider Threat Report by Gurucul reveals a significant increase in insider threats, with a staggering 74% of organisations reporting a rise in the frequency of such attacks. This pervasive risk is further emphasized by the finding that over half (more than 50%) of organisations have experienced insider threats in the past year, with a concerning 8% facing more than 20 incidents. As organisations transition to hybrid work models, a significant portion (68%) express concern about insider risk. This growing concern and the report's finding that most organisations (approximately 74%) are considered moderately or highly vulnerable to insider threats, underscores the critical need for robust security measures to safeguard sensitive data and IT infrastructure [15, 16].

[1]Source: https://techjury.net/blog/insider-threat-statistics/#gref

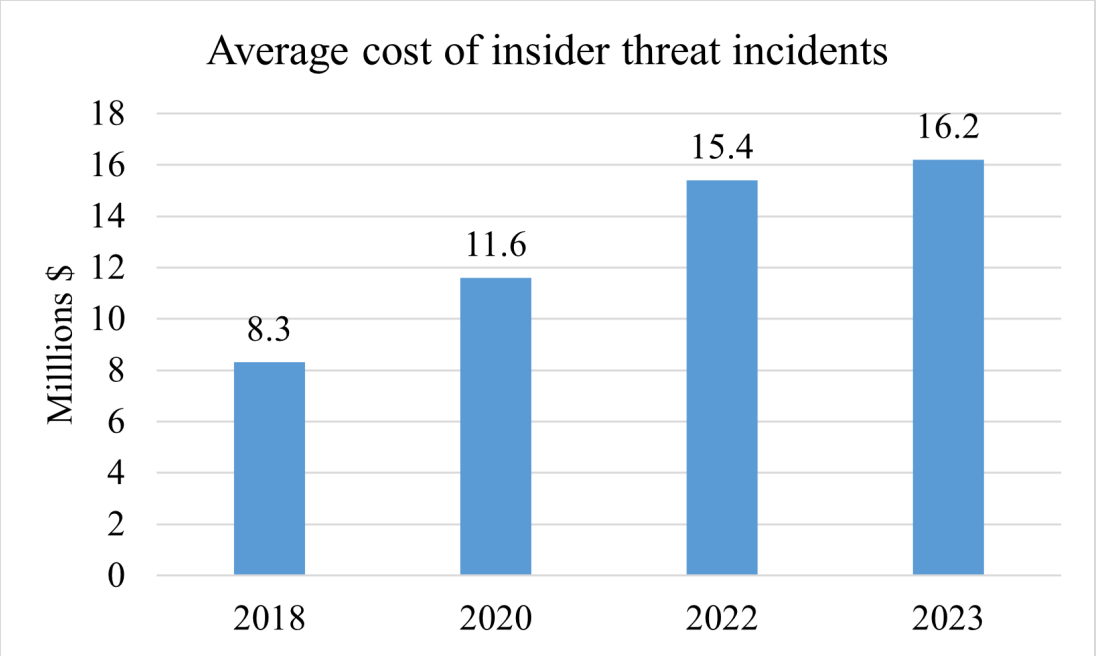


Fig. 1.1. Average cost of insider threat incidents

The COVID-19 pandemic has further amplified the problem. The shift to remote work and increased reliance on digital platforms have created new opportunities for malicious insiders to exploit vulnerabilities [17]. organisations must remain vigilant in the face of this evolving threat landscape. Implementing robust security measures, including user monitoring, access controls, and data loss prevention techniques, is crucial to detect and prevent insider attacks [18, 19, 20].

Insider threats can inflict a financial triple whammy on organisations. Direct costs encompass the immediate spending required to identify, contain, investigate, and recover from the incident. These are followed by indirect costs, representing the value of lost employee productivity and resources dedicated to managing the fallout. Finally, there are lost opportunity costs, reflecting potential profits forfeited due to the disruption caused by the attack. The Ponemon Institute's 2023 report reveals a troubling trend – the total cost of insider threats has surged nearly 95% since 2018, highlighting the critical need for robust preventative measures [21].

From Figure 1.1, the average cost of insider threat incidents in the United

States has steadily increased since 2018. The cost in 2018 was $8.3 million, and it rose to a staggering $18.33 million in 2023 [21]. This significant increase highlights insider threats' growing financial risk to organisations.

1.1.1 Insider Threats Across Industries

The threat of insider attacks isn't limited to a single industry. Malicious actors with authorized access can pose significant risks in various sectors. Here are some examples:

- Financial Industry: Employees with access to sensitive customer data, like account numbers or credit card information, could steal and sell it on the black market for fraud.
- Technology Industry: Disgruntled employees with knowledge of a company's intellectual property, such as trade secrets or software code, could leak or misuse that information to harm the organisation or give themselves or another company a competitive edge.
- Healthcare Industry: Employees with access to patient records might intentionally disclose or sell confidential medical information for personal gain. This could involve selling patient data to pharmaceutical companies or identity thieves.
- Government Sector: Insiders with access to classified information could leak sensitive data, such as national security secrets, to unauthorized individuals or foreign entities. This could be done for personal gain, ideological reasons, or even blackmail.
- Retail Industry: Employees with access to inventory management systems could manipulate data to steal valuable merchandise or sell product information to competitors.
- Energy and Utilities: Insiders with access to control systems could disrupt critical infrastructure in this industry. For example, a disgruntled employee at a power plant could manipulate controls to cause a blackout.

- Media and Entertainment: Employees with access to sensitive or unreleased content could leak it to the public before its intended release date. Additionally, insider access to celebrity or customer data could be sold to third parties for malicious purposes.

1.1.2 Recent Insider Threats in world-wide

We will examine a few recent cases of insider threats involving data breaches.

In the 2019 Capital One breach, a former Amazon employee who participated in the attack was convicted in 2022 [1]. At the beginning of the COVID-19 pandemic, a disgruntled ex-employee from a medical packing company misused a previously established admin account. They created a fraudulent new user account and manipulated thousands of files to disrupt or halt the delivery of personal protective equipment to hospitals and healthcare providers [2].

Following the breach, Tesla promptly mitigated the damage and bolstered their security systems. Collaborating closely with law enforcement, they pinpointed the two former employees responsible. Affected staff were promptly notified and provided with resources to safeguard personal information. Tesla comprehensively reviewed their IT security and data protection policies, identifying and addressing potential vulnerabilities. They instituted additional security measures, including stricter access controls, heightened user activity monitoring, and enhanced encryption protocols. Moreover, Tesla underscored the significance of employee training in cybersecurity best practices. These proactive steps demonstrate Tesla's commitment to fortifying its defences and safeguarding sensitive information [22].

In 2022, Yahoo sued a former research scientist who stole proprietary source code about their AdLearn product [3]. Minutes after receiving a job offer from a competitor, the employee downloaded approximately 570,000 pages of Yahoo's

[1] https://firewalltimes.com/recent-data-breaches/

[2] https://www.justice.gov/usao-ndga/pr/former-employee-medical-packaging-company-allegedly-sabotages-electronic-shipping

[3] https://www.thedrum.com/news/2022/05/19/yahoo-lawsuit-alleges-employee-stole-trade-secrets-upon-receiving-trade-desk-job

intellectual property (IP) to his personal devices, knowing that the information could benefit him in his new job. In the lawsuit, Yahoo claimed the stolen data would give competitors an immense advantage. Furthermore, in 2022, Microsoft employees accidentally exposed login credentials on GitHub, potentially granting access to Azure servers and other internal systems [23]. Fortunately, this leak, which could have included source code, was discovered by a security firm before exploitation. The incident highlights the risk of unintentional insider threats and the potential for hefty fines under regulations like GDPR.

In November 2021, a security breach at South Georgia Medical Center exposed sensitive patient information [1]. A disgruntled former employee, with legitimate access even after quitting, downloaded private data, including test results, names, and birth dates, onto a USB drive. This incident highlights the risk of insider threats motivated by personal motives. While the medical centre's security software eventually detected the unauthorized download, the breach emphasizes the need for proactive measures.

1.1.3 Motivation for Insider Threat Research

Several factors have prompted us to direct our research towards insider threat detection and prediction. The increasing number of insider threats, both malicious and accidental, has become a critical issue for organisations of all kinds. This surge in threats has significantly exposed sensitive data and intellectual property.

Proactive solutions are essential to address this escalating threat landscape. Insider threat detection and prediction research is at the forefront of this fight. The primary goal is to harness advanced technologies to create effective strategies that identify potential insider threats and implement preventative measures to safeguard critical information. By achieving this, organisations can mitigate internal risks, protect their valuable data assets, and reduce insider threats' financial and reputational risks.

Furthermore, data breaches, often a consequence of insider threats, can have devastating financial implications. organisations face significant costs associated

[1] https://www.hipaajournal.com/former-south-georgia-medical-center-employee-arrested-over-41k-record-data-breach/

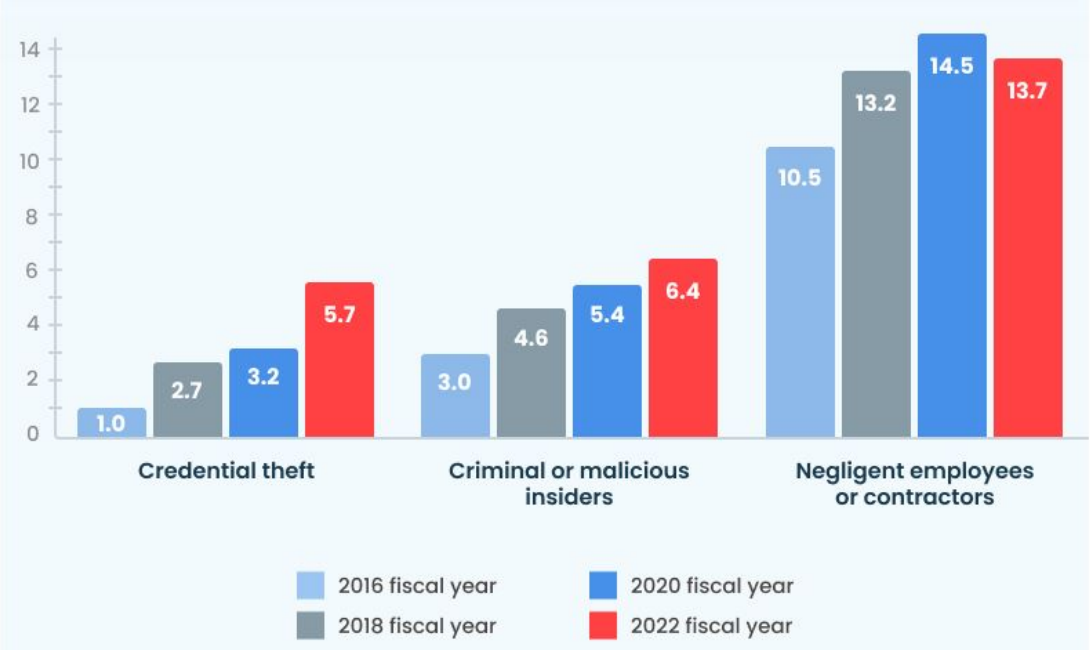


Fig. 1.2. Insider threat increases

with remediation, legal repercussions, and lost business opportunities. Research is driven by the need to develop proactive methods to detect and prevent insider threats. This proactive approach aims to lessen the financial impact of data breaches and ensure an organisation's financial stability.

Figure 1.2 illustrates the rise in insider threats from 2016 to 2022 across various categories. The figure shows an upward trend in all categories, including credential theft, malicious insider threats, and negligent employee behaviour. Notably, the data indicates a significant increase in all these categories in 2022 compared to previous years.

1.2 Research Problems

In past decades, many techniques have been used to control insider threats. Access control techniques are essential for safeguarding data privacy and ensuring security [24, 25, 26, 27]. They are used in diverse domains, including healthcare systems and data dissemination. Despite authorized access, traditional security

measures often fail to thwart insider threats. Conventionally, access control systems are used. In [28] introduced an adaptive risk management and access control framework to mitigate insider threats in organisations. It extends the traditional role-based access control model by incorporating risk assessment and trust in users' behaviour. Even though users can access data, these access controls can't stop them from misusing it. Following these access control systems, much research has focused on understanding insiders and developing methods to detect insider threats [29, 30, 31, 32]. Insider threats can include data exfiltration, espionage and fraud, exposure of classified information, IT sabotage, and theft of intellectual property [33, 34, 35, 36].

The growing prevalence of insider threats has spurred a surge in research aimed at understanding and mitigating these risks. However, despite this research effort, our ability to effectively address insider threats remains limited. We can formulate and explore two primary research questions through subquestions to address this gap.

Research Question 1: Can machine learning and deep learning algorithms leverage standalone and sequential features to achieve superior detection performance for insider threats?

Traditional security methods often struggle to identify insider threats because these individuals have authorized access to data and systems [37, 38, 39, 40]. Therefore, machine learning and deep learning algorithms have emerged as promising tools for insider threat detection. These algorithms can analyse vast amounts of user activity data to identify patterns and anomalies that might indicate malicious intent. However, the effectiveness of these algorithms depends on the features used to train them.

Previous research has explored two main types of features for insider threat detection: standalone features and sequential features. While both offer valuable insights, some studies have focused solely on one type or the other [2, 41, 42, 43, 44]. This leaves a gap in our understanding of the potential benefits gained by combining these features to create a more comprehensive picture of user activity. Subquestion 1 compares the performance of various supervised machine learning algorithms on the CERT r4.2 balanced dataset and in real-world scenarios with

imbalanced datasets. Additionally, subquestion 2 addresses the combination of standalone and sequential features in insider threat detection.

Subquestion 1: How do various supervised machine learning algorithms perform on the CERT r4.2 balanced dataset compared to real-world scenarios with imbalanced datasets, particularly under varying levels of class imbalance, ranging from 40% to as low as 0.5% of insider cases in the dataset?

Machine learning has emerged as a promising tool for various cybersecurity applications, including insider threat detection and cyber-attack prediction [45, 46, 47, 48, 49]. However, a major challenge lies in effectively comparing the performance of existing approaches across different datasets and settings.

Previous research has utilized diverse datasets and settings, making direct comparisons between these approaches difficult due to the influence of these varying parameters [50, 51, 52]. To address this challenge, we propose a controlled evaluation methodology. We compare the performance of various supervised machine learning algorithms on a standardized balanced dataset and under consistent settings. Furthermore, we investigate the impact of hyperparameter tuning on the algorithms' performance. Additionally, we explore the effectiveness of these same algorithms in handling datasets with different levels of class imbalance, reflecting real-world scenarios where insider representation may be much lower than the number of normal user cases.

Subquestion 2: How does combining standalone and sequential features extracted from user activity data impact the performance of insider threat detection algorithms compared to using standalone or sequential features alone?

Existing research on insider threat detection has explored various machine learning and deep learning techniques [6, 53, 54]. These approaches typically focus on two main types of features: behavioural features and standalone features. Behavioural features capture user activity patterns to identify suspicious behaviour, while standalone features focus on static characteristics of user activity. However, a key limitation of many existing approaches is their reliance on only one type of feature, either standalone or sequential. This can lead to a less comprehensive picture of user activity and potentially hinder detection accuracy.

To address this limitation, we propose a bilateral insider threat detection framework. This framework incorporates standalone and sequential features to create a more holistic view of user activity. By combining these features, we aim to improve the effectiveness of insider threat detection compared to approaches that rely solely on one feature type.

Research Question 2: Can incorporating daily ground truth data about insider threat incidents improve the prediction of future insider threats compared to methods that rely solely on historical user activity data?

In recent years, numerous studies have explored machine learning-based approaches for insider threat detection [55, 56, 57]. Similarly, research has focused on user behaviour analysis for threat detection [58, 59, 60]. However, these techniques primarily concentrate on identifying threats that have already occurred based on historical user activity data. Traditional access control methods focus on post-occurrence detection and can lead to response delays, especially for large organisations as highlighted in [61, 62, 63]. This research addresses this limitation by exploring methods for the prediction of insider threats.

Subquestion 3: Can recurrent neural networks (RNN, LSTM, Bi-LSTM) leverage daily ground truth data Xg ***to learn more effective patterns from user activity features*** Xm***,***Xs ***for improved insider threat prediction?***

Traditional methods often struggle with insider threat detection, highlighting the need for a more predictive approach. This research proposes a framework incorporating user activity features, including confirmation of whether an attack occurred each day (ground truth), to train RNN models. The model can learn user behavioural patterns by analysing standalone and sequential user activities. Including ground truth data as a feature allows the RNN to identify deviations from normal user behaviour and potentially refine its predictions. This systematic evaluation will compare the performance of RNNs utilizing this combined feature set with potentially less informative models, aiming to demonstrate the effectiveness of RNNs in learning from ground truth data for improved insider threat prediction.

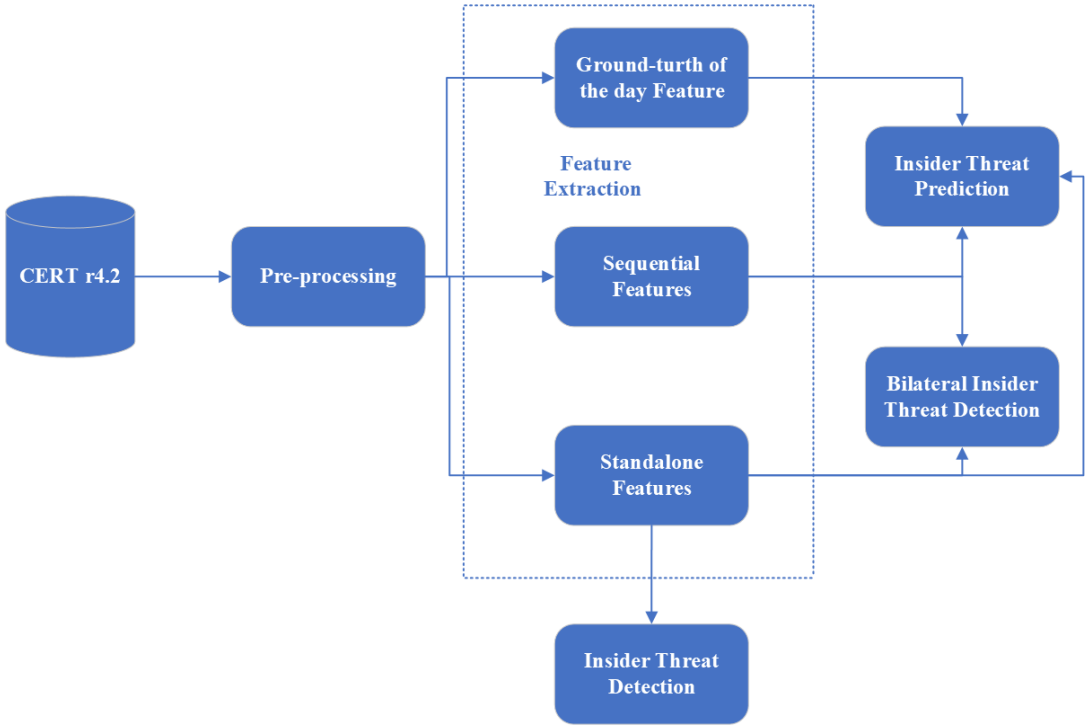


Fig. 1.3. Overall framework

1.3 Thesis Contribution

Outlined below are the primary contributions made by this thesis:

- This research compares the various supervised machine learning algorithms, including RF, XG Boost, KNN, GNB, DT, MLP, AdB, and QDA, using the CERT r4.2 balanced dataset to evaluate their performance in insider threat detection. It also investigates the influence of hyperparameter modifications on the performance of specific machine learning models, namely KNN, DT, and AdB, within the balanced dataset.

- This research examines the performance of various supervised machine learning methods in addressing imbalanced datasets, which are common in real-world scenarios. Specifically, we assessed their effectiveness in the presence of different levels of class imbalance, ranging from 40% to as low as 0.5% of insider cases.

- This research also introduces a novel Bilateral insider threat detection framework that utilizes both standalone and sequential activities from users' daily behaviours. Furthermore, it develops a feature extraction method based on RNNs and LSTM to capture and utilize sequential features in the data.

- The experiments compare the performance of our bilateral features with various classifiers on the CERT r4.2 dataset. Additionally, we assessed the effectiveness of RNN and LSTM feature extractors in combination with the same classifiers, namely KNN, MLP, LR, and SVM.

- This research introduces a comprehensive framework for insider threat prediction that leverages user activity features, including the ground truth of each day. This framework addresses the challenge of accurately identifying potential insider threats by considering both standalone and sequential user activity data from previous days.

- The experiments conduct a systematic evaluation to assess the impact of integrating standalone features Xm, sequential features Xs, and the ground

truth for a specific day Xg on insider threat prediction accuracy. This comprehensive assessment involves a comparative analysis of the performance of three distinct models: RNN, LSTM, and Bi-LSTM on $Xm \| Xs \| Xg$.

- This research investigates the impact of varying predictive lengths on Bi-LSTM's ability to predict threats. Our goal is to identify the optimal length that maximises Bi-LSTM's efficiency in threat prediction. It is achieved by comparing its performance with other models (KNN, LR, AdB, GNB) across different predictive lengths. Additionally, it explores the impact of various embedding sizes (16, 32, 64, and 128) on a BiLSTM architecture with a fixed sequence predictive length (e.g., 5). All models are evaluated using the combined feature set $Xm \| Xs \| Xg$).

1.4 Thesis Structure

This thesis comprises seven chapters, the current one included. The remaining chapters are structured as follows:

Chapter 2 meticulously examines existing knowledge to establish a strong foundation for the research. To understand insider threat detection and prediction, the chapter first defines "insiders" and explores the various insider threats organisations face. It then delves into the motivations behind insider attacks, examining reasons ranging from financial gain and revenge to emotional ones. Following this foundational understanding, the chapter dives deeper into insider threat detection and prediction research. Critically, the chapter also explores various methods for analysing user activity data, a crucial component for both detection and prediction.

Chapter 3 takes a technical turn, delving into the core of the proposed framework: the learning algorithms and datasets. It details the various machine and deep learning algorithms employed in the research, including K-Nearest Neighbors (KNN), Logistic Regression (LR), AdaBoost, Gaussian Naive Bayes (GNB), LSTM, and BiLSTM etc. The chapter explains the functionality of each algorithm, highlighting its strengths and potential applications in insider threat prediction. Furthermore, the chapter explores a specific dataset commonly used

in insider threat detection - CERT r4.2. It delves into the details of this dataset, explaining the types of files and information it contains.

Chapter 4 introduces insider threat detection using supervised machine learning algorithms. This chapter compares a wide range of algorithms, including Random Forest, XGBoost, KNN, GNB, Decision Tree, MLP, AdaBoost, and QDA, on a balanced version of the CERT r4.2 dataset. Furthermore, the chapter presents the impact of hyperparameter tuning on the performance of specific algorithms (KNN, DT, and AdaBoost) within the balanced dataset. Finally investigates how various supervised machine learning methods handle imbalanced datasets, which are common in real-world scenarios.

Chapter 5 introduces a novel approach to insider threat detection: the bilateral framework. This framework leverages standalone user activities (individual actions) and sequential activities (sequences of actions) to enhance detection accuracy. By incorporating this bilateral approach, the research aims to improve traditional methods. Additionally, the chapter proposes a feature extraction method that utilizes Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks to capture the sequential nature of user activity data. To evaluate the effectiveness of this approach, the chapter conducts experiments on the CERT r4.2 dataset. These experiments compare the performance of bilateral features with various classifiers and the effectiveness of RNN and LSTM feature extractors using the same set of classifiers.

Chapter 6 builds upon the previous chapters and proposes a comprehensive framework for insider threat prediction. This framework incorporates user activity data, including a crucial element – the daily ground truth (whether an insider threat occurred that day). This approach addresses the challenge of accurately identifying potential threats by considering individual user actions and sequences from past days. The research employs four supervised learning algorithms to achieve robust and effective threat prediction: KNN, LR, AdaBoost, and GNB. Furthermore, the chapter delves into the effectiveness of Bi-LSTM networks.

Chapter 7 concludes the research findings, highlights the thesis's contributions, and outlines potential future research directions. In Figure 1.4, the overall thesis structure is presented.

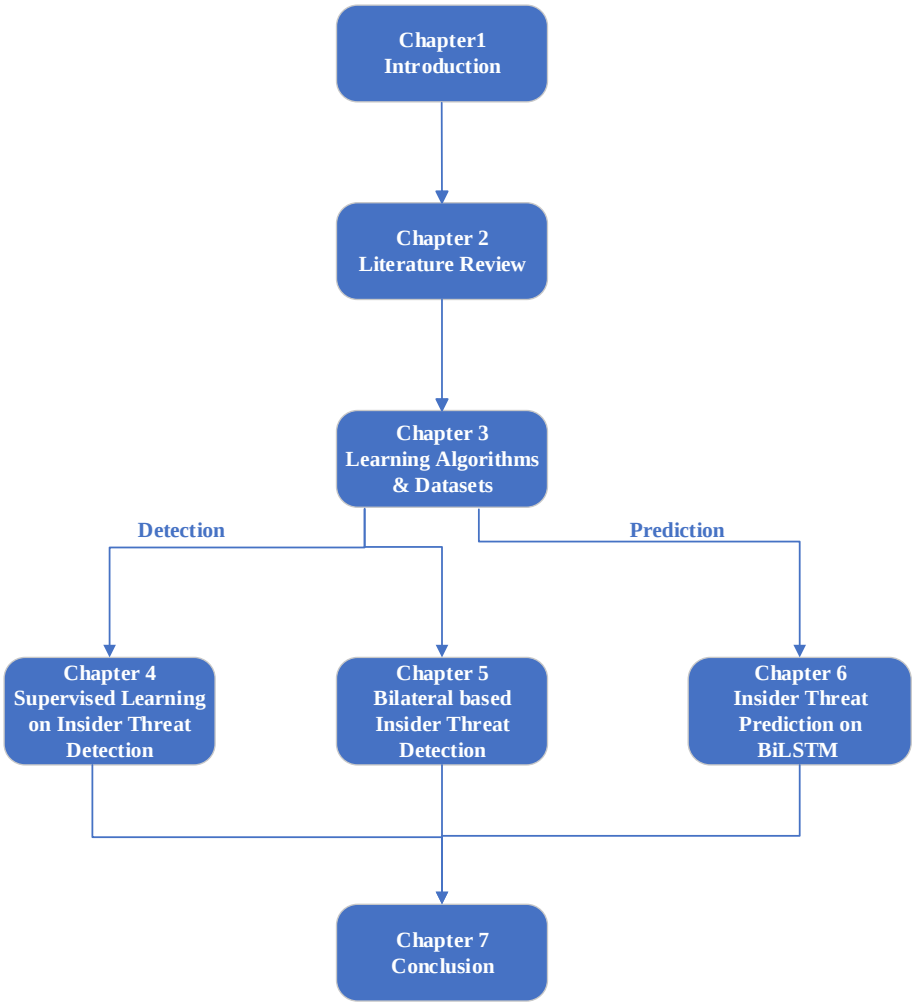


Fig. 1.4. Thesis Structure

Chapter 2

Background and Literature View

Insider threats are one of the most challenging tasks in today's cyber world. Over the past decade, these threats and broader cyber-security issues have become focal points of attention for researchers and organisations alike. Detecting insider threats has emerged as a crucial undertaking within organisational cybersecurity protocols, emphasising the need for robust measures to identify and mitigate risks originating from within the organisation. This chapter offers background information on the nature of insider threats and examines relevant research findings from the literature. By delving into the intricacies of this cybersecurity issue, the chapter aims to contribute to the collective knowledge base and enhance the capabilities of organisations in addressing and safeguarding against insider threats.

Technological advancements and data analytics have played a pivotal role in enhancing insider threat detection and prevention capabilities in this rapidly changing environment. Machine learning and behavioural analytics have become central to identifying and mitigating these risks.

2.1 Insider & Insider Threats

Many of the following definitions draw a clear distinction between insiders and the notion of insider threats.

2.1.1 Who is Insider?

The term "Insider" is defined by the CERT Coordination Center (CERT/CC) [64] as: "A current or former employee, contractor, or business partner who has or had authorised access to an organisation's network, system, or data, and has intentionally exceeded or intentionally used that access in a manner that negatively affected the confidentiality, integrity, or availability of the organisation's information or information systems."

The Rand Corp [65] defined the insider as "anyone with access, privilege, or knowledge of information systems and services.". They also described a malicious insider as "motivated to intentionally adversely impact an organisation's mission" (e.g., deny, damage, degrade, destroy). The term "Insider" is described by Kim et al. [66] as "Someone who has the authority to enter a company as an employee, contractor or guest, regardless of the authority of the information system".

A definition of insider threat described by the US's Cyber and Infrastructure Security Agency (CISA) [67] as: "The potential for an insider to use their authorised access or special understanding of an organisation to harm that organisation. This harm can include malicious, complacent, or unintentional acts that negatively affect the integrity, confidentiality, and availability of the organisation, its data, personnel, facilities, and related resources".

2.1.2 Insider Threats

The term insider threat is defined by Predd et al. [68] as "an insider's action that puts an organisation or its resources at risk". According to Schultz and Shumway [69], an insider attack is "the intentional misuse of computer systems by users authorised to access those systems and networks". As per Pfleeger et al., [70], an insider threat is "an insider's action that jeopardises an organisation's data, processes, or resources in a disruptive or unwelcome manner".

Greitzer and Frinke elaborate that insider threats involve "harmful acts that trusted insiders might carry out, causing harm to an organisation or engaging in an unauthorised act for personal benefit [71]". Hunker and Probst [72]frame insider threat as "an individual with privileges who misuse them or whose access results in misuse."

Bishop [73] conceptualises insider threat as when "a trusted entity abuses given power to violate one or more rules in a given security policy". Theoharidou et al. [74] defines insider threat as "threats originating from people with access rights to an IS (Information Systems) who misuse their privileges, violating the IS security policy of the organisation."

2.1.3 Insider Types

As we've established, insider threats pose a significant risk to organisations across various sectors [75, 76]. However, not all insider threats are created equal. To effectively mitigate these risks, we need to explore in more detail the different motivations and behaviours that can lead to insider breaches.

This section will explore the various types of insiders, categorised by their intent and potential impact. By recognising these distinctions, organisations can develop targeted security measures to address every threat. Figure 2.1 indicates various types of insiders.

In theory, insiders can be categorised into various groups based on their levels of access and authority within the organisation. There are several types of insiders:

1. Malicious insiders
2. Contractors
3. Inadvertent insiders
4. Negligent employees

2.1.3.1 Malicious Insiders

A malicious insider threat occurs when an individual in an organisation possesses the proper authorisation and permissions but engages in harmful activities and thereby poses a security risk. Malicious insiders are typically disgruntled current or former employees who intentionally misuse their access for revenge, financial gain, or both, often after failing to have their credentials revoked [77].

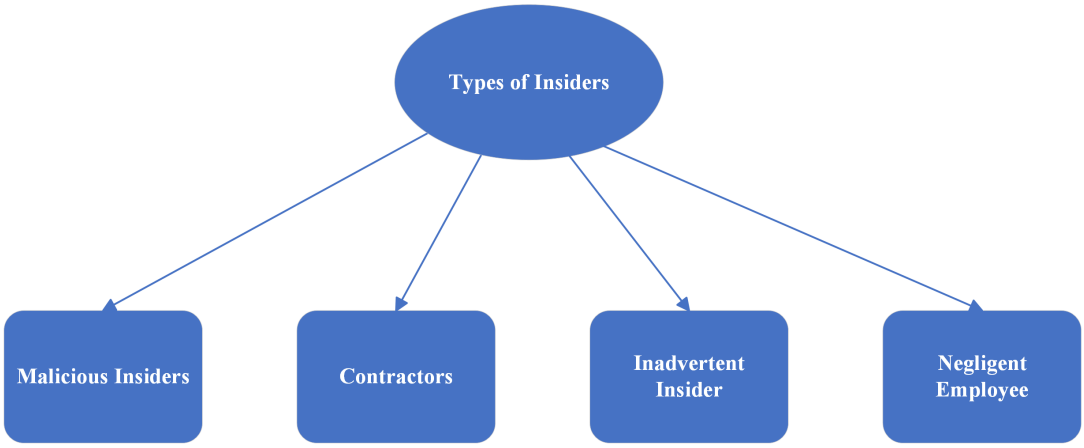


Fig. 2.1. Insider types

2.1.3.2 Contractors

Contractors pose a unique security challenge. Because their temporary presence makes managing access controls challenging, onboarding processes might be rushed, granting contractors more access than necessary. Revoking access after the project is complete can also be lax, potentially leaving contractors with lingering privileges. Furthermore, their lack of familiarity with internal security procedures makes them more vulnerable to social engineering attacks. Essentially, contractors can become unintentional security weaknesses due to the impermanent nature of their work and potential gaps in security awareness.

2.1.3.3 Inadvertent Insiders

Inadvertent insiders are a hidden threat within organisations, the trusted employees who usually follow security protocols. However, their lack of complete security awareness can create vulnerabilities. A single click on a well-crafted phishing link, leaving a work laptop unlocked with sensitive data exposed, or unintentionally leaking confidential information through personal email or lost USB drives - these seemingly harmless mistakes can have serious consequences. Despite having good intentions, inadvertent insiders remain a significant risk, potentially exposing the organisation's data or network to potential breaches without even realising their

mistake.

2.1.3.4 Negligent Employees

A negligent employee unintentionally fails to meet the expected standard of care in their work, causing issues such as errors, accidents, or data breaches. This can stem from carelessness, a lack of knowledge, or poor decision-making, leading to reduced productivity, financial losses, and legal troubles for the employer. Additionally, negligent employees can be susceptible to social engineering attacks, further compromising sensitive information, and may create a negative work environment for colleagues due to unreliable work habits.

2.1.4 Insider Threat Activities

Insider threat attacks vary depending on the organisations, how they are identified, and how they are analysed. Based on these details, insider threats are categorised into the following types.

2.1.4.1 Information Technology (IT) Sabotage

The intentional disruption, manipulation, or destruction of an organisation's IT infrastructure or data characterises IT sabotage. Perpetrators achieve this using various methods, including:

- Denial-of-service (DoS) attacks: Overwhelming a system with traffic to render it unusable for legitimate users.
- Data deletion or manipulation: Deleting critical data files, corrupting databases, or modifying data to cause operational problems.
- Installing malware: Introducing malicious software that can steal data, disrupt operations, or provide unauthorised access to attackers.
- Turning off security controls: Intentionally bypassing or turning off security measures to facilitate other malicious activities.

2.1.4.2 Intellectual Property (IP) Theft

IP theft involves the unauthorised acquisition of sensitive organisational data, such as trade secrets, product designs, customer lists, or proprietary algorithms. Both technical and non-technical personnel can commit this theft.

- Technical personnel might exploit security vulnerabilities to access and steal data, while non-technical personnel might pilfer physical documents or misuse their access privileges to copy electronic data.

2.1.4.3 Fraud

Fraud is the unauthorised manipulation of data for personal benefit. This could involve:

- financial fraud: embezzlement, manipulating financial records, or using stolen credentials to make unauthorised purchases.
- data manipulation: changing data to hide mistakes, create false advantages, or sabotage projects.

2.1.4.4 Espionage

Espionage involves covert or illicit acts of spying on a company, person, or entity to obtain sensitive information. This information could be used for various purposes, such as:

- competitive advantage: spying on competitors to gain insights into their products, strategies, or future plans.
- compromising national security: spying on governments or organisations to obtain classified information.

2.1.4.5 Unintentional Insider Threats

Unintentional insider threats include current or former employees, contractors, or business partners who pose inadvertent risks due to authorised access. These threats can arise from:

- negligence: weak cybersecurity awareness, failure to follow security protocols (e.g., clicking on phishing links), or the accidental sharing of sensitive information.
- human error: downloading malware from untrusted sources, losing laptops or mobile devices containing sensitive data, or making configuration mistakes.
- disgruntled employees: individuals unhappy with the organisation might accidentally leak sensitive information or disrupt operations out of spite.

2.1.5 Levels of Insider Threats

Insider threats can be categorised into three levels based on the severity of their potential consequences and the harm they can inflict on an organisation.

- Low-level threats are unintentional or careless actions by authorised users. These individuals have no malicious intent but can unknowingly compromise security due to mistakes, lack of awareness, or falling victim to social engineering attacks [78].
- Medium-level threats involve insiders with some level of malicious intent but with limited goals. They might be disgruntled employees seeking revenge, opportunistic individuals looking for personal gain, or those pressured by external forces [78].
- High-level threats represent the most serious insider threat, involving individuals with significant malicious intent and the potential to cause substantial damage. These could be highly skilled insiders with privileged access, disgruntled employees with detailed knowledge of the organisation's vulnerabilities, or even foreign spies posing as insiders [78].

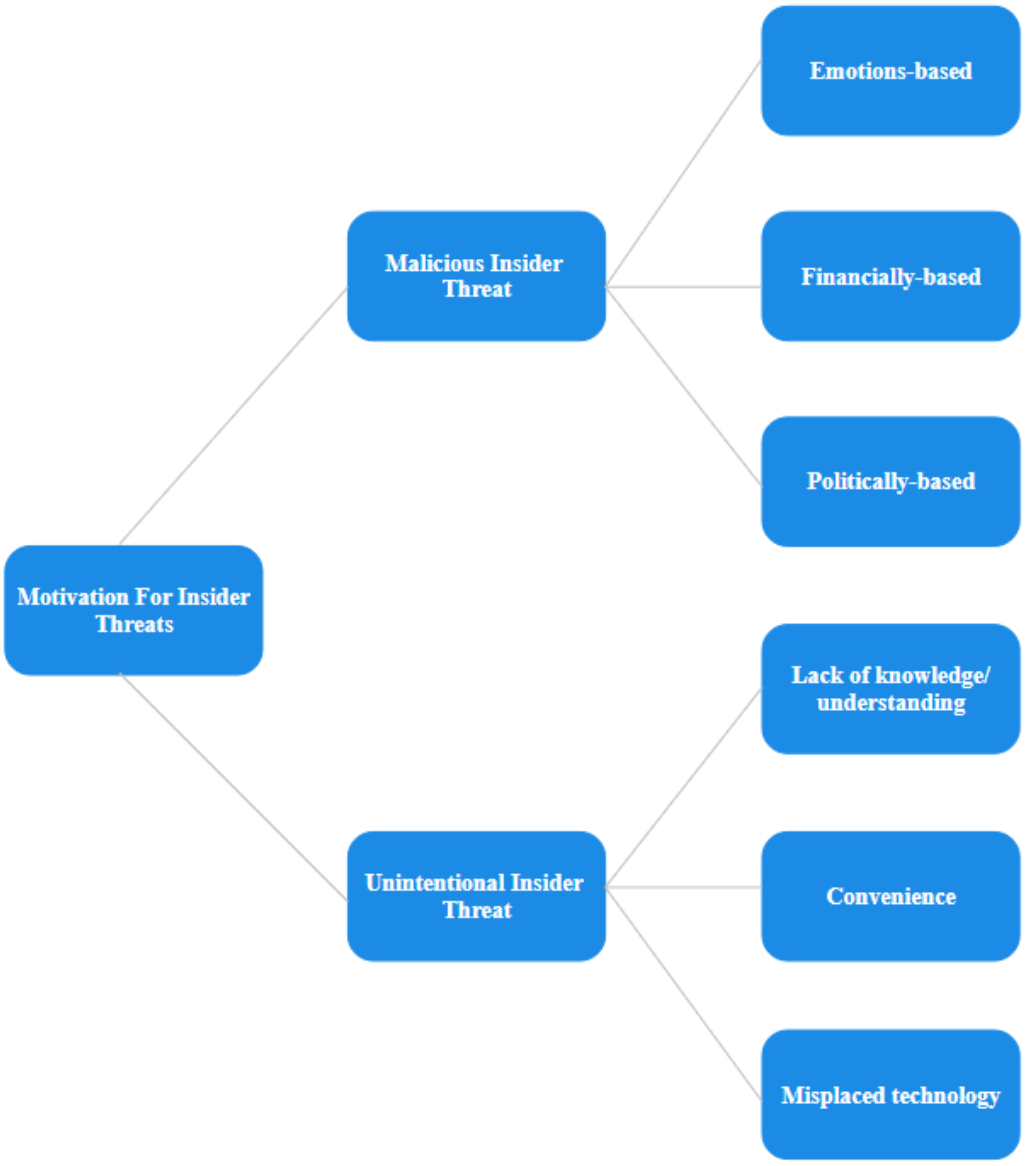


Fig. 2.2. Insider Motivation

Understanding these levels of insider threats can help organisations prioritise their security measures and focus on mitigating risks based on their potential impact.

2.1.6 Motivation for Insider Attacks

Understanding the motivations behind insider attacks is crucial for organisations to develop effective security strategies. Figure 2.2 shows the various motivations for the insider attacks.

1. Financial gain: This is the most common motivator. Insiders may steal or sell sensitive data, embezzle funds, or commit fraudulent activities by exploiting their access.

2. Revenge or retaliation: Disgruntled employees feeling wronged by the organisation, often due to termination, lack of recognition, or unfair treatment, might seek revenge by damaging systems or leaking confidential information.

3. Espionage: Insiders can be recruited by competitors or foreign governments to steal intellectual property, trade secrets, or classified information

4. Negligence: Perhaps the most widespread yet unintended threat, negligent insiders simply lack awareness of security protocols or make careless mistakes that compromise data or systems

5. Politically-based: A desire to expose the organisation's wrongdoings or advance a political agenda can lead to classified information leaks or damage to the organisation's reputation.

6. Emotion-based: Anger, frustration, or a desire for revenge against the employer can drive these insiders to leak information, sabotage systems, or commit fraud.

7. Lack of knowledge/understanding: Security awareness is crucial. Employees who don't understand cyber threats might click on phishing links, share sensitive information inadvertently, or fail to secure their devices properly.

2.2 Literature View

Over the last decade, humans have stored and transferred several bytes of data over the internet. According to a report by Rayaprolu, as of 2025, 463 exabytes of data will be generated per day[79]. These data require protection from both outsider and insider threats. These threats can seriously affect a company's reputation, financial assets, and intellectual resources. Firewalls, intrusion detection systems, access control, authentication, physical security and data encryption techniques control the external threats. However, threats caused by insiders are mostly undetected. A survey report states that 68% of organisations believe they are moderately to highly vulnerable to insider threats [80].

Insiders are often highly trained computer technicians with good internal networks and security control knowledge. They can circumvent conventional security mechanisms and perform a broader range of actions than outside attackers. Insider threat detection is one of the biggest challenges in the cyber world. Various techniques for dealing with insider threats are already in place, such as security information and event management(SIEM), Data Loss Prevention (DLP), User Activity Monitoring (UAM), and Privileged Access Management (PAM).

Insider threat detection has been researched for many years. However, the research community could not significantly contribute to this attack because of the scarce real-time datasets. Eventually, the increasing number of insider attacks attracted a wide range of researchers. Recently, many techniques have been proposed for insider threat detection. DARPA's project ADAMS, which seeks to find trends and anomalies in comprehensive datasets to address insider threats, is the basis of many insider threat detection systems [81].

The complex nature of insider threats necessitates a multifaceted approach to detection within the cybersecurity landscape. This section describes the related work and the literature on different insider threat detection techniques.

Cybersecurity professionals categorise these detection approaches and techniques into several vital strategies. When combined, these strategies enhance an organisation's ability to identify and respond to the diverse and evolving nature of insider threats. On the other hand, malicious insider methods can be clas-

sified into different categories based on the underlying methods and techniques employed.

The malicious threat detection literature commonly employs the following methodologies:

- behaviour-based detection methods
- graph-based detection methods
- anomaly detection methods
- learning-based insider threat detection techniques
- survey and review
- other approaches

2.2.1 Behaviour-based Detection Methods

Behavior-based detection methods analyse individual user activities to identify deviations from established patterns, such as unusual access times, data transfers, or system modifications. These deviations may signal malicious intent and can be used to detect potential insider threats. This subsection discusses some of the existing works on behaviour-based detection methods.

Yuan et al. [58] introduced a sequential method for detecting insider threats based on user behaviour, utilising a Deep Neural Network (DNN) approach. It leveraged the sequential nature of user actions over time by representing them as action sequences. These sequences were then processed in the past using an LSTM network to capture temporal dependencies and extract abstracted temporal features. Subsequently, the extracted features were fed into a Convolutional Neural Network (CNN) classifier to categorise the behaviour as normal or abnormal. The sequential analysis of user actions allowed the model to capture patterns effectively and detect anomalous behaviour within the sequences. In the best case, the proposed approach achieved an AUC of 0.9449 on a CERT r4.2 dataset of insider threats, indicating high accuracy in detecting anomalous behaviour.

In [82], the authors proposed an insider threat detection method based on User Behaviour Analysis (UBA) to address the challenges of insider threats in information security. The model addressed these challenges by aggregating user behaviour data over time, characterising user attributes, and leveraging the XGBoost algorithm for training. It offered several key advantages: overcoming information loss during feature extraction, addressing data imbalance, and minimising false alarms. The experiment results showcased high detection rates, with an outstanding F-measure of 99.96%, exceeding the performance of Support Vector Machines (SVM) and random forest algorithms.

Jiang et al. [83] proposed a novel approach for insider threat prediction, utilising sentiment analysis of network browsing and email content. To achieve this, they implemented a strategy to build user sentiment profiles by monitoring indicators such as web browsing habits, the presence of malicious URLs, and the language used in emails. The system could create daily and weekly sentiment profiles for each user by quantifying users' negative emotions and extreme psychological tendencies. Anomaly detection techniques were then used to identify deviations from a user's established behavioural patterns. This behaviour-based approach focuses on proactively detecting malicious insiders based on their potential attack motivations, such as feelings of revenge or dissatisfaction.

Liu et al. [84] proposed a method for detecting malicious user behaviour using Improved Hidden Markov Models (IHMM) for log data mining. This approach involved recording user actions, analysing data, and analysing them with IHMMs. The system could identify abnormal activities that might indicate potential threats by comparing user activity sequences to established normal behaviour patterns. This focus on behaviour analysis underscored the importance of dynamic monitoring for effective insider threat detection and enhanced information security. The study also suggested exploring the integration of IHMMs with deep learning and artificial intelligence to improve operational efficiency further. Overall, the approach aimed to achieve the comprehensive and accurate detection of malicious behaviour by continuously refining and leveraging user behavioural patterns within network security.

Wang et al. [85] investigated insider threats through a data-centric approach focused on user behaviour analysis, explicitly examining the actions of privileged

users during data interaction (shell commands, keystrokes, mouse events). Their system built models of normal behaviour and identified deviations that might indicate malicious intent. Statistical learning algorithms played a key role in anomaly detection [86]. By profiling user behaviour and intent, the approach aimed to effectively detect insider threats and offer organisations tools to prevent data breaches.

Nasir et al. investigated a novel approach for insider utilised detection that utilised analyse-Autoencoder to analyse user activity data represented as multivariate time-series data [87]. The approach involved collecting data from various sources, such as logon/logoff events, user roles, functional units, and departments. The model then extracted rich features and was trained to detect anomalies that might indicate insider threats. This deep learning technique leveraged the power of LSTM networks, known for their effectiveness in capturing long-term dependencies within sequences, particularly in natural language processing. They applied LSTMs in an autoencoder architecture to automatically learn complex patterns and relationships within the heterogeneous user data.

In the approach proposed by Song et al. [88], the Behavior Rhythm Insider Threat Detection (BRITD) scheme introduced a novel method for prioritising detection by emphasising time awareness and user adaptation. The system captured users utilising behaviour rhythm, using time information to enhance insider threat detection. Employing a feature extraction method that implicitly encoded absolute time information and adapted to behaviour rhythm, BRITD extracted user-day behaviour rhythms tailored to each individual. Additionally, the model outperformed standard insider threat detection models, demonstrating heightened accuracy and precision. The experiment validation and comparisons underscored BRITD's advantages as a comprehensive and innovative solution for insider threat detection in real-world cybersecurity scenarios.

2.2.2 Graph-based Approaches

Over the years, researchers have drawn on graph theory to develop methods for detecting insider threats. These methods analyze user relationships and informa-

tion flow to identify suspicious patterns. This subsection explores several existing studies on methods for graph-based approaches.

Gamachchi et al. [55] proposed a graph-based framework to address malicious insider threats. This framework represented users, systems, and their interactions as a graph, enabling relationships and behaviour patterns analysis. Anomaly detection techniques were employed to identify deviations from normal user behaviour within the graph, potentially uncovering hidden connections indicative of malicious intent. This approach offered advantages in distinguishing legitimate user activity from suspicious patterns.

Liu et al. proposed Log2vec, a system for detecting cyber threats within an enterprise network using heterogeneous graph embedding [89]. Log2vec constructs a heterogeneous graph representing relationships between log entries, including user actions, devices involved, and referenced files. Each log entry is then converted into a low-dimensional vector for efficient analysis. By identifying significant deviations in these vector representations from normal user behaviour, Log2vec is able to detect potential malicious activities. This approach offers advantages in its comprehensive analysis of user interactions and the ability to function without prior examples of cyberattacks while still allowing for the integration of expert knowledge through predefined relationship rules.

Mishra et al. proposed LAC LSTM Autoencoder with Community (LAC) for insider threat detection [90]. LAC utilises an autoencoder to analyse daily user action sequences. This model is trained to reconstruct user behaviour, allowing it to identify deviations that might indicate suspicious activity. The approach went beyond individual analysis by incorporating user communities. By training on interleaved activity sequences within communities, LAC considers the context of user roles and expected group behaviour, potentially improving anomaly detection.

The work in [10] addresses authorise threats, where authorised users exhibit malicious behaviour. The framework, built on daily activities and graph analysis, identifies suspicious behaviour. The proposed daily activity graph approach tracks user actions and connects them based on interactions and potential relationships. To understand user patterns, the framework combines manually selected features with those automatically extracted by an LSTM autoencoder, a

neural network adept at uncovering hidden patterns. Anomaly detection relies on ResHybNet, a deep learning model that merges GNNs to analyse user-activity connections and CNNs to extract patterns from user analysing sequences. By analysing features and network structure, ResHybNet identifies user behaviours deviating from established patterns, potentially indicating insider threats.

The work in [91] introduced the user action graph (UAG), a novel method for identifying insider threats. UAG transforms user actions extracted from system logs into a graph structure. This graph representation captures both the order and relationships between actions, effectively encapsulating the complexity of user behaviour across various system logs. UAG extracts two features: characteristics that characterise overall user activity and local features that capture specific patterns within the graph. Finally, a lightweight model compares a user's behaviour with their historical actions and those of their peers to detect anomalies indicative of malicious intent. The effectiveness of UAG in insider threat detection is confirmed through extensive experiments.

Xiao et al. [92] investigated a novel approach for insider threat detection utilizing GNNs. This method analyzes user interactions within a network by leveraging GNNs. Unlike traditional methods which focus on isolated user actions, GNNs captures user relationships, providing a more comprehensive picture of user behaviour. The model is designed to resist manipulation attempts by malicious insiders. The approach involves modelling user interactions as a graph, with users as nodes and interactions as edges. A GNN then extracts informative features from the graph, considering individual user activities and their connections within the network. Finally, these features are used to identify deviations from normal user behavior patterns, potentially indicating insider threats.

2.2.3 Anomaly Detection Methods

Anomaly detection methods play a vital role in uncovering potential insider threats by identifying unusual patterns in user activities. Subtle anomalies may indicate attempts to bypass security controls or engage in malicious activity. This subsection analyzes several existing studies on methods for detecting these anomalies in the context of insider threat detection.

Gayathri et al. [93] investigated a novel approach to insider threat detection using adversarial training. In this approach, the researchers leveraged adversarial training, a technique commonly used to improve the robustness of machine learning models. The researchers investigated using these adversarial examples to train the model, potentially improving its ability to identify malicious insider activity, even when attackers attempt to disguise their actions. By incorporating adversarial training, the research develops more robust and reliable methods for organizations to identify insider threats.

The approach proposed in [56] focuses on detecting insider threats by modelling a user's normal behaviour. It identifies anomalies using hidden Markov models (HMMs). The approach assumes that a user's anomalous behaviour indicates a potential insider threat. By analysing sequences of actions over time, the model can distinguish between normal and abnormal behaviour patterns. Using HMMs allows for learning parameters from observed sequences and predicting the probability of observing a given sequence. This approach improves insider threat detection by capturing deviations from the user's normal behaviour and detecting behaviour that might indicate a security risk.

Sharma et al. [7] proposed a novel approach for insider threat detection using anomaly detection in user behaviour analytics. An LSTM autoencoder models normal user behaviour by analyzing session-based user activities and extracting feature vectors. The method prioritises deviations (reconstruction errors) from this established pattern to improve security protocols. The model is trained unsupervised on the CERT r4.2 dataset, achieving high accuracy and a low false positive rate.

In [94], the authors investigated the effectiveness of combining supervised and unsupervised learning for insider threat detection. They proposed a workflow that analyzes various data streams, such as emails and logins, to uncover suspicious patterns. The researchers evaluated the impact of different training approaches on detection, finding that using 20% labelled data yielded the best results. This study highlights the importance of optimizing training regimes for superior insider threat detection and suggests supervised and unsupervised learning as a promising approach to improving security measures.

Al-Shehari et al. [95] proposed a novel, unsupervised approach for insider threat detection using an isolation forest algorithm. This method addresses the challenge of imbalanced datasets commonly faced in insider threat detection. The IF algorithm iteratively isolates anomalies by splitting data based on random features. Points that are easier to isolate are flagged as potential threats. Trained on user behaviour data, the system learns normal patterns and identifies deviations that might suggest insider activity. This approach avoids the need for scarce labelled data and focuses on detecting unusual behaviour. The model's effectiveness is demonstrated on a benchmark dataset, highlighting its potential for improved insider threat detection.

Jiang et al. proposed a method for insider threat and fraud detection using graph convolutional networks (GCNs) [96]. They constructed a graph where nodes represent entities (users, systems) and edges represent their interactions. GCNs are then applied to analyse the graph and learn representations for each entity, considering its own attributes and the attributes of its connected nodes. This approach is able to identify nodes with significantly different representations from the norm, potentially indicating anomalous activity. The method's advantage lies in capturing network effects through relationship analysis and learning the effective representations of entities in the network context.

2.2.4 ML & DL Approaches

Machine learning (ML) and deep learning (DL) approaches have emerged as powerful tools for insider threat detection. These techniques leverage large datasets to learn user behaviour patterns and identify anomalies indicative of potential threats. This subsection reviews existing studies that employ ML and DL approaches for insider threat detection.

Le et al. [6] investigated the impact of user data granularity on the ability of machine learning to detect insiders. Their system analysed user activity data at two levels, user-day and user-week. Evaluating its effectiveness against different insider threat scenarios at each level, the system provides insights into how granularity affects detection accuracy and speed. Importantly, it not only detects suspicious activity but also pinpoints the specific insiders involved. This

multi-granularity analysis achieves a balance between catching malicious acts and minimising false positives, ultimately offering guidance for optimising machine learning for better insider detection in organisations.

The paper by Gavai et al. identifies insider threats by analysing employee activity data, including social media interactions, browsing history, and file access patterns [97]. The system flags abnormal behaviours that might indicate insider threats by extracting features from this data and applying anomaly detection. Notably, the approach achieves a well-performing ROC score of 0.77. To aid visualization, a dashboard was developed to help managers and HR personnel identify employees with high-threat risk scores, enabling timely preventive measures to be taken. This method focuses on detecting statistically unusual behaviour, eliminating the need for complex baseline models of normal behaviour.

Le and Zincir-Heywood [98] tackled insider threats with a machine-learning approach. They gathered user activity data across various sources (system logs, network traffic, emails) and extracted features such as access times and locations that reflect user behaviour. By training machine learning models on historical data labeled as normal or suspicious, the system learns typical user patterns in the network. Once trained, the model continuously monitors user activity in real time. Any deviations from established patterns or behaviours flagged as suspicious triggers alerts for investigation. This approach boasts several advantages. Machine learning models are able to continuously learn and adapt to evolving threats, making the system more resilient. Additionally, it scales well to handle large datasets from a growing user base within an organization. Finally, the ability of machine learning to identify subtle behavioural changes surpasses traditional rule-based systems, potentially leading to the earlier detection of insider threats.

Bose et al. [99] tackled insider threats with RADISH, a system designed for real-time anomaly detection. In a departure from traditional methods, RADISH analyzes many data streams (emails, logins) concurrently. This simultaneous analysis allows it to identify suspicious patterns in real time, potentially stopping insider attacks as they unfold. To efficiently handle large data volumes, RADISH employs distributed computing frameworks. The authors argued that RADISH represents a significant advancement in streaming analytics and insider

threat detection. RADISH provides a more comprehensive and timely approach to mitigating insider threats by analysing diverse data streams simultaneously and offering real-time analysis.

In [100], the authors investigated using semi-supervised learning to improve insider threat detection in scenarios with limited labeled data. They proposed a novel system that combined various semi-supervised learning algorithms and considers different data availability situations to enhance detection capabilities. The study explores the impact of different training approaches on the effectiveness of this method. Key aspects examined include data pre-processing techniques, label propagation algorithms for graph-based models, and experiment settings that simulate real-world limitations on data availability. The analysis reveals that using 20% labeled data yields the best detection performance, with the RF algorithm outperforming the others.

In their paper, Bin Sarhan and Altwaijry [101] explored machine learning to identify insider threats and individuals with authorised access who intend to steal or damage data. The authors investigated two approaches: anomaly-based detection, which analyzes deviations from typical user behaviour, and classification-based detection, which trains a model to distinguish between normal and anomalous activity. They discussed algorithms like deep learning and addressed the challenges associated with imbalanced datasets (datasets where one class is significantly larger than others). Building on prior research on the effectiveness of machine learning in user and entity behaviour analytics (UEBA), the paper employs a public insider threat dataset. It achieves promising accuracy results for both detection methods.

In [102], the authors investigated using machine learning to detect insider threats through email analysis. They analysed emails from the TWOS dataset containing regular user activity and simulated malicious insider actions to train supervised machine learning models like AdaBoost and naive Bayes. The authors preprocessed the data by removing noise and converting it into a format suitable for the models. Their findings achieved high accuracy in identifying malicious emails, highlighting the importance of insider threat detection and the challenge insiders pose due to their authorised access.

The approach proposed in [103] addresses the challenge of detecting insider threats by introducing an online unsupervised deep learning system for structured cybersecurity data streams. The approach utilises deep neural networks and LSTM models to learn and adapt to real-time data continuously. It identifies anomalous user behaviour patterns that might indicate potential insider threats. The system also prioritizes interpretability, providing analysts with clear explanations for flagged anomalies. Performance evaluations against standard anomaly detection techniques demonstrate the superiority of the DNN and LSTM models in detecting insider threats. Ultimately, the approach aims to streamline the identification of insider threats by leveraging advanced deep-learning techniques for efficient and effective cybersecurity monitoring.

In [104], the authors proposed a novel approach for insider threat detection using a hierarchical neural temporal point process model. Unlike traditional methods which focus solely on activity types or timestamps, this model considers both aspects by combining temporal point processes and RNNs. The model's key feature is its hierarchical structure: a two-level architecture allows for capturing fine-grained details (intra-session activity) and broader patterns (inter-session behaviour) through separate LSTMs. This comprehensive information modelling considers activity times, types, session durations, and intervals, providing a richer picture of user behaviour. Trained on standard user activity sequences, the model can effectively identify deviations that might indicate malicious insider actions.

Le et al. [105] proposed a user-centric approach based on four supervised machine learning algorithms. The paper explores the effect of different data granularity levels on the accuracy of insider threat detection using machine learning. The authors investigated the use of different feature sets at different levels of granularity, such as user, session, and activity levels, and evaluated their performance using different classifiers.

A recent study proposed a novel method for insider threat detection that combines deep learning with the Dempster-Shafer theory [106]. This approach utilises attention-LSTM classifiers and multi-head attention mechanisms to identify anomalous network behaviour patterns in real time effectively. The experiment results show that the proposed method surpasses traditional perimeter security mechanisms in effectiveness. Future research aims to improve the method's

efficiency and scalability by incorporating user and content metadata, leading to more robust defence strategies. Backed by research grants and building on existing work in data leakage prevention and machine learning cybersecurity applications, the study presents a cutting-edge solution for insider threat detection, paving the way for further advancements in the field.

Anju et al.[107] investigated unsupervised machine learning techniques, specifically anomaly detection with deep learning, to identify insider threats within organizations. The study detects unusual user behaviour that could indicate malicious activity. They introduced novel data representations for these algorithms and explored the effectiveness of combining different strategies to improve performance. The paper also discusses key points such as the importance of accuracy and precision metrics for evaluating models in insider threat detection, the use of specific techniques like one-class classification (OCC) and SVMs for anomaly detection, and the challenges of interpretability faced by complex deep learning models despite their power. Finally, the paper highlights the importance of feature extraction, model description, and deep learning-based insider threat detection training. Overall, the research underscores the potential of deep learning and anomaly detection for improved insider threat detection and the need for continuous development in this critical cybersecurity area.

Lu et al. [54] proposed a framework called Insider Catcher based on the deep learning technique, the LSTM model, to represent the system logs structured sequence. In [108], a deep learning model consisting of CNN and LSTM models was proposed based on the users' behaviours and character embeddings.

In [109], the proposed insider threat detection framework integrates statistical and sequential analysis using three steps and four core modules. These modules encompass log file merging, parallel processing for the statistical and sequential analysis of user behaviours, and a decision-making module for identifying a threat framework, which utilises CNNs and a transformer model, proving to be effective and robust in detecting insider threats and malicious scenarios using the CERT dataset.

A significant approach to predicting insider threats utilizes LSTM models on system logs. Ma et al. [110]investigated the methodology, treating log data as sequential sequences of user activities. They modelled system logs as natural

language sequences to capture long-term dependencies and patterns. Organized sequences are then organized into action workflows, with deviations from established patterns, indicating potential threats. The LSTM models are trained to detect anomalous behaviour and classify threats using the CERT insider threat dataset. The method, which involves the sequential analysis of system logs, is highly successful, achieving a remarkable 93% accuracy in predicting insider threats. This success can be attributed to its ability to examine temporal relationships and dependencies within the log data thoroughly. This related work underscores the importance of leveraging LSTM models for effective insider threat prediction within the cybersecurity domain.

2.2.5 Survey and Review

This section presents a comprehensive review of the existing literature on insider threat detection methods to establish a strong foundation for the research presented later. It summarizes the key findings, limitations, and research trends in insider threats.

The survey conducted in [57] investigated the dangers of authorized users within organizations turning malicious. These insider threats pose a significant risk to sensitive data and systems. To mitigate these threats, the survey proposed a layered approach. The first layer focuses on detection through techniques like user behaviour monitoring, anomaly detection, and network traffic analysis to identify suspicious activity. Deviations from regular user patterns, such as unusual access attempts or data transfers, might indicate malicious intent. Early identification allows organizations to prevent further damage. The second layer emphasizes prevention through access controls. Organizations follow the principle of least privilege, only granting users the access level necessary for them to perform their job. Additionally, data encryption safeguards sensitive information even if unauthorized individuals access it. Finally, security awareness training programs educate employees about cybersecurity best practices and the potential consequences of insider threats. This layered approach offers a comprehensive strategy for organizations to combat insider threats.

Sabir et al. [111] investigated ML to combat data exfiltration, where attackers steal sensitive information. Analyzing 92 research papers, they identified two main ML approaches: data-driven (focusing on the content being transferred) and behaviour-driven (analysing how data is accessed/transferred). Feature engineering trains the ML models by selecting relevant data points, such as user behaviour or transfer patterns. The study explores how researchers assess these models' effectiveness using simulated data, real-world datasets, and various metrics. Finally, the researchers recommended future research directions, such as combining these approaches, developing better evaluation methods, and making ML models more resistant to attacker manipulation. This research highlights the potential of ML as a powerful weapon in the fight against data exfiltration.

Audit data encompasses the documented computer-related activities performed within an organization. Organization administrators closely monitor these records to formulate strategies for mitigating potential insider threats. Traditional approaches to user profiling-based detection rely on three primary categories of audit data sources: host, network, and contextual [112].

Alsowail and Al-Shehari's study [113] tackled insider threats, where authorized users act maliciously. They proposed a layered prevention approach. The base is formed by robust access controls with minimum privileges (enforced by RBAC). DLP tools monitor and restrict data transfer, hindering exfiltration. UAM systems track user actions to detect suspicious behaviour. Beyond technical measures, security awareness training educates employees to identify and report suspicious activity. Finally, a well-defined incident response plan ensures a swift and effective response to contain damage and investigate the incident. This combination of preventative measures helps organizations significantly reduce the risk of insider threats.

Homoliak et al. [114] conducted a critical survey of insider threats in IT systems, a pressing security concern where authorized users pose a significant risk. Their work is a comprehensive resource, examining the issue from various angles. The survey explores different taxonomies for classifying insider threats that consider motivations and methods. Additionally, the paper examines the analysis techniques used to identify and assess these threats, exploring techniques like user behavior analytics and anomaly detection in detail. Furthermore, the

survey delves into methods for modelling insider behaviour. These models predict potential threats by analyzing user activity patterns. Finally, the paper explores various countermeasures organizations can implement to mitigate insider threats. These countermeasures include stricter access controls, employee monitoring, and security awareness training.

2.2.6 Other Approaches

Yuan et al. [115] proposed an attention-based LSTM model for more effective insider threat detection. Traditional methods miss subtle anomalies in user behaviour. This approach addresses this by leveraging LSTMs which are adept at capturing long-term patterns in sequential user actions. The model further incorporates an attention mechanism to prioritize the most relevant parts of these actions, focusing on those indicative of potential threats. This ability to capture long-term patterns and focus on crucial actions could better improve accuracy compared to traditional methods.

Rashid et al. [56] employed HMMs to model users' weekly activity sequences and identify potential insider assaults from small variations in weekly user activities, as indicated by HMM probabilities (of user sequences) below a predetermined threshold.

Yilmaz and Can [116] explored the potential of artificial intelligence (AI) for improved insider threat detection. Traditional methods often struggle with complex or evolving threats. The paper highlights AI's ability to analyze vast amounts of user data (network activity, logs, emails) to identify subtle anomalies and patterns in user behaviour that might indicate malicious intent. AI models can even be trained for predictive analytics. This approach offers potential benefits such as improved accuracy, proactive threat identification, and reduced reliance on manual analysis by security professionals. The paper delves into different AI techniques suitable for insider threat detection tasks, along with the challenges and limitations.

Brdiczka et al. [117] investigated a proactive approach for insider threat detection that combines social network analysis and psychology. To identify potential

threats before they cause harm, the approach utilizes structural anomaly detection (SA) to find unusual patterns in user interactions and information networks. Psychological profiling (PP) is also used to build dynamic user profiles based on the five-factor personality model to identify deviations from established behavioural patterns. This combination of social network analysis and individual behavioural assessment offers a more comprehensive approach to insider threat detection with the potential for proactive risk identification.

The NIST Model for Role-Based Access Control (RBAC) addresses the historical lack of a unified standard in access control [118]. It incorporates concepts from existing models, commercial products, and research prototypes. This approach establishes a foundational standard and categorizes RBAC into four levels with progressively more advanced features (offering greater granularity of control). The model acknowledges traditional group-based access control by emphasizing flexible user assignment. While emphasizing a foundation built on areas of consensus within the RBAC community, it also recognizes aspects that require further standardization. Various applications have since been implemented using RBAC methods [45, 46, 47].

The work in [119] addresses the challenge of balancing privacy requirements with the utility of threat detection systems. The authors proposed a risk-based approach to access control, where each access request is evaluated based on its potential privacy risk and the user's trustworthiness. When the privacy risk exceeds a threshold, adaptive adjustments such as data obfuscation or enforceable obligations are applied. This framework effectively balances privacy needs with utility requirements and is implemented in an industrial threat detection solution. Overall, it provides a dynamic and context-aware approach to access control, making it well-suited for threat detection systems.

2.3 Summary

Chapter 2 sets the stage for the primary research. It defines insider threats, explores their types and detection challenges, and examines the motivations and activities of malicious insiders 2.1. The chapter then reviews existing methods

for insider threat detection 2.2, highlighting their key findings and limitations. Existing methods often struggle with two main issues: imbalanced datasets, where malicious activities are rare compared to normal behaviour, and limitations in capturing the sequential nature of user actions. This can lead to missing subtle anomalies indicative of threats. By establishing this context and highlighting the limitations of current methods, the background and related work chapter paves the way for the main research and its potential contribution.

Chapter 3

Classic Learning Algorithms and Datasets

This chapter describes the classical learning algorithms and datasets employed in the thesis. The chapter encompasses machine learning and deep learning algorithms, which play pivotal roles in subsequent chapters which focus on insider threat detection and prediction. These algorithms form the foundation for comprehensive analyses, enabling the identification and anticipation of insider threats. Furthermore, we delve into the nuances of their application and effectiveness within cybersecurity, providing a thorough understanding for the reader. Firstly, we summarize the machine learning algorithms followed by deep learning algorithms and the datasets.

3.1 Learning Algorithms

In relation to machine learning, this section unveils a comprehensive exploration of algorithms essential to our insider threat detection and prediction thesis. These machine learning algorithms serve as the cornerstone for rigorous analyses, offering a nuanced understanding of their application to cybersecurity. In subsequent sections, we delve into the intricacies of these algorithms, unravelling their functionalities and impact on enhancing cybersecurity measures.

3.1.1 RF

Random forest (RF), categorized as an ensemble learning method, combines multiple models, primarily decision trees, to enhance overall performance [120]. Decision trees, fundamental in supervised learning, serve dual purposes for classification and regression tasks. Operating through recursive data splitting based on input feature values, decision trees refine subsets until each contains a singular class (for classification) or a solitary value (for regression) [121].

Random forest is a popular machine-learning algorithm that uses an ensemble of decision trees to make predictions. The algorithm constructs multiple decision trees by randomly selecting a subset of features and a subset of observations to train each tree. Such a methodology aids in mitigating overfitting and enhancing generalization performance. The final prediction is made by aggregating the predictions of all the individual trees. For regression tasks, the final prediction is typically the average of the predictions from all the individual trees in the forest. For classification tasks, the final prediction is based on the majority class predicted by the individual trees [122, 123].

The random forest algorithm is a powerful machine learning method that can handle high-dimensional data and nonlinear relationships between variables. It is known for its accuracy, robustness, and ability to handle missing data. The algorithm also provides variable importance measures, which can be used to identify the most important features for making predictions. The random forest algorithm is a versatile and effective machine-learning method that can be applied to various prediction problems.

The equation for a random forest is an aggregation of the predictions from individual trees. For regression tasks, the final prediction is the average of the predictions from all the individual trees in the forest. Mathematically, this can be represented as:

$$\hat{Y} = \frac{1}{N} \sum_{i=1}^{N} f_i(X) \tag{3.1}$$

where $\hat{Y}$ is the predicted value, N is the number of trees in the forest, $f_i(X)$ is the prediction of the ith tree for input X.

For classification tasks, the final prediction is based on the majority class predicted by the individual trees.

3.1.2 XGB

Extreme Gradient Boosting (XGBoost) has emerged as a leading machine learning framework due to its scalability, efficiency, and effectiveness in handling large-scale datasets. It consistently achieves state-of-the-art results in diverse machine learning challenges [124]. Its remarkable scalability is a standout feature, running more than ten times faster than existing solutions on a single machine and effortlessly scaling to handle vast datasets in distributed or memory-limited settings. Innovative systems and algorithmic enhancements underpin the scalability. These include a specialized tree learning algorithm for sparse data, a theoretically justified weighted quantile sketch procedure accommodating instance weights in approximate tree learning, and efficient parallel and distributed computing, streamlining the learning process for quicker model exploration.

In supervised learning, XGBoost excels in regression and classification tasks, offering a versatile solution for predicting target variables based on input features. Trained on labeled datasets where the target variable is known for each example, the algorithm optimizes a loss function to learn the intricate mapping from input features to the target variable. Once trained, the model proves adept at predicting target variables for new, unseen examples. Its widespread adoption in supervised learning scenarios is driven by its exceptional scalability, speed, and accuracy. XGBoost's success extends across various applications, ranging from store sales prediction to ad click-through rate prediction, as validated by its consistent dominance in machine learning competitions, where it often outperforms competitors and holds its ground against ensemble methods[125].

3.1.3 DT

Decision Trees (DT) are fundamental in machine learning and data mining, providing a hierarchical representation of decision-making processes. They are used for classification and predictive modeling, where the data is recursively split into

subsets based on attribute values to make decisions about class labels. This hierarchical structure consists of nodes representing decision points and connections representing possible outcomes, allowing for an intuitive and interpretable model. Decision trees are versatile, capable of handling both categorical and numerical data, and are known for their ease of use and ability to handle noisy or missing data [126, 127].

However, decision trees are susceptible to overfitting, where the model fits the training data too closely and performs poorly on new data. Techniques such as pruning and ensemble methods have been developed to improve their generalization performance. Despite this limitation, decision trees remain a powerful and widely used tool in various fields, providing valuable insights and practical solutions for classification and predictive modeling tasks. The goal is to create a model that predicts the class of a new instance by traversing the tree from the root to a leaf node [128].

3.1.4 GNB

The Gaussian naive bayes (GNB) algorithm is a supervised learning method designed explicitly for classification tasks. It is a popular classification algorithm grounded in Bayes' theorem, operating under the assumption that features are conditionally independent given the class. The algorithm calculates the probability of a data point belonging to a particular class based on the distribution of features in that class. It assigns the class label with the highest probability. Additionally, it calculates the probabilities of each attribute belonging to each class and uses these probabilities to make predictions. It assumes that the probability of each attribute belonging to a given class value is independent of all other attributes. It is particularly well-suited for situations where the attributes follow a Gaussian (normal) distribution [129, 130, 131].

One of the key advantages of the GNB algorithm is its simplicity and efficiency, especially for high-dimensional data. It requires a small amount of training data to estimate the parameters necessary for classification, making it particularly useful when available training data is limited. The GNB algorithm's ability to

handle continuous data and its efficiency in training and classification makes it a valuable tool in various real-world applications.

$$P(M|N) = \frac{P(N|M) \cdot P(M)}{P(N)} \tag{3.2}$$

Equation 3.2 represents the conditional probability of event M given event N. In the context of GNB, this equation is used to calculate the probability of a class value (M) given the observed attribute values (N). Here, $P(N|M)$ represents the conditional probability of the observed attribute values given the class value, $P(M)$ is the prior probability of the class value, and $P(N)$ is the probability of the observed attribute values. By calculating this conditional probability for each class, GNB selects the class with the highest probability as the predicted class for the given attribute values[132].

3.1.5 KNN

K-Nearest Neighbors (KNN) is a versatile machine-learning algorithm for classification and regression tasks. It operates on the principle of proximity, where the prediction for a new data point is based on the majority label or value of its K-nearest neighbors in the feature space. KNN is non-parametric, meaning it does not assume any specific form for the underlying data distribution, making it suitable for various applications. It is particularly effective for small to medium-sized datasets and can handle numerical and categorical data.

One of the strengths of KNN is its simplicity and ease of implementation. It does not require training or model fitting, making it a straightforward choice for the quick prototyping and exploration of datasets. However, KNN's performance can be impacted by the curse of dimensionality, especially in high-dimensional feature spaces, and it may be computationally expensive for large datasets. Various KNN variants, such as distance-weighted KNN and semi-supervised KNN, have been developed to address these challenges, aiming to enhance the algorithm's efficiency and accuracy in different scenarios [133, 134].

3.1.6 QDA

Quadratic discriminant analysis (QDA) is a classification algorithm used in machine learning and statistics to classify data into multiple classes based on their features. QDA is a supervised learning method requiring labeled training data to learn the relationships between features and classes. QDA aims to find a decision boundary that best separates the classes in the feature space [135].

In QDA, it is assumed that the data for each class follows a multivariate normal (Gaussian) distribution. It also implies that the features of each class are assumed to be normally distributed and that the covariance matrix can differ between classes. The decision boundary is determined by fitting a quadratic surface to the data, allowing for more complex decision boundaries compared to linear classifiers such as linear discriminant analysis (LDA). The essential components of QDA include the mean vector and the covariance matrix for each class. The mean vector represents the average value of each feature for a given class, while the covariance matrix describes the spread and relationships between the features within each class. These parameters are estimated using maximum likelihood estimation or Bayesian estimation from the training data.

To classify a new data point, QDA calculates the probability of the data point belonging to each class based on the class-specific multivariate normal distributions. The decision rule assigns the data point to the class with the highest probability.

The equation gives the probability density function (PDF) of the multivariate normal distribution:

$$f(x|\mu, \Sigma) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)} \tag{3.3}$$

where - x represents the feature vector, - μ is the mean vector, - Σ is the covariance matrix, - p is the number of dimensions.

The decision boundary in QDA is determined by fitting a quadratic surface to the data. It can be expressed as the following quadratic equation:

$$x^T Ax + B^T x + C = 0 \tag{3.4}$$

where - A is a symmetric matrix of coefficients, - B is a vector of coefficients, - C is a constant.

To classify a new data point, QDA calculates the probability of the data point belonging to each class based on the class-specific multivariate normal distributions. The decision rule assigns the data point to the class with the highest probability, which can be expressed using Bayes' theorem and the PDF of the multivariate normal distribution.

The classification rule for QDA can be expressed as:

$$\hat{y} = \arg \max_{k \in \{1,2,\ldots,K\}} \left\{ \log \pi_k - \frac{1}{2} \log |\Sigma_k| - \frac{1}{2}(x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k) \right\} \tag{3.5}$$

where

- $\hat{y}$ is the predicted class label,
- π_k is the prior probability of class k,
- μ_k is the mean vector for class k,
- Σ_k is the covariance matrix for class k,
- x is the feature vector of the new data point.

QDA has several advantages over linear classifiers such as LDA, including its ability to handle non-linear decision boundaries and its flexibility in modeling the relationships between features and classes. However, QDA requires more parameters to estimate compared to LDA, which can lead to overfitting when the number of features is large relative to the number of training samples[136].

3.1.7 AdB

AdaBoost (AdB), introduced in 1995 by Freund and Schapire, is a popular ensemble learning algorithm that combines the predictions of multiple weak classifiers to create a robust classifier. The algorithm is designed to iteratively train a sequence of weak classifiers on weighted versions of the training data, where the

weights are adjusted at each iteration to focus on the examples misclassified in the previous iterations. By giving more emphasis to the problematic examples, AdaBoost aims to improve the overall performance of the ensemble [137, 138].

The fundamental idea behind AdaBoost is to minimize the exponential loss function, which is defined as follows:

$$L(y, F(x)) = \exp(-y \cdot F(x)) \tag{3.6}$$

where y is the true label of the example x and $F(x)$ is the weighted sum of the weak classifiers' predictions on x.

The AdaBoost algorithm determines the weights of the weak classifiers during training.

The algorithm starts by assigning equal weights to all training examples and then trains a weak classifier on this weighted data, such as a decision stump. After each iteration, it increases the weights of the misclassified examples while decreasing the weights of the correctly classified examples. Subsequent weak classifiers then train on the updated weighted data, with each classifier focusing on the examples that were previously misclassified. The final robust classifier is constructed by combining the individual weak classifiers based on their performance, with more accurate classifiers being given higher influence in the ensemble. The equation for the final classifier is:

$$H(x) = \text{sign}(F(x)) \tag{3.7}$$

where sign is the sign function and $F(x)$ is the weighted sum of the weak classifiers' predictions on x.

One of the strengths of AdaBoost is its ability to adapt to complex decision boundaries and handle noisy data. AdaBoost can effectively learn from difficult instances and improve its generalization performance by focusing on the challenging examples to classify. AdaBoost is less prone to overfitting than other machine learning algorithms, making it suitable for various applications. However, AdaBoost is sensitive to outliers and noisy data, as it may excessively focus on misclassified examples, leading to decreased performance. Furthermore, the algorithm's performance can be affected by the choice of weak classifiers and the quality of the training data.

3.1.8 MLP

The multilayer perceptron (MLP) is a fundamental type of artificial neural network (ANN) that has gained widespread popularity due to its ability to learn and model complex relationships in data [139]. It is a feedforward neural network (FNN), meaning that the flow of information moves in one direction, from the input layer through one or more hidden layers to the output layer. Interconnected nodes, or neurons, comprise each layer, and the training process adjusts the weights associated with the connections between neurons. The MLP's versatility allows it to apply to various tasks, including pattern recognition, classification, regression, and function approximation [129, 140].

The structure of an MLP typically includes an input layer, one or more hidden layers, and an output layer. The input layer receives the initial data, which is then processed through the hidden layers, and the output layer produces the final result. Neurons in the MLP use activation functions to introduce non-linearity into the network, allowing it to learn and model complex relationships in the data. Common activation functions include the sigmoid function for hidden layers and the softmax function for the output layer in classification problems.

In training an MLP, one adjusts the weights and biases of the network to minimize the difference between the predicted and actual output. Typically, achieving this involves employing the backpropagation algorithm, a supervised learning method. The network iteratively adjusts its weights and biases during training based on the error between the predicted and actual output. The backpropagation algorithm calculates the error gradient to the network's weights and biases. It uses this information to update the network parameters in a direction that minimizes the error [141, 142].

In an MLP, the learning process entails iteratively optimizing the network's parameters to minimize a predefined loss function, such as the mean squared error for regression tasks or the cross-entropy loss for classification tasks. Typically, gradient-based optimization algorithms like stochastic gradient descent (SGD) or its variants are employed for this optimization, adjusting the weights and biases in the direction that reduces the error.

The equations used in training MLP are fundamental to its learning process. During training, the backpropagation algorithm is employed to adjust the weights and biases of the network based on the error between the predicted and actual outputs.

The weight update equation for a connection between neuron k in layer l-1 and neuron j in layer l is given by:

$$\Delta w_{jk}^{l} = -\eta \cdot \delta_{j}^{l} \cdot y_{k}^{(l-1)} + \alpha \cdot \Delta w_{jk}^{l} \tag{3.8}$$

- Δw_{jk}^{l} represents the change in weight for the connection between neuron k in layer $l-1$ and neuron j in layer l,
- η is the learning rate, which controls the step size of the weight updates,
- δ_{j}^{l} is the error term for neuron j in layer l representing the contribution of neuron j to the overall error,
- $y_{k}^{(l-1)}$ is the output of neuron k in layer $l-1$,
- α is the momentum term, which influences the impact of the previous weight update on the current update.

These equations are fundamental in the iterative process of adjusting the network's weights and biases during training, enabling the MLP to learn and model complex patterns and relationships in the data.

3.1.9 LR

Logistic regression (LR) is a statistical method that models the relationship between a binary outcome variable and one or more independent variables. Widely employed in various fields, including medical research, economics, and social sciences, LR is particularly valuable when the outcome of interest is dichotomous, such as the presence/absence of a disease, success/failure of a treatment, or yes/no responses [143, 144, 145].

The fundamental concept of LR involves estimating the probability of the binary outcome based on a linear combination of the independent variables, transformed using the logistic function. The logistic function ensures that the predicted probabilities fall within the range of 0 to 1, making it suitable for modeling binary outcomes. LR allows for assessing the impact of each independent variable on the likelihood of the outcome while controlling for the effects of other variables.

One of the key advantages of LR is its ability to handle continuous and categorical independent variables. LR provides insights into the direction and strength of the relationships between the independent variables and the probability of the outcome. Additionally, LR can be extended to include interactions and higher-order terms, allowing complex relationships to be explored. However, LR does have limitations. It assumes a linear relationship between the independent variables and the log-odds of the outcome, which may not always hold. Furthermore, LR assumes that the observations are independent, which may not be the case in clustered or correlated data.

We now delve into the logistic regression equation. The logistic regression equation is expressed as:

$$P(Y = 1|X) = \frac{1}{1 + e^{-(b_0+b_1X_1+b_2X_2+...+b_iX_i)}} \tag{3.9}$$

Explanation:

- $P(Y = 1|X)$ represents the probability of the outcome variable (Y) taking the value 1 given the values of the independent variables (X).
- e is the base of the natural logarithm.
- b_0 is the intercept, indicating the log-odds when all independent variables are zero.
- b_1, b_2,..., b_i are the coefficients associated with the independent variables X_1, X_2,..., X_i, representing the change in the log-odds of the outcome variable for a one-unit increase in the corresponding independent variable, holding all other variables constant.

This equation forms the core of logistic regression modelling. It enables the estimation of the binary outcome's probability based on the independent variables' values and their associated coefficients.

3.1.10 RNN

A recurrent neural network (RNN) is an artificial neural network for processing sequential data. Unlike traditional FNNs that handle input data in a single pass, RNNs excel at managing sequential information by maintaining an internal state that captures dependencies and patterns across time. In contrast to FNNs, which process each input independently through hidden layers without considering the order or context of other inputs, RNNs are more effective in handling sequential data. They are well-suited for sequential data tasks, such as time series analysis, natural language processing, and speech recognition [146].

At the core of RNNs is the concept of recurrence, where the network's internal state undergoes updates at each time step based on current and past inputs. This unique feature empowers the network to retain a memory of preceding inputs, influencing its present output and enabling the capture of temporal dependencies in the data. The ability to capture temporal dependencies is a key strength of RNNs, allowing them to effectively model and analyse sequential patterns. Centered around the processing of sequential data, RNNs distinguish themselves from traditional FNNS. With their internal memory, RNNs maintain a state that encapsulates information about previous inputs, facilitating dynamic temporal behaviour. The recurrent nature of RNNs makes them particularly well-suited for a range of tasks involving sequences, such as predicting the next word in a sentence, generating music, or analyzing stock market trends [147].

The RNN architecture is generally built upon recurrent connections that form a loop, allowing information to persist over time. At each time step, the network receives an input, produces an output, and updates its internal state based on the current and previous input. This recurrent nature enables RNNs to capture dependencies and patterns in sequential data, making them powerful tools for modeling and understanding time-varying phenomena.

One of the crucial components of an RNN is the hidden state, functioning as the network's internal memory. At each time step, the hidden state undergoes updates based on the current input and the previous hidden state. This process dynamically incorporates information from past inputs, effectively encoding the network's memory. The recurrent updating mechanism empowers RNNs to capture long-range dependencies in sequential data, a capability distinguishing them from traditional FNNs lacking this memory aspect. Fig 3.1 shows the architecture of the RNN.

An RNN's architecture typically consists of three main components: an input layer, a hidden layer, and an output layer. However, unlike FNNs, the hidden layer in RNNs has a feedback connection to itself, allowing the network to maintain a memory of previous inputs.

The architecture of an RNN can be represented as follows:

- Input layer: The input layer receives the input sequence, a sequence of words, images, or any other type of sequential data. Each element in the sequence is fed into the network one at a time.

- Hidden layer: The recurrent connections are located in the hidden layer. The hidden state at each time step is computed based on the current input and the previous hidden state. This allows the network to maintain a memory of previous inputs and capture temporal dependencies in the sequence.

- Output layer: The output layer produces the output sequence, a sequence of predicted words, images, or any other sequential data type. The output at each time step is computed based on the current hidden state.

RNNs can be represented using the following equations:

The hidden state at time step t:

$$h_t = f(W_{hh}h_{t-1} + W_{xh}x_t + b_h) \tag{3.10}$$

The output at time step t:

$$y_t = g(W_{hy}h_t + b_y) \tag{3.11}$$

where

-h_t is the hidden state at time step t

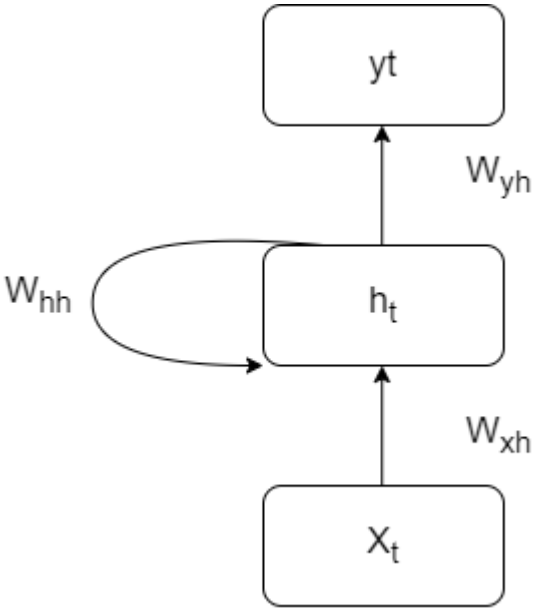


Fig. 3.1. RNN Architecture

-f is the activation function for the hidden state
-W_{hh} is the weight matrix for the hidden state
-h_{t-1} is the hidden state at the previous time step
-W_{xh} is the weight matrix for the input
-x_t is the input at time step t
-b_h is the bias for the hidden state
-y_t is the output at time step t
-g is the activation function for the output
-W_{hy} is the weight matrix for the output
-b_y is the bias for the output

These equations describe RNNs' recurrent nature, where the hidden state at each time step depends on the current input and the previous hidden state. This allows RNNs to capture temporal dependencies in sequential data.

3.1.11 SVM

Support vector machines (SVMs) are powerful supervised machine learning algorithms that excel in classification and regression tasks. They work by finding an

optimal hyperplane in the feature space, essentially a multidimensional dividing line separating different classes [148]. The core principle maximises the margin—the distance between the hyperplane and the closest data points (support vectors) from each class. A wider margin translates to a more robust separation, leading to better performance on unseen data.

One key strength of SVMs is their ability to handle high-dimensional data efficiently. Even when the number of features (characteristics) exceeds the number of samples (data points), SVMs can still deliver accurate and generalizable results. This makes them well-suited for tasks involving complex and multifaceted data, such as bioinformatics, image recognition, and text classification.

SVMs can handle both linear and nonlinear classification problems. Linear SVMs use a straight line as the decision boundary. In contrast, nonlinear SVMs employ a kernel function to map the data into a higher-dimensional space where a linear boundary can be found. This flexibility allows SVMs to tackle complex decision boundaries and capture intricate patterns in the data.

Training an SVM model involves finding the optimal parameters (weights and bias) that define the hyperplane. This optimization process balances minimizing classification errors with maintaining model complexity to avoid overfitting (failing to generalize to new data). Regularization techniques are often used to prevent this.

Feature selection is crucial in SVMs, and the importance of each feature is reflected in its weight. By focusing on informative features, SVMs can improve classification accuracy and efficiency. Additionally, SVMs are robust to outliers, as the decision boundary is primarily determined by the support vectors closest to the hyperplane. By excelling in high dimensions and offering linear and non-linear solutions, SVMs solidify their place as powerful tools for diverse machine-learning challenges.

3.1.12 LSTM

RNNs are powerful tools for handling sequential data, where information like sentences or stock prices unfolds over time. However, their ability to remember past information diminishes as they process longer sequences. This fading memory,

known as the vanishing gradient problem, hinders their ability to learn long-term dependencies. For example, when trying to predict the next word in a sentence, it may be necessary to recall a word mentioned much earlier to make an accurate guess [149].

LSTM networks (LSTMs) address this limitation head-on. Unlike RNNs, LSTMs boast a sophisticated memory mechanism that excels at handling sequences of varying lengths. This makes them particularly well-suited for tasks like network traffic analysis, where data packets arrive at different times and can be of different sizes.

The key to LSTMs' success lies in special "gates" that act like memory filters. These gates control the flow of information, allowing LSTMs to remember or forget crucial details from past data points selectively. In network traffic analysis, for example, the gates would focus on relevant information within each packet. This selective memory empowers LSTMs to identify complex patterns and relationships within sequential data, even when the order and timing of elements are critical.

By overcoming the vanishing gradient problem, LSTMs have become powerful tools for various tasks that require remembering information over extended periods. This improvement marks a significant advancement in RNN technology.

But LSTMs offer more than just superior memory. They are adept at handling noise and inconsistencies often present in real-world data. Additionally, LSTMs can effectively process complex data representations where information is spread across multiple elements, a crucial capability for tasks like natural language processing. Unlike RNNs, which are limited to discrete categories, LSTMs can also excel at handling continuous values. Finally, LSTMs don't require a pre-defined number of states like Hidden Markov Models (HMMs), allowing them to learn from past data more flexibly. LSTMs also offer a wider range of parameters for fine-tuning, providing greater control over model behaviour.

An LSTM unit, the core building block of LSTMs, can be considered a smart memory cell. Four interconnected layers work together to manage information flow and memory. Unlike regular neural network layers, these layers have full connections, meaning every neuron is linked to all others in the layer. Figure 3.2 represents the single LSTM cell [150].

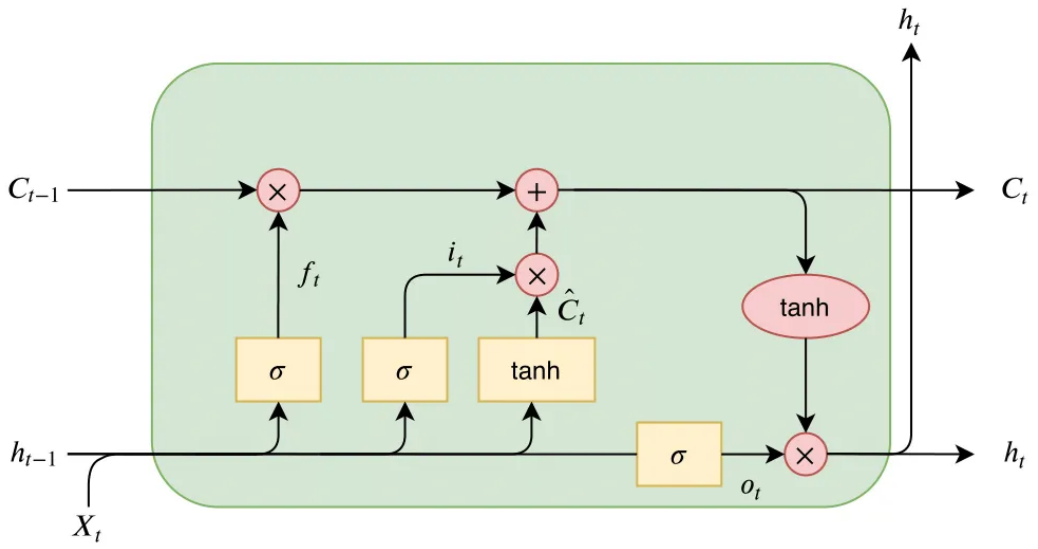


Fig. 3.2. Structure of LSTM Cell

Cell State (C^t): This acts like the LSTM's central memory, a long-term storage unit that can retain information for extended periods across sequences. At each time step, the cell state can be updated with new data, cleared of old information, or accessed for use.

Hidden State (h^t): This layer intermediates between the cell state (storage) and the external world. It retrieves information from the cell state, remembering or forgetting details as needed based on the forget and input gate outputs. The hidden state ultimately produces the final output at each time step.

Input Gate (i^t): This gate controls the flow of new information entering the cell state. Imagine it as a security checkpoint deciding whether to allow incoming data based on relevance. The input gate can selectively accept or reject information based on the current input (X_t) and the previous hidden state h_{t-1}.

Forget Gate (f^t): This gate acts like a clean-up crew, sifting through the information stored in the cell state c_{t-1} from the previous time step. It decides what information to keep and what to discard. This allows the LSTM to forget irrelevant details and free up space for important information.

Output Gate (o^t): This gate functions like a product selection gate at the LSTM's exit. It controls what information from the cell state c^t is ultimately

released as the output h^t. Based on the current input (X_t) and the previous hidden state h_{t-1}, it decides which parts of the stored information are most valuable for the external world.

The following equations represent the core calculations within an LSTM unit at each time step (t). They control the flow of information through gates and update the cell state, ultimately influencing the hidden state output.

$$f_t = \sigma(W_f \cdot [x_t, h_{t-1}, c_{t-1}] + b_f) \tag{3.12}$$

Equation 3.12 calculates the forget gate's activation value f_t at time t. It uses a sigmoid function (σ) to determine how much information to forget from the previous cell state (c_{t-1}) based on the current input (x_t) and the previous hidden state (h_{t-1}).

$$i_t = \sigma(W_i \cdot [x_t, h_{t-1}, c_{t-1}] + b_i) \tag{3.13}$$

Equation 3.13 calculates the input gate's activation value i_t at time t. Similar to the forget gate, it uses a sigmoid function (σ) to decide how much of the new information from the current input (x_t) is allowed to enter the cell state (c_t), considering the context provided by the previous hidden state (h_{t-1}).

$$C'_t = \tanh(W_c \cdot [x_t, h_{t-1}] + b_c) \tag{3.14}$$

Equation 3.14 calculates the candidate memory (C'_t) at time t. It represents the potential new information that could be added to the cell state. The hyperbolic tangent function (tanh) captures the range of this information between -1 and 1. The weight matrix (W_c) and bias vector (b_c) determine how the current input (x_t) and previous hidden state (h_{t-1}) contribute to this candidate memory.

$$c_t = f_t \odot c_{t-1} + i_t \odot C'_t \tag{3.15}$$

Equation 3.15 updates the cell state (c_t) at time t by combining information from the forget gate (f_t), the input gate (i_t), the previous cell state (c_{t-1}), and the candidate memory (C'_t). Element-wise multiplication ($\odot$) allows the forget and input gate values to selectively influence the information retained from the past and the new information added.

$$o_t = \sigma(W_o \cdot [x_t, h_{t-1}, c_t] + b_o) \tag{3.16}$$

Equation 3.16 calculates the activation value (o_t) of the output gate at time t. It uses a sigmoid function (σ) to determine how much information from the current cell state (c_t) is used to create the final hidden state output (h_t). The weight matrix (W_o) and bias vector (b_o) influence the importance of different elements in the current input (x_t), previous hidden state (h_{t-1}), and current cell state (c_t) for this decision.

$$h_t = o_t \odot \tanh(c_t) \tag{3.17}$$

Equation 3.17 calculates the hidden state (h_t) at time t. It uses the output gate activation (o_t) to control how much information from the current cell state (c_t) is passed on as the final output. The hyperbolic tangent function (tanh) ensures the hidden state values are between -1 and 1.

LSTMs overcome the limitations of traditional RNNs by incorporating forget gates, input gates, and output gates that control information flow within the network. These gates allow LSTMs to learn long-term dependencies in sequences and effectively capture temporal information. As a result, LSTMs are well-suited for various tasks involving sequential data, such as speech recognition, machine translation, time series forecasting, and video analysis.

3.1.13 Bi-directional LSTM

LSTM networks have revolutionized deep learning by enabling effective processing of sequential data. However, traditional LSTMs process information only in a forward direction, potentially missing valuable context from previous elements in the sequence. Bidirectional LSTMs (Bi-LSTMs) have emerged as a powerful technique that leverages information from past and future elements within a sequence to address this limitation. This section delves into Bi-LSTMs, exploring their architecture, advantages, and applications [151].

In LSTM, gates work together to selectively update the cell and hidden states at each time step, enabling LSTMs to learn long-term dependencies within sequences. Bi-LSTMs utilise two separate LSTMs working in tandem:

- **Forward Pass:** The first LSTM processes the sequence in the forward direction (left to right), capturing context from past elements. This forward LSTM's hidden states are H_f^t for each time step t.
- **Backward Pass:** The second LSTM processes the reversed sequence in the backward direction (right to left), capturing context from future elements. The hidden states this backward LSTM generates are H_b^t for each time step t.

A core aspect of Bi-LSTMs lies in how they combine the hidden states from both LSTMs. A common approach is a concatenation, where the forward hidden states H_f^t and backward hidden states H_b^t are joined at each time step to create a richer representation a richer representation (H_t) of the sequence:

$$H_t = [H_f^t, H_b^t] \tag{3.18}$$

This concatenated hidden state, (H_t), incorporates information from past and future elements, providing a more comprehensive understanding of the sequence's context.

By processing information bidirectionally, Bi-LSTMs offer several advantages over standard LSTMs:

- **Improved Context Capture:** They can effectively capture long-term dependencies within sequences, even when the relevant information is scattered throughout the sequence (e.g., understanding pronouns based on their antecedents).
- **Enhanced Performance:** In tasks where understanding the full context is crucial, Bi-LSTMs often perform better than standard LSTMs. This is particularly true in natural language processing (NLP) tasks like sentiment analysis, machine translation, and speech recognition.

Bi-LSTMs can incorporate peephole connections. These connections allow the hidden state of the previous time step to influence the current time step directly's forget gate, input gate, and output gate. This can improve the model's ability

to learn long-term dependencies by providing additional context to the gating mechanisms.

Applications of Bi-LSTMs:

Bi-LSTMs find applications in various domains that involve processing sequential data:

- **Sentiment Analysis:** Classifying the sentiment (positive, negative, or neutral) of text data such as reviews or social media posts.
- **Machine Translation:** Translating text from one language to another while preserving meaning and context.
- **Text Summarization:** Generating concise summaries of lengthy documents by capturing the most important points.
- **Speech Recognition:** Converting spoken language into text by capturing the sequence of sounds and their context.
- **Financial Time Series Forecasting:** Predicting future stock prices or other financial metrics based on historical data by analyzing the temporal relationships within the data.
- **Video Analysis:** Recognizing objects and activities within videos by processing the sequence of frames and their visual features.

Limitations of Bi-LSTMs:

While Bi-LSTMs offer significant advantages, they also have limitations:

- Training complexity: Bi-LSTMs typically require more data compared to standard LSTMs due to the increased number of parameters. Techniques like dropout and careful weight initialization are crucial to prevent overfitting.
- Computational cost: Training and running Bi-LSTMs can be computationally expensive, especially with deeper architectures.

Table 3.1: Datasets for insider threat detection

Dataset	Threat types	Description
RUU [152] (2009)	Masquerader	14 masqueraders and 34 normal users
Enron [153] (2015)	Traitor	500,000 emails from 150 employees
Schonlau [154] (2001)	Substituted Masquerader	50 users' Unix shell commands
Greenberg [155] (1988)	Authentication	50 users' Unix C shell commands
TWOS [156] (2017)	Malicious	12 masqueraders users, 5 traitor sessions, and 24 users
CERT (2013)	Malicious	5 insiders and 3995 normal users

3.2 Datasets

This section details the datasets categorized into five groups based on the type of insider activity: masquerader-based, traitor-based, malicious, substituted masqueraders, and authentication-based. Table 3.1 describes the insider threat datasets.

3.2.1 Masquerader-based Datasets

RUU dataset [152] consists of host-based events from 34 regular users, with the help of 14 volunteers who act as masqueraders to look for information with a financial value. RUU is a masquerader-based dataset. Salem and Stolfo introduced the RUU dataset in 2009 and 2011.

Enron dataset [153] comprises 500,000 emails from 150 Enron Corporation employees over the course of five years. It is a traitor-based dataset.

Schonlau dataset [154] comprises 50 users in a substituted masquerader dataset. Each user produces 15,000 Unix shell commands. In a masquerade session, random commands from unknown users are injected.

Greenberg dataset [155] provides complete Unix C shell commands from 168 users in an authentication-based dataset. In contrast to the Schonlau dataset, Greenberg's dataset contains arguments and time stamps in command instances.

TWOS dataset [156] includes a variety of data, both traitors and masqueraders. The dataset comprises behaviours from 24 users over five days that were gathered using a multiplayer game that simulates 12 masquerader sessions and five traitor sessions.

CERT dataset[1] is a synthetic dataset that contains system logs labeled as involving an insider threat. The dataset contains logon, email, http, device, and file access details.

3.2.2 TWOS Dataset

TWOS is a malicious insider threat behaviour dataset based on a gamified competition. A team of researchers from the ST Electronics-SUTD Cyber Security Laboratory at the Singapore University of Technology and Design created the dataset. The competition's goal was to obtain a dataset containing realistic instances of insider threats, a major concern for organisations of all sizes. The TWOS dataset is unique in that it contains data labeled as malicious that was logged as a result of spontaneous user interactions with the workstation.

The TWOS dataset was collected over two years from 24 employees using Microsoft Word on a Macintosh operating system. It contains 74,783 commands corresponding to 11,334 sessions. The data was collected from several host-based heterogeneous data sources, such as mouse, keyboard, processes, and file system. The dataset contains a mixture of normal and malicious activities designed to simulate real-world scenarios.

The gamified competition format was used to improve the quality of the TWOS dataset. The competition was designed to be engaging and fun, encouraging participants to behave more naturally and realistically. It was divided into several rounds, each with a different scenario. Participants were given a set of tasks to complete, and their behaviour was monitored and logged. The scenarios

[1] https://kilthub.cmu.edu/articles/dataset/Insider_Threat_Test_Dataset/12841247/1

were designed to simulate real-world insider threats, such as stealing sensitive data or sabotaging systems.

The TWOS dataset has several potential applications in the field of cybersecurity. It can be used to develop and test new insider threat detection and prevention tools and techniques. The dataset can also be used to train machine learning algorithms to detect and prevent insider threats. The TWOS dataset is unique in that it contains data labeled as malicious and logged due to spontaneous user interactions with the workstation. This makes it a valuable resource for researchers and practitioners interested in insider threats.

3.2.3 ENRON Email

The Enron email dataset is widely used in machine learning and data analysis. It contains approximately 500,000 emails from the Enron Corporation, which went bankrupt in 2001 due to fraudulent business practices. The Federal Energy Regulatory Commission (FERC) initially released the dataset while investigating the company's practices. The Enron email dataset is considered a hallmark for insider threats, counter-terrorism, and fraud detection research.

The Enron email dataset is a valuable resource for researchers because it contains real-world data that can be used to test and validate machine learning models. The dataset includes emails from various sources, including executives, employees, and outside parties. The emails cover a time window of four years, from 2000 to 2002, and provide a rich source of information for analysing email contents to detect insider threat involving collaborating traitors. The Enron email dataset has been used in numerous studies to develop and test machine learning models for identifying insider threats.

Despite its usefulness, the Enron email dataset has some limitations. One of the main challenges is the lack of ground truth labels for the emails. Researchers must rely on unsupervised or semi-supervised machine learning techniques to classify the emails. Another challenge is the sensitive nature of the data, which contains personal and confidential information. To address these challenges, researchers must take appropriate measures to ensure the privacy and security of

the data, such as anonymising the data and obtaining appropriate permissions for its use.

3.2.4 Other Datasets

The Computer Usage Activities Log dataset serves as a cornerstone for research in insider threat detection. Encompassing diverse computer activities like logins, file access, and messaging, it captures user actions in a business environment (often collected via tools like SureView). Researchers leverage this data to identify malicious behaviours (destruction, misuse, etc.) by analyzing known insider attack patterns. The dataset's rich features enable user behaviour profiling and anomaly detection, crucial for identifying potential insider threats. By studying this data, researchers can develop and evaluate algorithms that bolster organizational security against insider attacks [97, 157]. Notably, research by Senator et al. and Gavai et al. exemplifies that these types of datasets are used to develop effective insider threat detection methods.

Another dataset considers attacks on relational database management systems (RDBMS) as a major security concern due to their stealth [158]. Mathew et al. propose a method to identify abnormal access patterns by analyzing query semantics, a more reliable indicator than syntax. Their approach hinges on a historical dataset of queries and their corresponding results. This data allows them to define "normal" access patterns based on the statistical properties of past queries and their outcomes.

3.2.5 CERT Dataset

Data collection is an essential step in cybersecurity. Many publicly available data sources are available to assess insider threat detection models. Although many datasets are available, we use the CERT publicly available dataset. The CERT dataset is "free of privacy and restriction limitations". The insider dataset was proposed by CERT Division, in partnership with ExactData, LLC, and under sponsorship from DARPA I20 [1]. The institute provided ten unique test datasets

[1]https://kilthub.cmu.edu/articles/dataset/InsiderThreatTestDataset/12841247

(r1, r2, r3.1, r3.2, r4.1, r4.2, r5.1, r5.2, r6.1, r 6.2) that include synthetic data for regular and malicious activity. The databases, including user logs on computers, organisational structure, and user data in a directory, simulate corporate environments. The user activity logs include logons, device activity, emails, https, files, and psychometric scores for users. Similarly, the organizational structure and user data directory as in the shape of the lightweight directory access protocol (LDAP). The CERT dataset is generated to closely resemble real-world situations, reflecting the characteristics of user logs. The dataset includes a total of five insider threat scenarios.

1. A user who has not previously used removable drives or work after hours begins logging in after hours, using a removable drive and uploading data to wikileaks.org. Leaves the organization shortly thereafter.

2. A user begins surfing job websites and soliciting employment from a competitor. Before leaving the company, they use a thumb drive (at markedly higher rates than their previous activity) to steal data.

3. A system administrator becomes disgruntled. Downloads a keylogger and uses a thumb drive to transfer it to his supervisor's machine. The next day, he uses the collected keylogs to log in as his supervisor and sends out an alarming mass email, causing panic in the organization. He leaves the organization immediately.

4. A user logs into another user's machine and searches for interesting files, emailing these to their home email account. This behaviour occurs more and more frequently over a 3-month period.

5. A member of a group decimated by layoffs uploads documents to Dropbox, planning to use them for personal gain.

Multiple insider threat datasets exist, with versions denoted by release time. The most common ones are r4.2 and r6.2. Table 3.2 summarizes their key characteristics. In simpler terms, r4.2 is a "dense" dataset containing a significant number of insider profiles and malicious activities. In contrast, r6.2 is a "sparse"

Table 3.2: Comparison for CERT r4.2 and r6.2

	No of Employees	No of Insiders	No of Activities	No of Malicious Activities
r4.2	1000	70	32,770,227	7323
r6.2	4,000	5	135,117,169	470

dataset focusing on 5 identified insiders and activity data for 3,995 regular users between January 2010 and June 2011. Each user record in r6.2 includes roughly 40,000 activity entries.

In this research, we have chosen the CERT r4.2 dataset due to the scarcity of scenario instances in most datasets, where each occurred only once. Dataset 4.2, in contrast, was characterized as a "dense needle" dataset with many cases for each scenario, making it a valuable choice for analysis or modeling tasks that benefit from increased scenario density. The CERT r4.2 dataset has over 20 GB of system log files of 1,000 users, 70 malicious users over 500 days, and both normal and malicious behaviour. The dataset only has 0.03% anomalous incidents and 99.7% normal ones. The r4.2 dataset comprises 930 normal users and 70 malicious insiders. The dataset used in our study consists of several CSV files, each containing specific information related to user activities and attributes. The following are the key CSV files included in the dataset:

1. Device- This file records connecting and disconnecting external devices, specifically USB drives. It contains the date, user, PC, activity (connect/disconnect)

2. Logon- The logon file contains user login and logout times information. The features included in the logon are the date, user, PC, and activity (logon/logoff).

3. File- This file contains logs of user activity on files, such as opening, writing, copying, and deleting files. The file contains features such as the date, user, PC, filename, and content.

4. Email- This file logs employee communication, specifically email exchanges. The email.csv file includes features such as the date, user, PC, To, Cc, Bcc, from, size, attachments, and content.

5. HTTP- The HTTP file captures the user's browsing activity, including the URLs visited. The http.csv contains date, user, PC, URL, and content features.

6. Lightweight Directory Access Protocol (LDAP): The LDAP file contains user information and their job roles. The LDAP file includes features such as Employee_Name, User_Id, Email, Role, Business_Unit, Functional_Unit, Department, Team, and Supervisor.

7. Psychometric - The psychometric file includes information on user personality attributes, specifically the OCEAN model, which stands for openness, conscientiousness, extraversion, agreeableness, and neuroticism. The file includes features such as Employee_Name, User_Id, and personality traits O (Openness), C(Conscientiousness), E (Extraversion), A (Agreeableness), and N (Neuroticism).

3.3 Performance Metrics

Evaluating the performance of ML and DL models in insider threat detection requires a set of comprehensive performance metrics. These metrics evaluate the model's capacity to discern malicious activities from typical user behaviour. This section examines several commonly employed performance metrics for ML and DL methodologies within this research.

3.3.1 Confusion Matrix

The confusion matrix is fundamental for understanding a classification model's performance [159]. It clearly shows the model's ability to identify positive and negative cases correctly.

Table 3.3: Summary of CERT r4.2 dataset

Item	Count
Duration	500 days
Users	1000
Scenarios	3
Logon	854,860
Device	405,381
Email	2,629,980
HTTP	28,434,424
File	445,582
Total Events	32,770,227
Total Threat Events	7,323

TP (True Positive): This refers to the number of malicious samples the model correctly classified as malicious. In insider threat detection, these would be the actual insider threat activities that the model successfully identified.

FN (False Negative): This represents the number of malicious samples the model incorrectly classified as normal. These are the missed detections, where the model failed to identify actual insider threats.

TN (True Negative): This indicates the number of normal data samples the model correctly classified as normal. These are the true negatives, where the model didn't mistakenly flag normal activity as a threat.

FP (False Positive): This represents the number of normal data samples the model incorrectly classified as malicious. These are the false alarms, where the model identified normal activity as a potential threat.

3.3.2 Accuracy

Accuracy is a common way to measure a machine learning model's performance, especially in classification tasks. It essentially indicates how often the model

Table 3.4: Confusion Matrix

Confusion Matrix		Predicted	
		Positive	Negative
Actual	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

makes the correct prediction.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.19}$$

A higher accuracy value (closer to 100%) indicates that the model is making many correct predictions, both in identifying insider threats and classifying normal behaviour. However, accuracy alone can be misleading, especially in situations with imbalanced data. Therefore, accuracy should be used along with other performance metrics that provide more nuanced insights into the model's strengths and weaknesses.

3.3.3 Precision

Precision is a valuable metric used in ML & DL, particularly for classification tasks, to assess the quality of positive predictions.

$$Precision = \frac{TP}{TP + FP} \tag{3.20}$$

A high precision score (closer to 1) indicates that the model identifies real threats and minimises false alarms. Precision focuses on the positives (flagged threats) and their accuracy, not the overall number of correct predictions (like accuracy). It is a valuable metric, especially when the cost of false positives is high. However, precision is often used in conjunction with other metrics.

3.3.4 Recall

Recall focuses on a model's ability to comprehensively detect positive samples, ensuring it catches most of the actual threats present in the data.

$$Recall = \frac{TP}{TP + FN} \tag{3.21}$$

A high recall score (closer to 1) indicates the model excels at capturing most actual threats, minimizing missed threats.

3.3.5 F1-score

In evaluating a model's performance for insider threat detection, it is crucial to consider its ability to identify real threats and minimize false alarms. The F1-score is a metric that balances these two objectives well.

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{3.22}$$

A high F1-score (near 1) indicates the model effectively balances identifying actual threats and avoiding unnecessary alerts. When dealing with imbalanced data, where real threats are rare compared to normal activities, the F1-score provides a more informative performance measure than accuracy alone.

3.4 Summary

This chapter explores the critical security concern of insider threats. It defines various types of insider threats and their activities and motivations. Additionally, it reviews the existing literature on insider threat detection. In the detection field, extensive research has already been conducted on behaviour-based, graph-based, and anomaly detection techniques and more. Even though much research has been done in this field over the past few decades, the research methods and results are still not satisfactory. Moreover, this field can benefit significantly from incorporating recent advancements in machine learning and deep learning, particularly techniques like graph representation and sequential modelling.

Chapter 4

Insider Threat Detection using Supervised Machine Learning

Insider threats pose a constant and critical challenge to organisational security. Unlike external attackers who attempt to breach systems, insider threats lurk within, wielding the privileges granted for their jobs. This inherent trust and access make them particularly dangerous. Recent years have seen a surge in research on machine and deep learning techniques for insider threat detection. This focus is due, in part, to the unique capabilities of insiders.

Many insider threats originate from individuals with substantial technical expertise. This allows them to exploit vulnerabilities and bypass traditional security measures unseen [130, 131, 160]. Insiders know intimately about the organisation's internal networks and security protocols. This knowledge gives them a strategic advantage, allowing them to target specific assets and evade detection [161, 162, 163, 164].

The limitations of traditional security methods in addressing insider threats highlight the need for more sophisticated solutions. Machine learning and deep learning methodologies have emerged as powerful tools in the cybersecurity domain. These techniques are not only effective in identifying insider threats but also in predicting cyberattacks more broadly [45, 46, 47, 48, 49].

Comparing the effectiveness of different algorithms for insider threat detection is a complex task. Existing research utilizes diverse datasets and problem settings, making clear comparisons difficult. Furthermore, the nature of insider threat data itself presents unique challenges. These datasets are inherently imbalanced, with a vast majority representing normal user activity and a tiny fraction reflecting malicious insider actions. This imbalance significantly impacts model performance. Traditional classification algorithms favor the majority class (normal data), decreasing the ability to detect the minority class (insider threats).

Despite these challenges, organisations increasingly use supervised machine learning techniques for insider threat detection. These techniques offer several advantages. Supervised learning can analyse complex patterns in employee behaviour, identifying potential insider threats before they escalate. Additionally, they enable real-time monitoring, allowing organisations to adapt security measures in dynamic IT environments. To address these challenges, this chapter aims to: *(i) Evaluate and compare the performance of various supervised learning algorithms within a controlled setting. (ii) analyse the impact of different imbalanced dataset ratios on supervised learning algorithms.*

The chapter is organised as follows: Section 4.1 presents the related work. Section 4.2 employs the Methodology for detecting insider threats, and Section 4.3 outlines the details of the Experimental settings and results. Following in section 4.4 explains the discussion of the findings, and finally, Section 4.5 concludes the chapter.

4.1 Related Work

Recognizing the inherent difficulty posed by authorized users' access and the challenge of discerning malicious actions from legitimate ones, insider threat detection has progressively turned to machine learning to confront the complexities arising from malicious insiders who misuse their authorized access.

In [165], this paper contributed by proposing a user-centered approach with supervised learning algorithms to identify new malicious insiders. The system

analysed user activity logs and organisational structure to extract relevant features and train classifiers. Evaluation using a public dataset demonstrated the effectiveness of this user-centred approach with high accuracy in detecting novel insider threats. This research aligned with existing work on insider threat detection using machine learning, but it emphasized user-centricity and limited data scenarios.

Haq et al. [166] proposed a hybrid model combining deep learning (GLoVeLSTM, Word2vecLSTM) and machine learning (XGBoost, AdaBoost, Random Forest, KNN, and Logistic Regression). Their study emphasized the importance of insider threat detection due to the high cost associated with insider attacks compared to external threats. The research leveraged a dataset from Enron containing emails and financial information for analysis. They employed pre-trained NLP models (Word2Vec and GLoVe) for word embedding and achieved an accuracy of 92% using XGBoost for insider threat detection. The paper also discussed ethical considerations, data volume, and lack of evaluation frameworks.

Le et al. [167] proposed a user-centered system that leveraged both unsupervised and supervised learning approaches to assist analysts. By learning from limited data on user behaviour, the system aimed to identify previously unseen malicious insiders. Supervised learning helped refine detections with higher precision and lower false alarms. User feedback on alerts further improved performance. The system prioritized user-based reporting to manage analyst workload, considering long-term user behaviour for a comprehensive view.

Yi et al. [168] proposed an approach that leveraged unsupervised outlier scoring functions to identify anomalies and hidden patterns in user data. These outlier scores were then used to create new features, aiding in distinguishing malicious behaviour. This method expanded on previous work by incorporating various unsupervised outlier detection functions and utilizing XGBoost to handle imbalanced datasets, a common challenge in insider threat detection. Additionally, the approach analysed outlier scores at different data granularities and employed Principal component analysis (PCA) to prevent model overfitting.

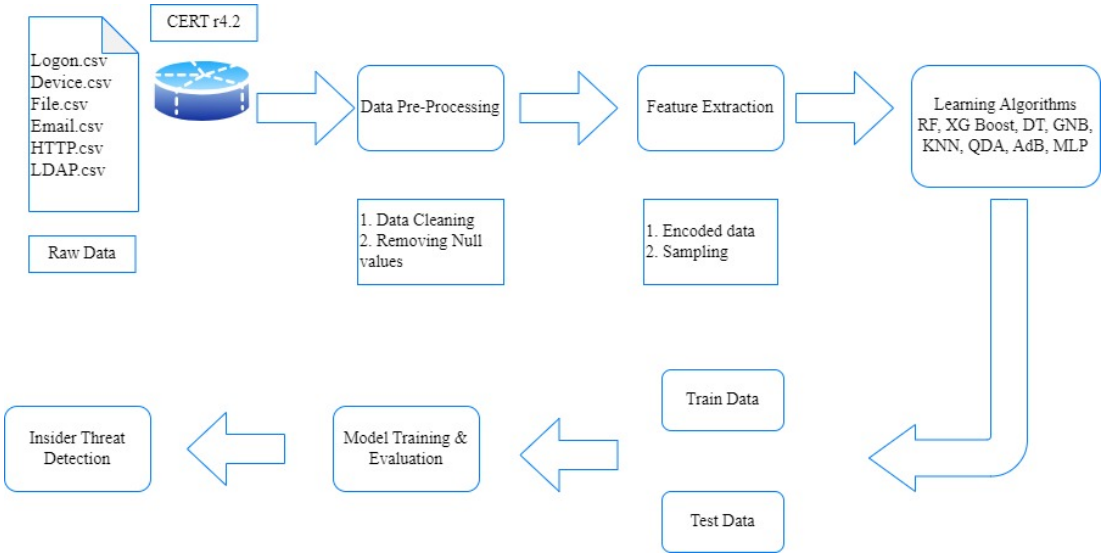


Fig. 4.1. Insider threat detection framework

4.2 Methodology

The evaluation in this approach encompasses eight supervised ML algorithms using the balanced CERT r4.2 dataset. Additionally, various hyperparameters for the KNN, RF, and AdB algorithms within this balanced dataset are explored. Furthermore, the effectiveness of different supervised ML techniques in handling imbalanced datasets is assessed. Specifically, these methods are evaluated under varying degrees of class imbalance, ranging from 40% to 0.5% of insiders. Figure 4.1 outlines the approach.

The CERT r4.2 dataset comprises several CSV files, including device logs, logon details, email activity, HTTP logs, file activity, and LDAP data. Each file contains raw data for every user, which were combined into a master file. From this aggregated master file, a feature set was extracted that includes both text strings and integers. These values need to be properly encoded to be used as input for our proposed approach. The psychometric.csv file is not selected for features, and its ID is not included. Firstly, all the CSV files are merged to create a master CSV file. As the dataset comprises malicious internal attacks, each malicious event was labeled "1", while normal events were assigned a tag of "0". Each row describes a particular event, including the user's name, role, event ID, date,

Table 4.1: List of features and their possible values

Features	Values
Day	0-6
Time	1-24
User_Id	1-1000
Role	1-42
Functional unit	1-6
Department	1-7
PC	Unique number
Activity	1-7

PC ID, type of activity, specific operation details, and attribute details (such as sender, recipient, and email content). Data cleaning removes inconsistencies like null values and duplicates in the pre-processing stage, creating a more reliable master file. This includes imputing missing numerical values using the estimated mean of the relevant feature.

The selected features from various CSV files contain both string and numerical values. However, our algorithm can only process numerical values. Therefore, the input values, such as Day, Time, User_Id, PC, User_Role, User_Functional_Unit, User_Department, and activity features, need to be encoded properly for accurate predictions. A feature's presence is indicated by "1", while its absence is indicated by "0". In terms of data labeling, a user-day is classified as an insider threat if the user has carried out at least one malicious activity on that day. Each day of the week is assigned a number from "0" for Monday to "6" for Sunday.

Logon activity is labeled as "1", and logoff activity is denoted as "2". Similarly, the device connection and disconnection are labeled as "3" and "4", respectively. Email and file activities are represented by "5" and "6", respectively. Finally, HTTP (URL) activity is labeled as "7". Each user has a specific position within the organisation. Table 4.1 mentions the feature values of the dataset.

4.2.1 Handling Imbalanced Datasets

The encoded data is imbalanced, and since it is a large dataset, the downsampling technique is used to address the imbalanced dataset problem. In downsampling, the number of samples is reduced by deleting some of them to achieve a balanced dataset for training the model. The downsampling technique is employed for various levels of class imbalance, such as 40%, 30%, 20%, 10%, 1%, and 0.5% of insiders, to assess the effectiveness of different supervised machine learning techniques in handling imbalanced data using standard evaluation metrics.

4.3 Experiments and Evaluation

This section presents the experimental settings and results. Initially, the balanced CERT r4.2 dataset was employed to assess the performance of machine learning algorithms including RF, XGBoost, KNN, GNB, DT, MLP, AdB, and QDA. Subsequently, the hyperparameters of KNN, DT, and XGBoost were compared to enhance performance. Finally, the performance of various imbalanced CERT r4.2 datasets with status levels of 0.5%, 1%, 10%, 20%, 30%, and 40% was evaluated. The experiments were conducted using Python programming language and Sci-kit learn library. All the experiments were executed on Google Colaboratory.

4.3.1 Experiments on the Balanced Dataset

This experiment compares the performance of several supervised machine learning algorithms, including RF, XG Boost, KNN, GNB, DT, MLP, AdB, and QDA, on the balanced CERT r 4.2 dataset. The dataset contains 32,770,227 events, including 7,323 malicious instances. The performance of these classification algorithms on pre-processed data is evaluated.

For the experiments, the dataset was split into a training dataset comprising 70% of the data and a test dataset containing the remaining 30%. The training dataset was used to train the machine learning models, while the test dataset was employed to evaluate their performance. The balanced dataset was split into a

Table 4.2: Classifiers and their parameters

Classifier	Parameters
RF	n_jobs = -1, n_estimators=100, criterion='gini', max_depth=None, random_state=None
XGB	n_neighbors=5, metric='minkowski',p=2, max_depth=3, loss='log_loss', learning_rate=0.1
KNN	n_jobs=None, n_neighbors=5, p=2, metric='minkowski', algorithm='auto'
GNB	priors=None, var_smoothing=1e-09
DT	max_depth=50, max_features=None, random_state =None,max_leaf_nodes=None, criterion='gini'
MLP	random_state=1, max_iter=300, activation='relu', solver='adam', batch_size='auto'
AdB	base_estimator=DT, n_estimators=9, learning_rate=1.0, random_state=None
QDA	priors=None, reg_param=0.0

70:30 ratio, resulting in a training dataset of 10,252 samples and a test dataset of 4,394 samples.

Precisions and recalls are equally important in a balanced dataset since both classes are equally represented. Therefore, in such cases, the F1 score becomes a valuable metric for evaluating the overall performance of a classifier. A high F1 score indicates that the model balances precision and recall, meaning it can accurately identify positive and negative instances. The F1 score provides a way to compare the performance of different models when both precision and recall are essential.

This experiment utilised the following supervised learning classifiers and their parameters in the balanced CERT r4.2 dataset, as illustrated in Table 4.2. The overall performance of the test data was used in this experiment.

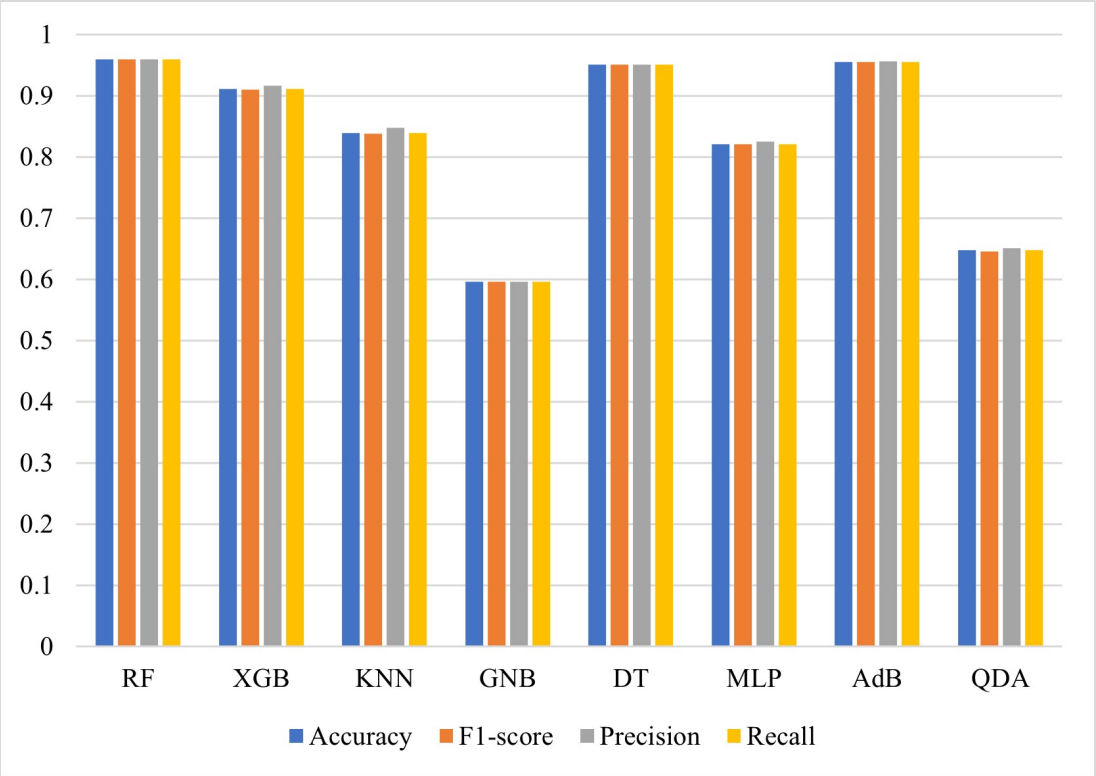


Fig. 4.2. Performance of supervised learning algorithms on a balanced dataset

Table 4.3: Classification performance comparison on a balanced dataset

Classifier	Accuracy	F1-score	Precision	Recall
RF	**0.959**	**0.959**	**0.9598**	**0.959**
XGB	0.9106	0.9103	0.916	0.9106
KNN	0.8393	0.8383	0.848	0.8393
GNB	0.5956	0.5956	0.5956	0.5956
DT	0.9506	0.9506	0.9506	0.9506
MLP	0.8209	0.8204	0.8245	0.8209
AdB	0.9554	0.9554	0.9556	0.9554
QDA	0.6475	0.6453	0.6512	0.6475

Table 4.3 presents the results of eight supervised machine learning classifiers evaluated for insider threat detection on a balanced dataset. The RF classifier achieved the highest accuracy score of 0.959, indicating that it accurately predicted 95.9% of the data points. The F1-score, which measures the balance between precision and recall, is also 0.959 for RF, suggesting high accuracy in both precision and recall. The precision score of RF is 0.9598, indicating that when it makes a positive prediction, it is correct 95.98% of the time. The recall score of RF is 0.959, indicating that it correctly identifies 95.9% of all positive instances. AdB also performs well, with an accuracy of 0.9554, a similar F1-score, and recall. Meanwhile, DT achieved both accuracy and F1-score at 0.9506.

On the other hand, the GNB classifier appears to have performed the worst, with an accuracy, F1-score, precision, and recall score of 0.5956. The QDA classifier also performed relatively poorly, with an accuracy of 0.6475 and a lower F1-score value of 0.6453, precision, and recall scores compared to other classifiers in Table 4.3. Overall, Figure. 4.2 shows that RF and AdB are the top-performing, while KNN, GNB, and QDA had lower performance than the other classifiers on the CERT r4.2 balanced dataset.

4.3.2 Hyperparameter Impact Analysis for AdB, KNN, and DT

This section demonstrates the results of the different hyperparameters for AdB, KNN, and DT on the balanced CERT r4.2 dataset. This experiment utilised the balanced dataset containing 10,252 samples for the training dataset and 4394 samples for the test dataset.

4.3.2.1 AdB Model Results

AdB is a boosting ensemble technique that turns several weak classifiers into robust classifiers. This experiment used various values of the hyperparameter 'n_estimator' ranging from 10 to 50 and the base estimator as DT on a balanced CERT insider threat dataset. The accuracy measures the proportion of correctly classified instances from the total number of instances in the dataset.

The AdB hyperparameter in Table 4.4 demonstrates that the performance of the AdB classifier increases significantly as the number of estimators increases. AdB 30 achieved the maximum accuracy score of 0.9609. The F1-score measures the balance between precision and recall by taking the harmonic mean of both. The F1-score values for the AdB classifier are consistently high, ranging from 0.9581 to 0.9609 for different values of n_estimator. Figure 4.3 shows that as the value of n_estimator increases, the F1-score and precision values decrease. AdB 40 has an F1-score and recall value of 0.9602.

4.3.2.2 KNN Model Results

This experiment used a balanced CERT insider threat dataset with k n_neighbours ranging from 1 to 11 and metric='minkowski', p=2 as the classifier parameters. Table 4.4 presents the performance metrics of the KNN classifier on the dataset. Initially, the accuracy value is 0.8682 while k=1. As the number of neighbors increased, the accuracy values decreased from 0.8505 to 0.8211.

The F1-score values also decreased from 0.8680 for KNN1 to 0.8196 for KNN11. The precision values show a similar trend, with KNN1 having the highest precision value of 0.8713 and KNN11 having the lowest precision value of 0.8322. On

Table 4.4: Performance comparison with different hyperparameters

Classifier	Accuracy	F1-score	Precision	Recall
AdB hyperparameters				
AdB10	0.9581	0.9581	0.9581	0.9581
AdB20	0.959	0.959	0.9591	0.959
AdB30	**0.9609**	**0.9609**	**0.961**	**0.9609**
AdB40	0.9602	0.9602	0.9602	0.9602
AdB50	0.9593	0.9593	0.9593	0.9593
KNN hyperparameters				
KNN1	**0.8682**	**0.8680**	**0.8713**	**0.8682**
KNN3	0.8505	0.8498	0.8566	0.8505
KNN5	0.8391	0.8381	0.8478	0.8391
KNN7	0.8336	0.8324	0.8439	0.8336
KNN9	0.8289	0.8275	0.8399	0.8289
KNN11	0.8211	0.8196	0.8322	0.8211
DT hyperparameters				
DT5	0.7745	0.7699	0.798	0.7745
DT10	0.9176	0.9173	0.9235	0.9176
DT20	**0.9529**	**0.9529**	**0.9529**	**0.9529**
DT30	0.9504	0.9504	0.9504	0.9504
DT40	0.9506	0.9506	0.9506	0.9506

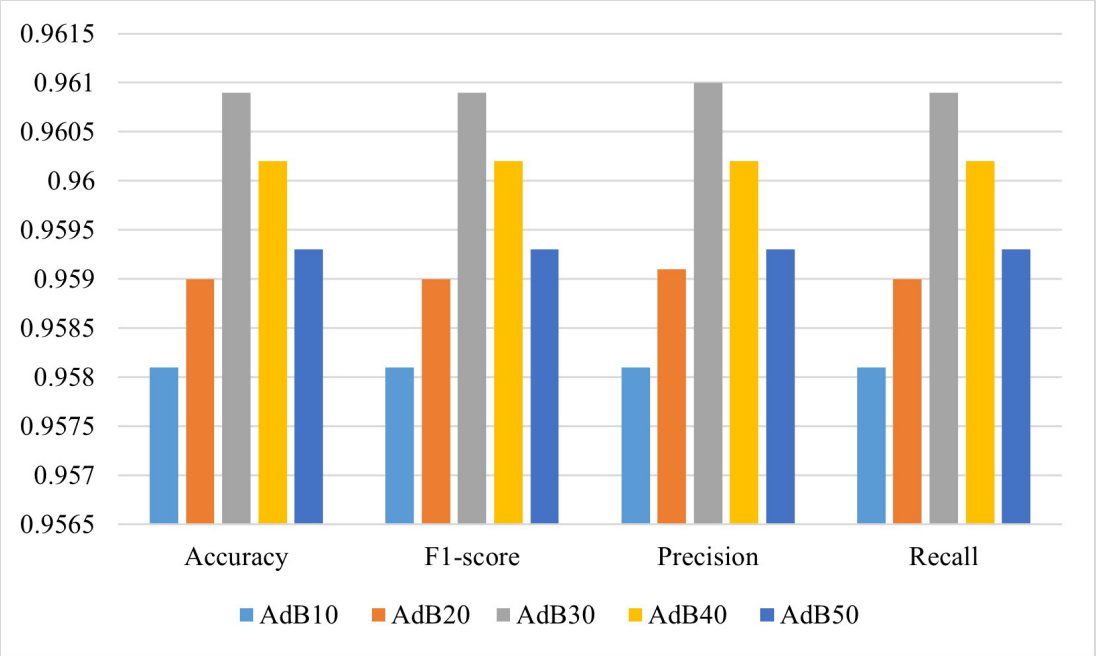


Fig. 4.3. Performance comparison of AdB with different hyperparameters

the other hand, recall values are relatively consistent across all classifiers, with KNN1 having the highest value of 0.8682 and KNN11 having the lowest value of 0.8211. As the number of k increases, the accuracy, F1-score, precision, and recall decrease. The results in Figure 4.4 show that the performance decreases when k=11 with an accuracy of 0.8211, F1-score of 0.8196, and recall of 0.8211.

4.3.2.3 DT Model Results

Table 4.4 provides the performance evaluation of the DT classifier using various maximum depths ranging from 5 to 40. The hyperparameters represent the decision tree's maximum depth, which determines the model's level of complexity.

Table 4.4 reveals that the decision tree with a maximum depth of 20 achieved the highest F1-score of 0.9529, indicating good overall performance. In contrast, the decision tree with a maximum depth of 5 had a significantly lower F1-score of 0.7699. The F1 scores for the decision trees with maximum depths of 10, 30, and 40 were all around 0.95, indicating that the performance of these models is not significantly different

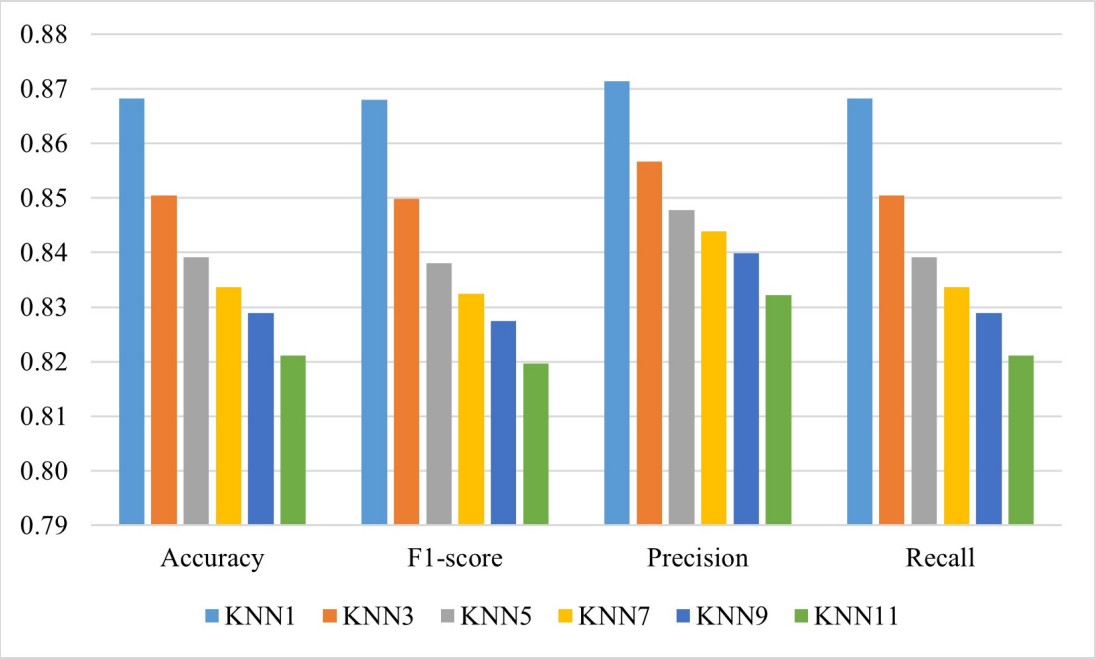


Fig. 4.4. Performance comparison of KNN with different hyperparameters

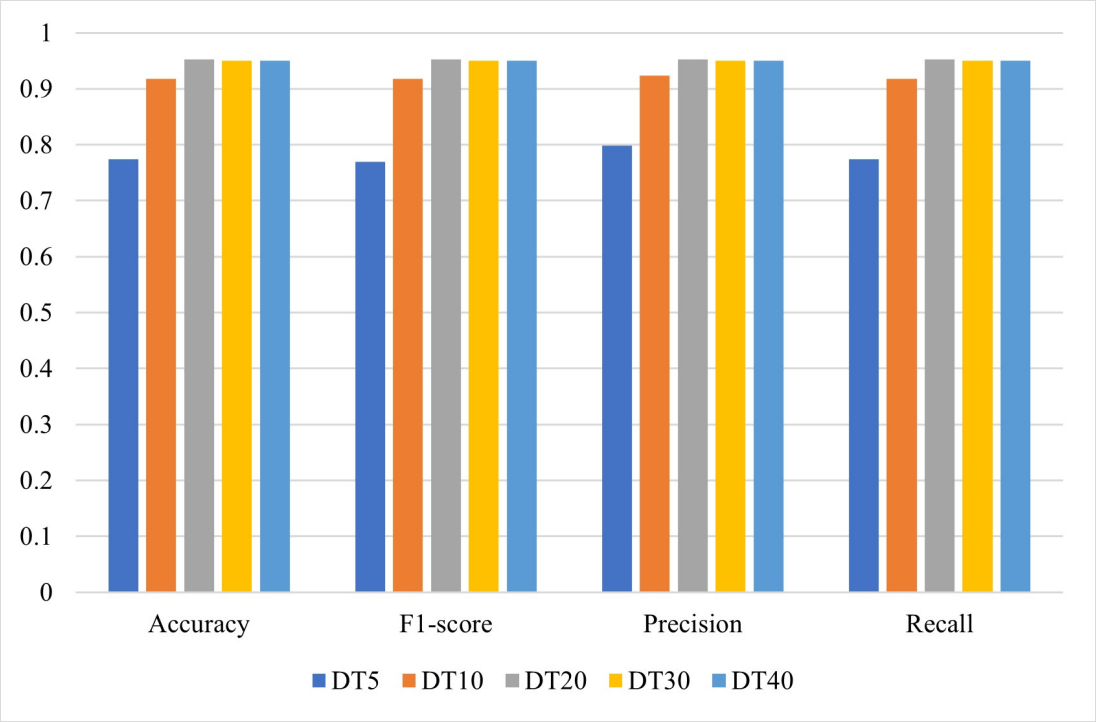


Fig. 4.5. Performance comparison of DT with different hyperparameters

The models' accuracy scores, ranging from 0.7745 for DT5 to 0.9529 for DT20 and DT30, indicate they are effective at classifying the data points.

All models have high precision, indicating that they effectively recognize positive events. As shown in Figure 4.5, the DT10 model correctly predicted 92.35% of all occurrences to be positive, with a precision of 0.9235. Regarding recall, all hyperparameters perform equally well, scoring 0.9504 or 0.9506. DT20 performs well in accuracy, precision, recall, and F1 score.

4.3.3 Experiments on Various Imbalanced Datasets

This section examines the effectiveness of various supervised machine learning approaches for handling imbalanced datasets, which are common in many real-world scenarios. Table 4.5 illustrates the levels of data imbalance in the pre-processed CERT r4.2 dataset, with 30% of the data used for testing and 70% for training. The extremely unbalanced dataset with only 0.50% positive instances has 1,025,219 training samples.

Table 4.5: Sample size details of imbalanced datasets

	50%	40%	30%	20%	10%	1%	0.50%
Training data	10252	12814	17085	25630	51261	512609	1025219
Test data	4394	5493	7323	10985	21969	219691	306932

4.3.3.1 Accuracy for Various Imbalanced data

The accuracy values of different machine learning methods are shown in Table 4.6 for different data imbalance levels, from a balanced dataset to highly unbalanced datasets with only 0.50% of positive samples. Accuracy is a measure of the overall performance of a classification model, representing the proportion of correctly classified instances out of the total number of instances in the dataset.

Table 4.6 shows that RF consistently demonstrated high accuracy, with values ranging from 0.9590 for a balanced dataset to 0.9933 for the imbalanced dataset of only 0.50% positive samples. XGB and KNN established good accuracy for

Table 4.6: Accuracy comparison of various algorithms on imbalanced data

	Balanced	40%	30%	20%	10%	1%	0.50%
RF	**0.9590**	**0.9598**	**0.9537**	**0.9496**	**0.9598**	**0.9903**	**0.9933**
XGB	0.9106	0.9359	0.8914	0.8858	0.9188	0.9902	0.9928
KNN	0.8393	0.8529	0.8655	0.8859	0.9221	0.9874	0.9900
GNB	0.5956	0.6080	0.7038	0.8002	0.9000	0.9900	0.9928
DT	0.9506	0.9481	0.9478	0.9445	0.9577	0.9906	0.9934
MLP	0.8209	0.8358	0.8581	0.8751	0.9210	0.9900	0.9928
AdB	0.9554	0.9552	0.9504	0.9476	0.9590	0.9906	0.9934
QDA	0.6475	0.6738	0.7156	0.8025	0.8985	0.9900	0.9928

moderately imbalanced datasets but struggled with highly imbalanced datasets with only 1% or 0.50% positive samples. GNB and QDA demonstrated poor accuracy for moderately to highly imbalanced datasets, with values ranging from 0.5956 to 0.8985.

For all levels of data imbalance, DT and AdB showed consistently good accuracy, with values ranging from 0.9476 to 0.9577 and 0.9504 to 0.9590, respectively. Employing moderately imbalanced datasets, MLP demonstrated great accuracy but struggled with highly imbalanced datasets. Figure 4.6 illustrates the accuracy for the different imbalanced datasets. The findings in the experiment show that DT, AdB, and RF are suitable for classification tasks using the CERT r4.2 imbalanced datasets.

4.3.3.2 F1-score for Various Imbalanced data

F1 score combines both precision and recall into a single metric. F1 scores range from 0 to 1, with 1 signifying perfect precision and recall and 0 signifying poor precision and recall. A high F1 score implies that the model is performing well in precision and recall, whereas a low F1 score suggests that the model is not performing well in either precision or recall.

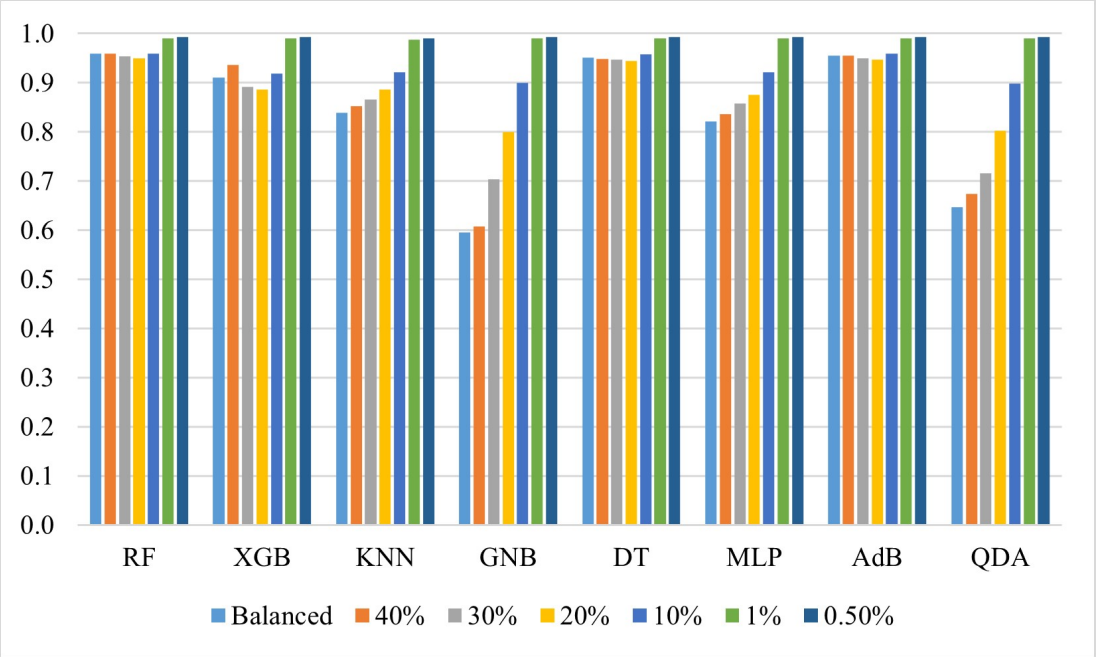


Fig. 4.6. Accuracy comparison of various algorithms on imbalanced data

Table 4.7: F1 score comparison of various algorithms on imbalanced data

F1	Balanced	40%	30%	20%	10%	1%	0.50%
RF	**0.9590**	**0.9507**	**0.9240**	**0.8782**	**0.8040**	**0.3654**	**0.2926**
XGB	0.9103	0.9234	0.8031	0.6511	0.3764	0.0521	0.0009
KNN	0.8383	0.8281	0.7864	0.7221	0.6050	0.2745	0.2279
GNB	0.5956	0.1872	0.0339	0.0018	0.0000	0.0000	0.0000
DT	0.9506	0.9349	0.9129	0.8619	0.7843	0.3655	0.3010
MLP	0.8204	0.7959	0.7627	0.6729	0.4807	0.0135	0.0027
AdB	0.9554	0.9450	0.9175	0.8694	0.7933	0.3695	0.2934
QDA	0.6453	0.5203	0.2461	0.0936	0.0355	0.0000	0.0000

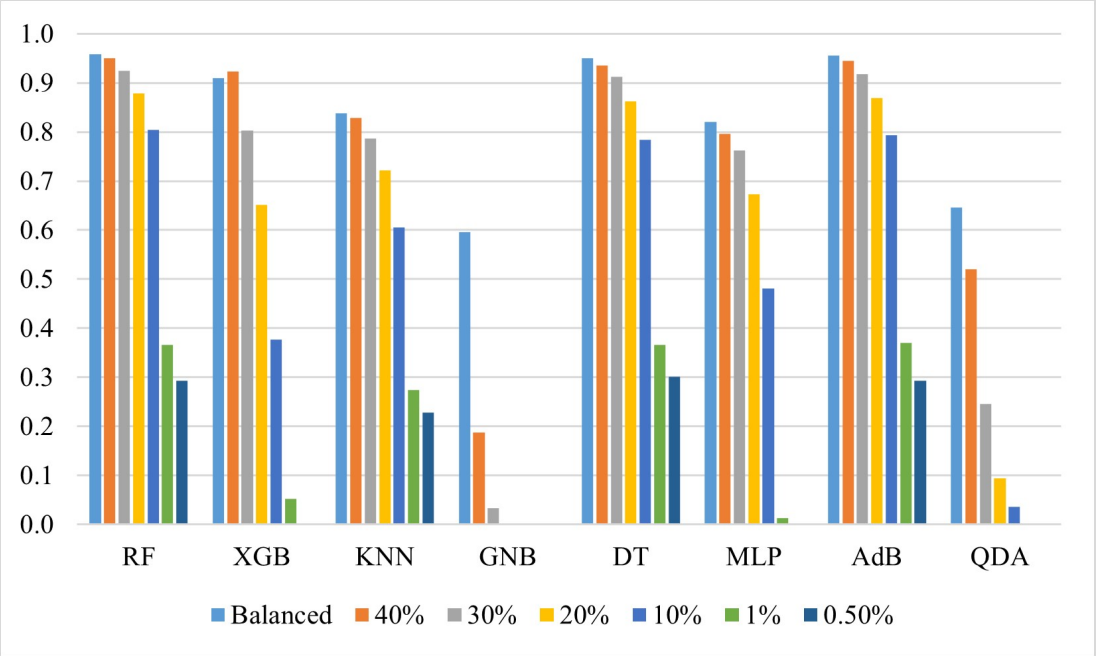


Fig. 4.7. F1 score comparison of various algorithms on imbalanced data

Table 4.7 displays the F1 scores for different classifiers at multiple levels of imbalance, from datasets with a balanced to datasets with just 0.50% of the minority class. With values ranging from 0.9590 to 0.2926, the RF model has the highest F1 score across all imbalance levels.DT has F1 scores that are consistently high across all levels of imbalance, with the range of 0.9506 to 0.3010 but not as high as RF. In datasets with only 0.50% of the minority class, DT achieves the highest F1 score across all algorithms, while still maintaining consistently high scores across all levels of imbalance, ranging from 0.9506 to 0.3010, although not as high as RF

Conversely, the QDA model, with values ranging from 0.6453 to 0, has the lowest F1 scores across all imbalance levels. Figure 4.7 shows that the QDA model cannot perform well in terms of both precision and recall in highly imbalanced data. Similarly, KNN performs relatively better on imbalanced datasets as the level of imbalance increases. On the other hand, XGB has an F1 score of 0.9103 on the balanced dataset, but only 0.0009 for the highly imbalanced dataset. This suggests that as the level of imbalance increases, the F1 score drops significantly.

With the lowest F1 scores across all imbalanced levels, GNB is unsuitable for imbalanced datasets. When the level of imbalance rises, MLP's F1 scores decline, showing that it performs poorly on imbalanced datasets. Overall, the RF classifier was most effective in accurately identifying the samples, whereas the GNB and QDA classifiers performed the least effectively.

4.3.3.3 Precision for Various Imbalanced data

Precision is a measure used to assess the accuracy of a binary classification algorithm. Out of all the positive examples it detects, precision indicates how well a classifier can identify true positive cases. Table 4.8 shows the precision values for each classifier at different levels of class imbalance, ranging from 0.5% to 40% of the minority class.

The RF and AdB models show the best precision values across all imbalance ratios, demonstrating that these models are more accurate at identifying true positives and minimizing false positives. On the other hand, the GNB classifier has extremely low precision scores, particularly at high degrees of imbalance. The low precision score indicates that it is ineffective in identifying positive cases; it either correctly identified all negative cases or failed to identify any positive ones.

At the higher imbalance ratios (40% and 30%), the XGB model also performs well, with high precision values. Compared to the top-performing models, the precision values for the KNN and MLP models are considerably lower at 0.2546 and 0.3750, respectively, as shown in Figure 4.8. Low precision levels show a lack of ability to recognize true positives and a propensity to classify negative events as positive mistakenly.

4.3.3.4 Recall for Various Imbalanced data

The performance of a binary classification model is measured using recall, also referred to as sensitivity or true positive rate. It measures the proportion of actual positive samples that the model correctly identifies. A high recall score indicates that the model can correctly identify most positive cases.

Table 4.8: Precision comparison of various algorithms on imbalanced data

Precision	Balanced	40%	30%	20%	10%	1%	0.50%
RF	**0.9598**	0.9318	0.9103	0.8494	0.7847	0.5327	0.5958
XGB	0.9160	0.8849	0.8810	0.8364	**0.8127**	**0.8676**	**1.0000**
KNN	0.8480	0.7775	0.7510	0.7042	0.6135	0.3250	0.2546
GNB	0.5956	0.5487	0.7917	**1.0000**	0.0000	0.0000	0.0000
DT	0.9506	**0.9381**	0.9150	0.8576	0.7995	0.5602	0.6181
MLP	0.8245	0.7913	0.7654	0.7066	0.7019	0.5172	0.3750
AdB	0.9556	0.9288	**0.9164**	0.8662	0.8006	0.5558	0.6166
QDA	0.6512	0.6316	0.6007	0.5685	0.3596	0.0000	0.0000

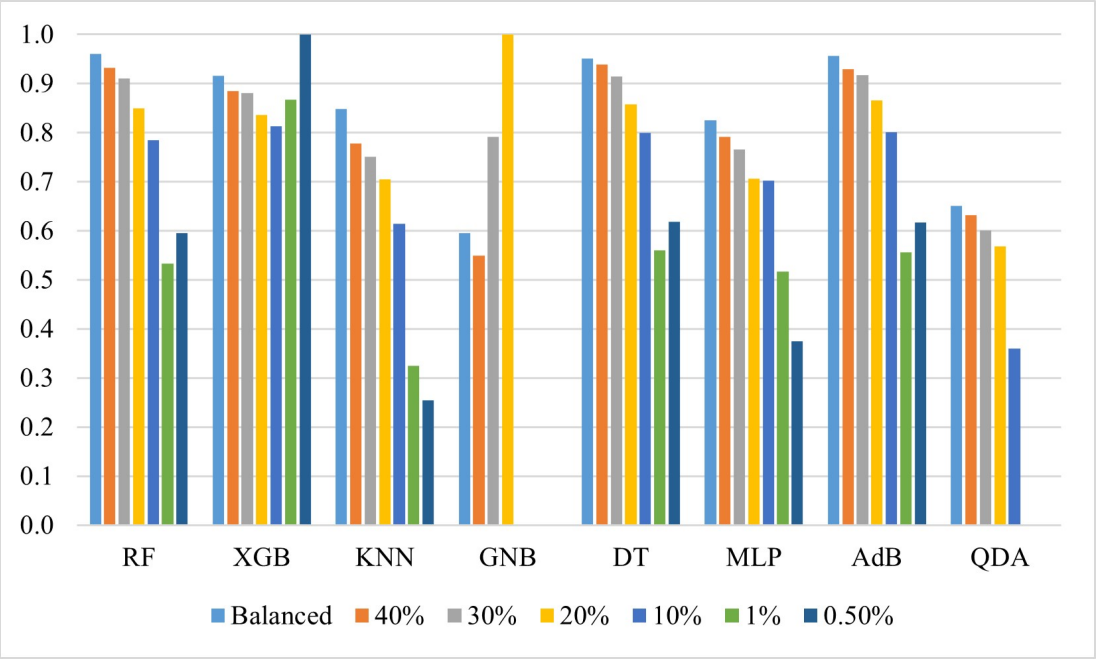


Fig. 4.8. Precision comparison of various algorithms on imbalanced data

Table 4.9: Recall comparison of various algorithms on imbalanced data

recall	Balanced	40%	30%	20%	10%	1%	0.50%
RF	**0.9590**	**0.9704**	**0.9381**	**0.9090**	**0.8243**	**0.2781**	0.1939
XGB	0.9106	0.9654	0.7378	0.5330	0.2449	0.0269	0.0005
KNN	0.8393	0.8858	0.8252	0.7410	0.5967	0.2376	**0.2062**
GNB	0.5956	0.1129	0.0173	0.0009	0.0000	0.0000	0.0000
DT	0.9506	0.9317	0.9108	0.8662	0.7697	0.2713	0.1989
MLP	0.8209	0.8006	0.7601	0.6422	0.3655	0.0068	0.0014
AdB	0.9554	0.9618	0.9185	0.8726	0.7861	0.2767	0.1925
QDA	0.6475	0.4424	0.1548	0.0510	0.0187	0.0000	0.0000

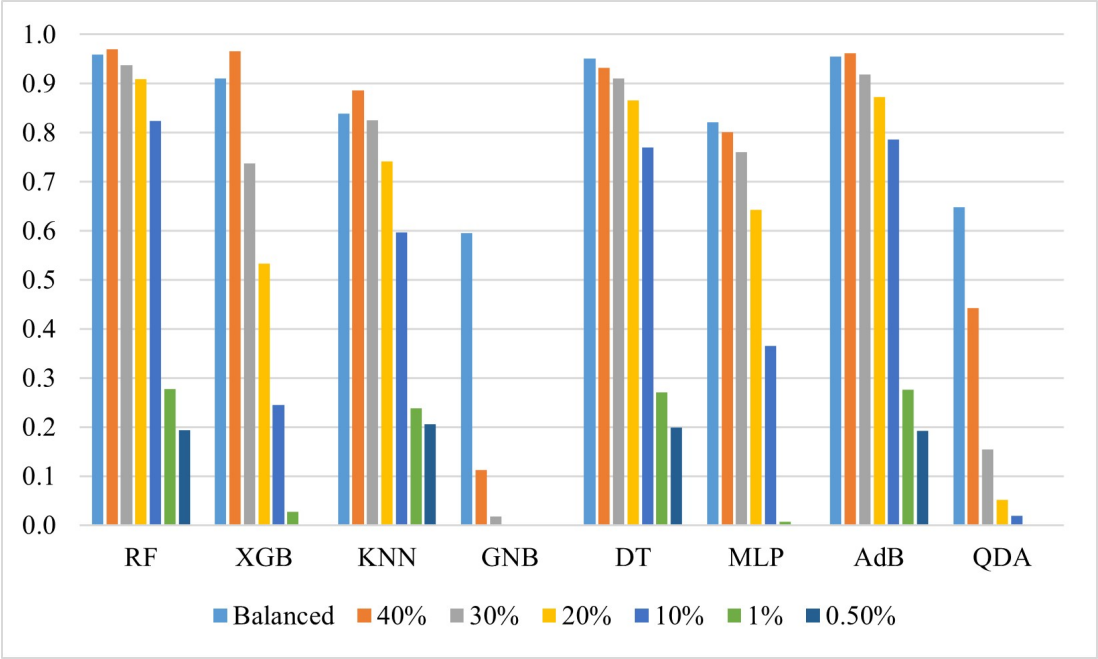


Fig. 4.9. Recall comparison of various algorithms on imbalanced data

The accompanying Table 4.9 reports recall scores for each model at various thresholds, ranging from 0.5% to 50%. The RF and AdB models show significant recall scores across all thresholds, demonstrating their ability to identify a significant portion of positive data accurately in Table 4.9. At a 40% level of class imbalance, the RF model has a recall score of 0.9704, showing that it correctly identified 97.04% of all actual positive cases in the dataset.

However, the recall scores for QDA and GNB models are comparatively low, especially at higher thresholds, and these models may have difficulty appropriately identifying positive samples. Figure 4.9 shows that, at a 1% imbalance level, DT has a recall score of 0.2713, whereas MLP has a recall score of 0.0068. As the level of class imbalance increases, the recall scores generally decrease for all supervised models, as it becomes more challenging to recognize the relatively few positive instances from the vast majority of negative cases.

4.4 Comparison with Existing work

In Table 4.10, a comparison of existing work with the results of the proposed work is presented using performance evaluation metrics, including accuracy, precision, True Negative Rate, Area Under Curve, and False Positive Rate. These results are then compared with those of four established approaches: DNN [169], OCSVM based on DBN [170], LSTM Autoencoder [7], and User behaviour Analysis [171]. The results demonstrate that supervised learning with a balanced dataset in RF achieves the highest accuracy and F1-score of 95.9% compared to the existing works.

Furthermore, it's important to note that, to the best of our knowledge, no existing work has conducted a comprehensive analysis of the impact of hyperparameters on the performance of AdB, KNN, and DT algorithms. Additionally, no extensive research currently investigates the effects of imbalanced datasets with varying class distribution percentages, including 40%, 30%, 20%, 10%, 1%, and 0.5%, on the performance of diverse supervised machine learning algorithms.

Table 4.10: Comparison with Existing work Abbreviations: S-Supervised, U-Unsupervised, M-Method, DV-Dataset Version, A-Accuracy, P- Precision, R-Recall, TNR- True Negative Rate, AUC- Area Under Curve, FPR- False Positive Rate

Approach	M	DV	A	P	R	F1 score	TNR	AUC	FPR
DNN [169]	S	4.2	N/A	N/A	N/A	N/A	N/A	0.944	N/A
OCSVM based on DBN [170]	U	4.2	87.79	N/A	81.04	N/A	N/A	N/A	12.18
LSTM Autoencoder [7]	U	4.2	90.17	N/A	91.03	N/A	90.15	N/A	9.84
User Behaviour Analysis [171]	U	4.2	87.3	84.9	81.7	81.9	N/A	0.89	N/A
Our Approach	S	4.2	**95.9**	**95.98**	**95.9**	**95.9**	N/A	N/A	N/A

4.5 Summary

This chapter explored the application of supervised machine learning algorithms for insider threat detection. The study evaluated various CERT r4.2 balanced dataset algorithms, including KNN, DT, AdB, and others. It investigated the impact of hyperparameter tuning on the performance of KNN, DT, and AdB. The results showed significant improvements with adjustments like modifying the number of neighbors in KNN or the number of estimators in AdB. The chapter then delved into the challenges of imbalanced datasets, which are common in real-world insider threat scenarios. The performance of the algorithms was evaluated on datasets with varying levels of class imbalance, ranging from balanced to a severe imbalance with only 0.5% insider data. The findings demonstrated that Random Forest consistently achieved the best overall performance across all metrics. XG Boost and AdB also performed well, while KNN and MLP struggled with recall scores. Notably, GNB and QDA exhibited poor performance, suggesting they may not be suitable for highly imbalanced datasets like CERT r4.2. In conclusion, this chapter highlighted the importance of considering hyperparameter tuning and class imbalance when applying supervised machine learning algorithms for insider threat detection. Random Forest emerged as the most robust performer across balanced and imbalanced datasets, while other algorithms like KNN and DT exhibited sensitivity to hyperparameter adjustments.

Chapter 5

Bilateral Insider Threat Detection: Harnessing Standalone and Sequential Activities with Recurrent Neural Networks

Insider threat refers to the potential danger posed by employees or individuals within an organisation with access to sensitive information and resources. It is a severe concern for organisations of all sizes and sectors. Insider threats can be intentional or unintentional and may result in data breaches, theft of intellectual property, and damage to an organisation's reputation [172, 173]. The complexity of insider threat lies in the fact that those with authorized access can cause significant security risks. Insider threat is one of the most sophisticated information security threats, and organisations must establish a process for tracking unusual behaviour or potential incidents [10, 126, 174, 175].

analysing user device and application operation logs has emerged as a prominent method in recent research to detect internal threats. This approach is cur-

rently considered the primary method for uncovering potential insider threats [176]. analysing user behaviour patterns allows for detecting deviations that may indicate malicious intent or policy violations. By comparing current behaviour to established normal patterns, it becomes possible to identify and address potential internal threats [173]. Similarly, a supervised machine-learning-based approach was presented in Chapter 4. However, the supervised learning approach has a drawback. The approach used manual or standalone activities only to detect the insider threat detection. To eliminate the issues, this chapter aims to: *propose a novel approach named Bilateral Insider Threat Detection for standalone and sequential activities.*

The primary aims of this chapter can be outlined as follows:

- A bilateral insider threat detection framework was proposed. It combines both standalone activities and sequential activities to enhance the performance of insider threat detection.
- A feature extraction method based on RNNs and LSTM was developed to extract the sequential features of the data.
- Experiments were conducted on the CERT r4.2 dataset to compare the performance of bilateral features using different classifiers, including KNN, MLP, LR, and SVM classifiers. Additionally, the performance of RNN and LSTM feature extractors was compared using the same classifiers, namely KNN, MLP, LR, and SVM.
- Comparing the performance of the proposed bilateral framework against previous insider threat detection methods.

This chapter is organised as follows: Section 5.1 delves into the related work. Section 5.2 applies the methodology to detect insider threats, and section 5.3 delineates the implementation process. In Section 5.4, the experiments are elaborated upon, and subsequently, Section 5.5 draws the chapter to a summary.

5.1 Related Work

Many ideas have been explored, and researchers have proposed numerous techniques to address insider threats [141, 177, 178, 179].

In [180], they have proposed a trust-aware, unsupervised learning approach for Insider Threat Detection. The approach focused on extracting relevant features from system logs and utilizing unsupervised learning algorithms to identify potential insider threats. Additionally, a trust score is assigned to each user based on their anomaly score, taking into account their psychometric score. The study also explored the impact of different system log structures on the effectiveness of the approach.

The paper [181] serves as a tutorial that explains the fundamental concepts of LSTMs and RNNs. It describes the derivation of the conventional RNN formulation from differential equations and discusses the challenges encountered in training conventional RNNs. The article also presents the equations related to the LSTM system and explores ways to further improve it, highlighting new opportunities in the field.

Meng et al. [182] presented a comprehensive framework using LSTM-RNNs for insider threat detection based on attribute classification. The method outperformed KNN, IF, SVM, and PCA-based techniques on the CERT insider threat dataset v6.2. Optimized hyperparameters improve detection rates and reduce false alarm rates.

The authors proposed a multilayer framework for insider threat detection that combines misuse and anomaly detection methods [176]. The framework is evaluated using performance metrics and computation time to effectively detect both known and unknown insider threats.

This paper [183] proposed a novel approach for insider threat detection that leveraged the power of deep learning. The system utilized an ensemble of stacked LSTM and Gated Recurrent Unit (GRU) models with attention mechanisms. These models were trained on user activity sequences and categorized into 282 distinct actions for improved detection performance. To capture the intricacies of user behaviour, the approach employed stacked ensemble models for feature extraction, resulting in a richer representation. Furthermore, the system addressed

the challenge of data imbalance by introducing an equally weighted random sampling technique, ensuring that both normal and malicious activities contributed equally to the training process.

Building upon these prior works, this chapter proposes a novel bilateral approach that combines standalone and sequential features. This combined model leverages the strengths of both architectures for improved feature extraction and detection of insider threats using user activity data.

5.2 Methodology

In this section, we recognize that user behaviours can change over time, with a malicious user often appearing harmless on most days and only showing abnormal behaviour occasionally. As a result, we adopt the standard practice of analyzing user behaviours on a user-day. This section presents a comprehensive demonstration of the complete workflow of the proposed bilateral insider threat detection framework, which aims to detect insider threats.

In Section 5.3, we will delve into the detailed extraction of standalone and sequential activity features, providing a thorough exploration of the methodology. Malicious users exhibit distinct behavioural patterns compared to benign users, which are reflected in their daily activities, such as login frequency, email contacts, files and removable devices. To detect insider threats effectively, we extract standalone activity features from user behavioural log files. The detailed method for extracting these behavioural features is provided in Subsection 5.3.2.

Domain knowledge guides the selection of features for standalone activities on a per user-day basis. The standalone feature matrix, denoted as X_m, is extracted from the daily behaviours of isolated user-days, as depicted in Figure 5.1. This matrix is derived from the daily behaviours of isolated user-days, where $X_m \in \mathbb{R}^{n \times d_m}$. Here, m represents the total number of user-days and d_m represents the dimension of the manual features extracted from standalone activities. For each individual user-day, the extracted feature vector can be represented as $x_i^{(m)} \in \mathbb{R}^{d_m}$, where $i \in \{1, 2, \ldots, n\}$

To handle sequential activities, deep learning models that specialized in processing sequences are designed to extract relevant features through a supervised training procedure. After training, the output for all user-days is used as the feature matrix for sequential activities. Since those features are generated from the daily activity sequence, we denote them as H_s, where $H_s \in \mathbb{R}^{n \times d_s}$, with n representing the total number of user-days, and d_s denoting the dimension of sequential features. For each user-day, the sequential feature vector can be denoted as $x_i^{(s)} \in \mathbb{R}^{d_s}$, where $i \in \{1, 2, \ldots, n\}$.

$$H_s = f_{Seq}(Seq, \Theta_{Seq}) \tag{5.1}$$

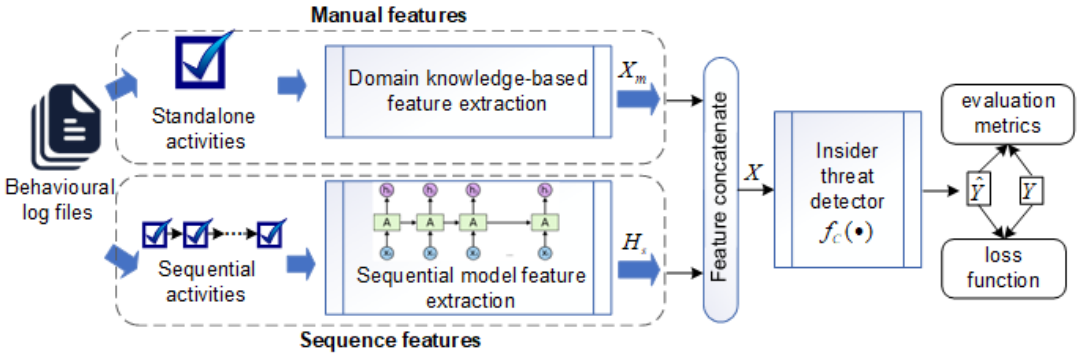


Fig. 5.1. Proposed Framework

The features extracted from the standalone and sequential methods are concatenated to form the final behavioural feature matrix, as illustrated in equation 5.2.

$$X = \text{concatenate}(X_m, H_s) \tag{5.2}$$

where, the final user-day feature matrix for insider threat detection is denoted as $X \in \mathbb{R}^{n \times (d_m + d_s)}$, where the dimension of the final feature is equal to $d_m + d_s$.

5.3 Implementation

5.3.1 Dataset and Pre-processing

In this chapter, we use the CERT r4.2 dataset. Due to the severe class imbalance in the original dataset, we opted to create a smaller, balanced dataset by downsampling the normal users [13]. The resulting dataset was then divided into a training set comprising 70% of the data and a testing set comprising the remaining 30%.

5.3.2 Feature Extraction

To provide a practical example of the proposed solution, we applied it to the CERT r4.2 dataset for insider threat research [184]. This dataset is synthetic, which helps address privacy concerns. We created a balanced sample dataset using the down-sampling technique. This section provides detailed information on the construction of the feature engineering and the execution of the classification task.

5.3.2.1 Manual Features

This paper utilizes two activity files, device.csv, and logon.csv, to extract five behavioural features from users' daily routines. Regarding data labeling, if a user engages in at least one malicious activity on a particular user-day, that user-day is labeled as an insider threat incident. The behavioural features are represented as F1, F2, F3, F4, and F5. For each user-day sample, the following are the five standalone behavioural features listed:

- F1 represents the feature "First logon time," which is extracted from the logon.csv file by mapping the timestamp of the initial login activity to the range of [0, 1] based on a 24-hour basis.
- F2 corresponds to the feature "Last logoff time," derived from the logon.csv file by mapping the timestamp of the final logoff activity to the range of [0, 1] on a 24-hour basis.

- The third behavioural feature, denoted as "F3" or "First device activity time," is obtained from the device.csv file. It involves mapping the timestamp of the initial device activity (connect or disconnect) to a range of [0, 1], considering a 24-hour basis.

- The fourth behavioural feature, "F4" or "Last device activity time," is derived from the device.csv file. It entails mapping the timestamp of the final device activity (connect or disconnect) to a range of [0, 1], considering a 24-hour basis.

- The fifth behavioural feature, denoted as "F5" or "Number of off-hour device activities," is obtained from the device.csv file. It involves counting the occurrences of device activities (connect or disconnect) that transpire during off-hour time. Off-hour time is defined as the period between 18:00 PM and 8:00 AM.

In manual feature engineering, the selection of potential indicators heavily relies on domain knowledge.

5.3.2.2 Sequential Features

To automate the feature engineering process, the first step is to encode the daily activities by assigning each activity a specific numerical representation or code. Once encoded, the activities are arranged into a sequence according to their time of occurrence. This sequential representation preserves the temporal order of the activities, facilitating subsequent analysis and modeling. The approach described in the referenced paper [58] was followed to perform the activities encoding.

We analysed activity logs contained in files such as file.csv, logon.csv, device.csv, and email.csv. Each activity was assigned a specific code based on predefined rules, provided in detail in Table 5.1 for reference. Our study encompassed 12 distinct activity types, and to enhance granularity, we further categorized them based on whether they occurred during working or off hours, resulting in 24 activity types. Each activity type was assigned a unique numerical identifier ranging from 1 to 24.

Table 5.1: Sequential activities Encoding

Nature of the activity	Code for on-duty hours	Code for non-working hours
Logon a pc	1	13
Logoff a pc	2	14
Connect a usb drive	3	15
Disconnect a usb drive	4	16
Open a .doc file	5	17
Open a .exe file	6	18
Open a .jpg file	7	19
Open a .pdf file	8	20
Open a .text file	9	21
Open a .zip file	10	22
Send an email to internal address	11	23
Send an email to external address	12	24

Our sample dataset identified the longest daily activity sequence consisting of 74 activities. Consequently, we selected a sequence length of 74 when training the sequential feature extraction model for feature engineering. The output of this model is a two-dimensional representation, which will serve as the new feature for the sequence of activities.

5.4 Experiments

To evaluate the performance of the proposed bilateral framework, we performed comparative experiments in Section 5.4.1 between the bilateral framework and the standalone activities, and in Section 5.4.2, we compared feature extraction power between RNN and LSTM model. A brief analysis of our work and previous similar work is presented in Section 5.4.3.

All experiments are conducted using the Python programming language. The scikit-learn library was used to implement the binary classifiers. Default parameter settings are adopted for SVM, LR, KNN, and MLP classifiers unless specified

Table 5.2: Performance improvements with bilateral features across classifiers

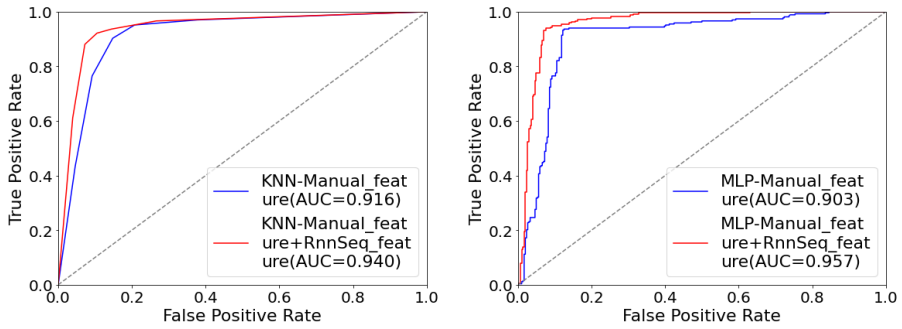
Classifier	Features	Acc	Pre	Rec	F1	ΔAcc	ΔPre	ΔRec	ΔF1
KNN	Manual	87.59	84.32	90.30	87.21	3.15	4.21	1.87	3.10
	Manual+RnnSeq	90.73	88.53	92.16	90.31				
MLP	Manual	89.51	85.14	94.03	89.36	2.97	5.19	0.00	2.78
	Manual+RnnSeq	92.48	90.32	94.03	92.14				
LR	Manual	90.03	86.51	93.28	89.77	2.97	4.80	0.75	2.88
	Manual+RnnSeq	93.01	91.30	94.03	92.65				
SVM	Manual	90.21	86.55	93.66	89.96	2.10	3.16	0.75	2.04
	Manual+RnnSeq	92.31	89.72	94.40	92.00				

otherwise [16, 185, 186].

5.4.1 Comparison between standalone activities and bilateral for different classifiers

To verify the impact of sequential features in our bilateral insider threat detection, we conducted binary classification using four popular classifiers: kNN, MLP, SVM, and LR. By comparing the performance of these classifiers when utilizing manual features versus incorporating sequential features extracted from a plain RNN model, our aim was to assess the improvement achieved by incorporating daily activity sequence in the domain of bilateral insider threat detection, considering its classifier-independent nature [187]. The comparison results between bilateral features and the standalone or manual features in detecting insider threats are presented in Table 5.2.

Table 5.2 presents the performance comparison of different classifiers using various sets of features, including Accuracy (Acc), Precision (Pre), Recall (Rec), and F1-score (F1). The table highlights the effectiveness of these approaches in detecting insider threats. Additionally, it showcases the differences (Δ) in these metrics between using manual features and incorporating the features ex-



(a) ROC Comparison for KNN

(b) ROC Comparison for MLP

(c) ROC Comparison for LR

(d) ROC Comparison for SVM

Fig. 5.2. ROC comparison between manual feature and bilateral feature for different classifiers

tracted from daily activity sequences using a simple RNN model (referred to as "RnnSeq").

The results demonstrate a significant improvement in performance for each tested classifier when incorporating the Manual+RNNseq features together compared to using manual features alone. The comparison results reveal significant improvements in precision and accuracy when incorporating the Manual+RnnSeq features for all tested classifiers. Specifically, precision has increased by 4.21% , 5.19%, 4.80%, and 3.16% for KNN, MLP, LR and SVM, respectively. Similarly, accuracy has increased by 3.15% , 2.97% , 2.97% , and 2.10%, respectively, for KNN, MLP, LR and SVM.

The F1 score, a widely used metric in classification tasks, offers a balanced evaluation of a classifier's performance, taking into account both precision and recall. By combining these measures into a single value, it enables a comprehensive assessment of the classifier's effectiveness.

Adding RnnSeq features into the classifiers' feature sets leads to a significant improvement in the F1 score. Specifically, when combining Manual+RnnSeq features, the KNN classifier demonstrates a 3.10% increase in F1 score compared to using Manual features alone. Similarly, the MLP classifier shows a 2.78% increase, the LR classifier exhibits a 2.88% increase, and the SVM classifier achieves a 2.04% increase in F1 score. These findings indicate that incorporating RnnSeq features enhances the classifiers' performance, as evident from the higher accuracy, precision, recall, and F1 score achieved when compared to using manual features alone.

Figure 5.2 presents the receiver operating characteristic (ROC) curves for all classifiers. The results demonstrate that combining manual features with RNN sequential features (Manual features+RNN seq) outperforms using manual features alone. The area under the ROC curve (AUC) values, which represent the classifiers' overall discriminative power, are provided in Figure 5.2.

For instance, in the case of KNN, incorporating Manual features+RNN sequential features leads to an improvement in the AUC from 0.916 to 0.940. Similarly, for SVM, the AUC increases from 0.914 to 0.966. These results highlight the effectiveness of integrating RNN sequential features with manual features, as

demonstrated by the improved discriminative performance reflected in the ROC curves and AUC values.

5.4.2 Comparison between RNN and LSTM features extractor

In the previous section, we could observe the performance improvement by implementing a simple sequence data-oriented model, which naturally leads to further exploration for better performance with the sophisticated model. We thus compare the LSTM and RNN model's sequential feature extraction power, as presented in the table 5.3.

The table illustrates the accuracy differences between the RNN and LSTM feature extractors for different classifiers. Specifically, the KNN classifier's accuracy improved by 4.37% when utilizing the LSTM feature extractor compared to the RNN feature extractor. Similarly, MLP, LR, and SVM have increased by 2.97%, 2.97%, and 3.32%, respectively. According to precision, the KNN classifier exhibited a 6.92% improvement when utilizing the LSTM feature extractor instead of the RNN feature extractor. Similarly, the LR classifier showed a precision improvement of 5.63% with the LSTM feature extractor. The SVM classifier demonstrated an increase of 6.13% in precision, and the MLP classifier also experienced a 6.13% increase in precision.

Figure 5.3 illustrates the F1-score performance. Comparing the KNN classifier, there was a 4.43% improvement in the F1-score when using the LSTM feature extractor instead of the RNN feature extractor. For the LR classifier, the F1-score showed a 3.01% improvement with the LSTM feature extractor. The SVM classifier exhibited a 3.31% increase in the F1-score, and the MLP classifier demonstrated a 2.99% increase when utilizing the LSTM feature extractor. These results indicate the effectiveness of the LSTM feature extractor in enhancing the F1-score performance across different classifiers.

Table 5.3: Performance comparison between RNN and LSTM feature extractors

Classifier	Feature Extractor	Acc	Pre	Rec	F1	ΔAcc	ΔPre	ΔRec	ΔF1
KNN	RNN	90.73	88.53	92.16	90.31	4.37	6.92	1.87	4.43
	LSTM	95.10	95.45	94.03	94.74				
MLP	RNN	92.48	90.32	94.03	92.14	2.97	5.17	0.75	2.99
	LSTM	95.45	95.49	94.78	95.13				
LR	RNN	93.01	91.30	94.03	92.65	2.97	5.63	0.37	3.01
	LSTM	95.98	96.93	94.40	95.65				
SVM	RNN	92.31	89.72	94.40	92.00	3.32	6.13	0.37	3.31
	LSTM	95.63	95.85	94.78	95.31				

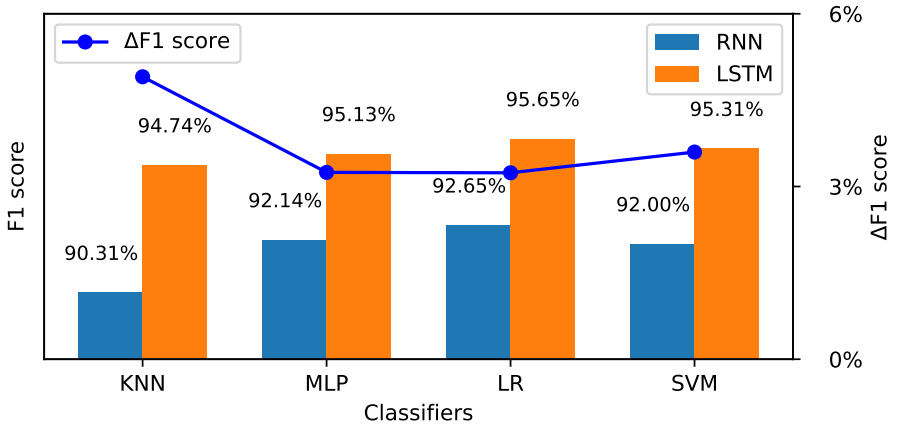


Fig. 5.3. F1-score comparison between RNN and LSTM

5.4.3 Comparison with previous similar work

In insider threat detection research, a challenge arises when comparing previous works fairly. This is mainly due to the complexity of user behaviours and the absence of widely recognized problem settings.

As presented in Table 5.4, we provide a comparison with the most relevant work regarding problem setting. Both studies use a supervised training approach to detect insider users on a user-day basis. Although other classifiers were tested

Table 5.4: Comparison with previous similar work

Classifier	Paper	Feature status	Acc	Pre	Rec	F1
LR	Previous work (Wei, H. et al.)	bi-channel	91.26	88.93	92.91	90.88
	Our work	Manual+RnnSeq	93.01	91.30	94.03	92.65
		Manual+LstmSeq	95.98	96.93	94.40	95.65
SVM	Previous work (Wei, H. et al.)	bi-channel	91.61	89.01	93.66	91.27
	Our work	Manual+RnnSeq	92.31	89.72	94.40	92.00
		Manual+LstmSeq	95.63	95.85	94.78	95.31

in each work, we specifically compared the performance of LR and SVM as the final detection stage classifiers.

In the work conducted by Wei et al., they employ a more complex feature extraction method based on graph neural networks (GNN). However, considering the complexity of graph construction and the GNN model, the performance gain from their approach appears to be overshadowed. As shown in the table, even with our inferior feature extractor (RNN), our work performs better than the comparative study.

It is worth mentioning that Wei et al. improved without introducing new user activity information, attributing it to discovering hidden connections within an organisation. In contrast, our work adopts a comprehensive feature engineering process that leverages the characteristics of daily activity sequences. Therefore, we consider our approach more practical and focused on detection performance.

5.5 Summary

This chapter proposed a bilateral insider threat detection approach that incorporates both standalone and sequential activities to enhance the performance of insider threat detection. Experimental results on the open-accessed CERT 4.2

dataset demonstrate that our Bilateral approach outperforms algorithms that rely solely on features extracted from standalone activities. The experiments also show that combining manual and sequential features improves the overall performance compared to using manual features alone. Furthermore, when comparing the performance of the RNN and LSTM feature extractors, it is observed that LSTM achieves better results than RNN. However, it is important to acknowledge the limitations of the study, which can be addressed in future research. One such limitation is the imbalanced learning challenge in detecting insider threats, as real-world scenarios often have a higher proportion of benign user-days. To address this, a balanced dataset was created by randomly downsampling the majority class for the experimental evaluation.

Chapter 6

Optimising Insider Threat Prediction: Exploring BiLSTM Networks and Sequential Features

The evolving threat landscape of cybersecurity demands a paradigm shift. As technology advances and malicious actors become more sophisticated, insider threats are poised to become even more complex and challenging to detect. A recent report from the Ponemon Institute underscores this critical need. According to their 2023 Cost of Insider Risks Global Report, the total average cost of insider threat incidents skyrocketed by 95% between 2018 and 2023. These figures highlight a troubling trend and emphasize the growing financial burden of insider threats on organisations. In previous chapters 5, 4 discussed insider threat detection under various features and imbalanced dataset ratios. However, relying on techniques that primarily focus on detecting threats after they occur may result in delays in corrections, particularly for large or critical organisations, posing significant risks.

In the past few years, various approaches and techniques have been suggested

to identify and address insider threats. In [55, 56, 57, 188], machine learning-based insider threat detection techniques have been proposed to identify insider threats. Similarly, [58, 59, 60] proposed user behaviour-based insider threat detection techniques. Generally, most academic researchers have primarily concentrated on finding and identifying threats that have already occurred. They usually start by examining the recorded actions of users, extracting important information, and then using a detection model to differentiate between harmful actions and those that are normal. Therefore, a proactive approach should be advocated, emphasising prevention over mere detection. To overcome the drawbacks, this chapter objective is *to propose a novel, standalone, sequential approach to insider threat prediction based on users' daily activities.* To ensure accurate threat prediction, include the day's ground truth (whether there was malicious activity or not) as a feature in our model. This approach leverages a BiLSTM architecture to analyse a user's behavioural patterns in the days leading up to a specific day, allowing us to predict potential malicious behaviour on that particular day.

The main objectives of this chapter can be summarized as follows:

- Introduce a comprehensive framework for insider threat prediction that leverages user activity features, including the ground truth of each day. This framework addresses the challenge of accurately identifying potential insider threats by considering both standalone and sequential user activity data from previous days.

- Conduct a systematic evaluation to assess the impact of integrating standalone features Xm, sequential features Xs, and the ground truth for a specific day Xg on insider threat prediction accuracy. This comprehensive assessment involves a comparative analysis of the performance of three distinct models: RNN, LSTM, and Bi-LSTM on $Xm \| Xs \| Xg$.

- Investigate the impact of varying predictive lengths on Bi-LSTM's ability to predict threats. Our goal is to identify the optimal length that maximises Bi-LSTM's efficiency in threat prediction. It is achieved by comparing its performance with other models (KNN, LR, AdB, GNB) across different predictive lengths.

- Additionally, explore the impact of various embedding sizes (16, 32, 64, and 128) on a BiLSTM architecture with a fixed sequence predictive length (e.g., 5). All models are evaluated using the combined feature set $Xm \| Xs \| Xg$).

The chapter is structured as follows: Section 6.1 discusses related work on prediction-based insider threats. Following that, Section 6.2 outlines the methodology employed in this study. The implementation and discussion of the proposed methodology are detailed in Section 6.3. Section 6.4 showcases the experimental results on insider threat prediction, and the chapter concludes with a summary in Section 6.5.

6.1 Related Work

Insider threat prediction has garnered significant attention recently as organisations strive to safeguard sensitive information from malicious activities within their ranks. In this context, numerous studies have delved into various aspects of insider threat detection, employing multiple methodologies and techniques to address the unique challenges posed by individuals with legitimate access to an organisation's systems and information. This section provides an overview of previous studies on bilateral insider threat detection, highlighting both conventional approaches and some previous methods for insider threat prediction.

Manoharan et al. [61] introduced a novel framework for insider threat detection called a "bilateral" approach. This framework combined standalone and sequential activities using RNNs to improve insider threat detection by leveraging insights from user behaviours. It extracted behavioural traits from log files representing standalone activities and utilised RNN models to capture features of sequential activities. Features from both methods were then concatenated to form a final behavioural feature matrix, allowing for a comprehensive analysis for improved security measures. Experiments on the CERT 4.2 dataset demonstrated the effectiveness of this bilateral approach in detecting insider threats. It outperformed traditional methods that focused solely on standalone or sequential activities.

The proposed approach in [189] focuses on developing an Insider Threat Prediction Model that combines user taxonomy, psychological profiling, real-time usage data analysis, and decision algorithms to identify potentially dangerous users within an organisation. The model collects user characteristics and usage information from the IT components of the system to assess the risk level posed by each user. By categorizing users based on system roles (Novice, Advanced, Administrator) and analysing their behaviour patterns, stress levels, and predisposition to malicious actions, the model aims to predict insider threats effectively.

The paper [190] introduces a novel problem setting in insider threat research, shifting the focus from detecting threats to predicting them based on historical behaviour. By analysing user behaviour in the days leading up to a specific day, the goal is to forecast potential malicious activities in advance. The study utilized the CERT 4.2 dataset and tested various machine learning and deep learning models, finding that deep learning models did not consistently outperform machine learning models for this task.

Wei et al. [191] propose a proactive insider threat detection method using unsupervised anomaly detection. Unlike traditional methods that identify threats after they occur, this approach focuses on predicting potential insider threats by analysing user activity data. The system identifies deviations from normal user behaviour through a cascaded autoencoder model, flagging anomalies for further investigation.

Our comprehensive literature review reveals a critical gap in research concerning BiLSTM models for insider threat prediction. Specifically, no prior studies have investigated BiLSTM models that leverage a combination of manual features, sequential features, and ground truth labels for daily activity in insider threat prediction.

6.2 Methodology

This section details the methodology employed in our study. We begin by formally defining the problem setting in subsection 6.2.1 and introducing the overall framework in subsection 6.2.2 utilised for our insider threat prediction analysis.

Subsequently, we delve into the process of sequential feature embedding based on the BiLSTM subsection in 6.2.3. Finally, we discussed the various learning algorithms employed in this chapter are mentioned in Chapter 3.

6.2.1 Problem Setting

Traditionally, insider threat research has focused on identifying potential threats by analysing past user behaviour. This involves meticulously examining the characteristics of activities associated with specific users [10, 61]. For instance, because employees typically follow workload patterns aligned with the calendar day, researchers define the insider threat detection task as identifying days where a user might exhibit malicious behaviour. Researchers scrutinise these past behaviours to uncover patterns that might indicate malicious intent. Such analysis contributes to a comprehensive understanding of potential insider threats within organisations.

For example, the detection system might classify such activity as potentially malicious if a user's login duration is significantly longer than in previous days. An additional instance occurs when a user accesses a website significantly different from those frequented by colleagues; in such cases, the detection system may likewise flag it as a potential insider threat. As these detection tasks hinge on actions that have already occurred, their implementation is relatively straightforward but offers limited practical significance.

Our initial research explores a new approach that suggests a strong correlation between a user's recent behavioural patterns and the possibility of malicious activity on a specific day. This hypothesis is based on two key observations:

1. Specific previous user actions can serve as warning signs for potential malicious activities in the present day. For instance, consider a scenario where employees are subjected to continuous overtime demands or workplace hostility. Such stress could manifest as retaliatory actions that the system might flag as malicious. Similarly, if abnormal access attempts compromise the system's security in the previous days, the risk of further misuse on the current day becomes significantly increased.

2. Some past activities involve laying the groundwork for malicious actions in the present. Hackers often face difficulties in directly stealing sensitive data. To overcome this, they may conduct reconnaissance to access accounts with high privileges beforehand. However, these seemingly innocuous activities, such as meticulously mapping the network, identifying vulnerabilities, and creating a hidden access point (backdoor), wouldn't necessarily trigger alarms alone. By analysing a user's past behaviour for these patterns, our system can predict an increased risk of a future attack.

Therefore, our proposed approach can predict malicious intent by analysing a user's behavioural patterns over the past few days. This analysis includes login times, device usage patterns, and deviations from normal behaviour.

Since many organisational employees follow routine work patterns, and the risk of malicious activity can vary daily, we utilise a "user-day" approach similar to the one presented in [10, 61]. Our system then aims to predict the likelihood of a specific user engaging in malicious activity on a particular day by analysing their daily activity logs for the past few days.

6.2.2 Framework

In Figure 6.1, the illustrated framework outlines the user-day based insider threat prediction approach. Initially, behavioural logs were categorised based on various activities, including device, logon, file, etc. However, our new approach shifts the organisation of behavioural logs, emphasising the day for improved threat prediction. This modification entails forecasting whether a user could partake in malicious activities on a particular day, emphasising the pivotal role of behavioural logs in this predictive process. By analysing user behaviour daily, the framework offers a proactive approach to identifying and mitigating insider threats before they materialise.

In the initial stage of our proposed method, we begin by pre-processing the entire dataset. The comprehensive dataset, including all activities, was restructured during this crucial phase. Specifically, all user activities are systematically reorganised into daily activity logs for each user-day. After pre-processing, we

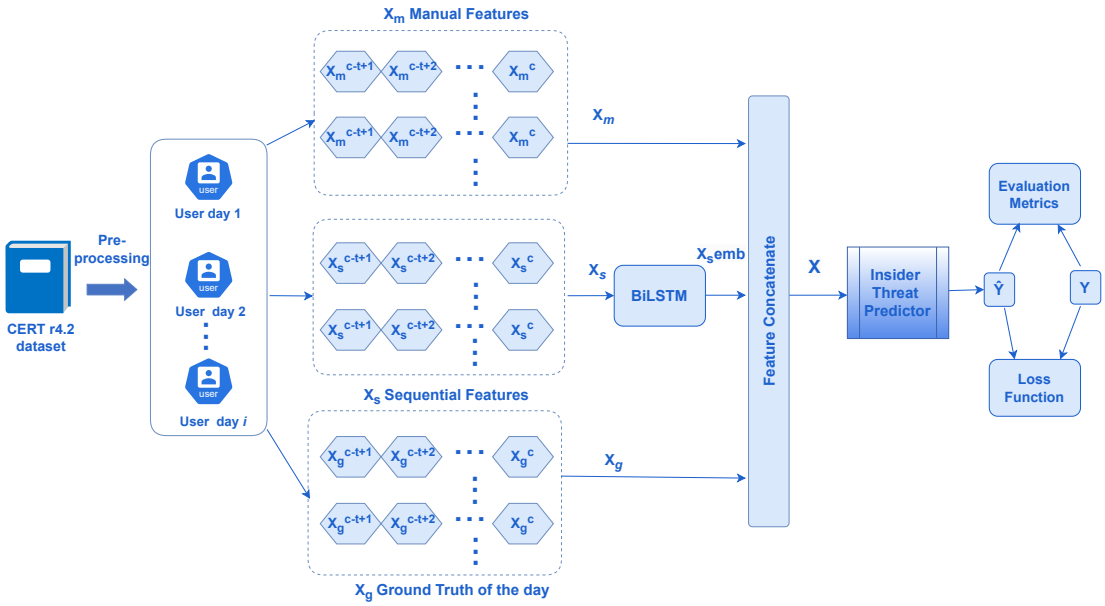


Fig. 6.1. Prediction Framework

extract the manual features Xm, sequential features Xs, and ground truth of the day Xg.

We retrieve activity features from log files that record user behaviour, and the detailed procedure for extracting these behavioural features is outlined in Section 6.3.

The standalone feature matrix, denoted as Xm, is derived from the daily behaviours of individual user-days, where each row corresponds to one user-day, and the columns signify various manual features extracted from standalone activities. The dimensions of Xm are $c \times d_m$, with c representing the total number of user-days and d_m indicating the dimension of the manual features obtained from standalone activities. Each row Xm^c captures the manual feature values for a specific user-day, forming a sequence Xm^{c-t+1}, Xm^{c-t+2}, ..., Xm^c. This sequence delineates the temporal evolution of manual features leading up to and including the target day t.

We use all user-days to obtain a sequential feature matrix from the activities within the daily sequence. Since these features are derived from this sequence, we label them as Xs. The matrix Xs has dimensions $c \times d_s$, where c signifies

the total number of user-days, and d_s represents the dimension of the sequential features. Similar to the manual features, each row Xs^c signifies the sequential feature values for a specific user-day, forming a sequence Xs^{c-t+1}, Xs^{c-t+2}, ..., Xs^c that illustrates the temporal evolution of sequential features.

Xg encompasses the ground truth corresponding to a user-day. This feature sequence, represented as Xg^c, spans the total number of days denoted by c. For a given user-day labeled with t, the associated ground truth values progress from Xg^{c-t+1}, Xg^{c-t+2}, ..., Xg^c. Here, i is an index representing the temporal distance from the user-day labelled t, taking values from t to 0. Consequently, Xg^{c-i} refers to the ground truth value for the specific day i units in the past relative to the target user-day. Each element in this sequence, Xg^{c-i}, where i ranges from t to 0, signifies the actual outcome for the respective day, forming an essential basis for evaluating and training predictive models.

Upon acquiring the reconstructed sequential features Xs, they are fed into the BiLSTM model. The output of the BiLSTM model is represented as Xs_emb. This will be explained in Section 6.2.3.

After receiving the reconstructed data, all the features extracted from standalone activities, sequential activities, and ground truth of the day approaches are joined to construct the ultimate behavioural feature matrix X as in equation 6.1.

$$X = \text{concatenate}(Xm, Xs\_emb, Xg) \tag{6.1}$$

Ultimately, a binary classifier serves as the insider threat detector, and the predicted values of the detector are determined by equation 6.2.

$$\hat{Y} = f_c(X, \Theta_c) \tag{6.2}$$

The equation 6.2 represents the predicted outcomes, indicating whether a user-day is malicious. The equation $\hat{Y}$ represents the predicted labels, f_c denotes the mapping function of the selected classifier, and Θ_c corresponds to the trainable parameters of the classifier. Optimisation of these parameters, Θ_c, can be achieved on the training set by comparing the predicted results $\hat{Y}$ with the actual labels Y and minimising the loss function.

6.2.3 Sequential Feature Embedding based on BiLSTM

BiLSTMs facilitate understanding sequential data Xs by analysing sequences, like user actions, based on their activities. However, preparing the sequential data Xs for BiLSTMs requires careful consideration. Traditionally, one-hot encoding has been the go-to method. This method transforms each unique activity in the sequence into a high-dimensional binary vector. We also performed one-hot encoding for the 24 activity types listed in Table 6.2 before feeding them into the BiLSTM.

One-hot encoding treats all activities as completely isolated entities, failing to capture any inherent relationships between them. It assigns a unique binary vector to each activity in Xs. The vector's dimensionality matches the total number of activities. In this vector, only one element is set to 1 (representing the active element), while all others are 0. As the number of unique activities increases, the one-hot encoded vectors become very high-dimensional, which can lead to overfitting.

To overcome these drawbacks, the embedding layer emerges as a powerful tool. The embedding layer aims to transform the one-hot encoded vectors into denser, lower-dimensional representations called embeddings. This layer acts as a translator between one-hot encoded vectors and a more meaningful representation, converting the activity vectors into embeddings.

These embeddings hold the key to unlocking the hidden relationships between activities. Unlike one-hot encoding, which indicates presence or absence, embeddings encode the semantic meaning of an activity and its connection to others in the sequence. With these embeddings, the BiLSTM can more effectively grasp the context and relationships within the sequence. This deeper understanding improves performance, allowing the BiLSTM to learn more effectively and generalise its knowledge to unseen data. The embedding layer bridges the gap between one-hot encoding's simplicity and the BiLSTM's need for meaningful representations, paving the way for a more robust and insightful analysis of sequential data.

After processing the sequential data Xs through the embedding layer, the BiLSTM takes over. The BiLSTM's output, representing the learned contextual

features of the sequence, is then denoted as Xs_emb.

6.3 Implementation

To implement the proposed BiLSTM-based insider threat prediction system, we leverage the open-source CERT r4.2 dataset. This section details the pre-processing steps, which involve preparing the data for the BiLSTM model. We extract manual features Xm, sequential features Xs, and ground truth of the day Xg.

6.3.1 Datasets and Data Pre-processing

In this research, we conducted all experiments on the CERT r4.2 dataset, which is recognised as the most comprehensive dataset [1]. All the descriptions on the CERT r4.2 dataset are described in Chapter 3. Due to the class imbalance in the original dataset (70 insider threat instances out of 1000 users), we performed additional downsampling on a balanced dataset. This involved pairing each malicious user-day with a benign counterpart, resulting in a downsampled dataset with a size of 1908. The resulting dataset was divided into a 70% training set and a 30% testing set for model evaluation. The paper [61] proposed a standalone and sequential-based insider threat detection method for daily user activity. This chapter adopts the same methodology to extract the daily activities of the user.

6.3.2 Manual Features

This section describes the process of extracting manual features from the CERT r4.2 dataset. We leverage two activity files, device.csv and logon.csv, to derive five features that capture different aspects of users' daily routines. These features are specifically chosen to capture behaviours and patterns in users' daily routines that might indicate potential insider threats within an organisation. In data

[1] https://kilthub.cmu.edu/articles/dataset/Insider_Threat_Test_Dataset/12841247/1

Table 6.1: Description of Manual Features

Manual Features	Description	Content
m1	First logon time	In logon.csv, map initial login timestamp to [0, 1] within 24h.
m2	Last logoff time	In logon.csv, map the final logoff timestamp to [0, 1] on a 24-h.
m3	First device activity time	In device.csv, map the initial device activity timestamp to [0, 1] over 24h.
m4	Last device activity time	In device.csv, map the final device activity timestamp to [0, 1] in 24h.
m5	Number of off-hour device activities	From device.csv, count device activities (connect/disconnect) between 18:00 and 8:00.

labelling, a user-day is classified as an insider threat incident if the user engages in at least one malicious activity on that day.

These features, denoted as m1, m2, m3, m4, and m5, provide insights into potential insider threats by capturing various behavioural patterns. The specific details of these features are listed in Table 6.1.

6.3.3 Sequential Features

To automate feature engineering, we first encode daily activities. Each unique activity is assigned a numerical code. These encoded activities are then arranged into sequences based on their chronological order within a user's day. This sequential format preserves the temporal information of user actions, enabling effective analysis and modelling by the BiLSTM.

The activity encoding approach used in this study is derived from the methodologies described in the referenced papers [10, 58]. We examine activity logs from

various files, including file.csv, logon.csv, device.csv, and email.csv. Each activity is assigned a unique numerical identifier (1-24) based on predefined rules. These rules consider the activity type and whether it occurs during working or off-hours, resulting in 24 unique codes. Table 6.2 shows the detailed mapping between sequential activities and their corresponding codes.

6.3.4 Ground Truth of the Day

This section describes the Ground truth of the day Xg feature, which is incorporated to enhance the prediction accuracy. This feature identifies the specific day users transition from regular activity to confirmed insider behaviour. Unlike other features that capture user actions, the ground truth of the day provides a more precise indicator of malicious behaviour onset through a timestamp.

We leverage timestamps associated with historical data for users identified as insiders. By analysing these timestamps, potentially including individual timestamps for each day within an activity log, we can pinpoint the exact day the user transitioned to insider activity. This specific day is then assigned as the ground truth of the day feature. The ground truth of the day captures the date a user's behaviour deviates from normal patterns, transitioning from regular users to confirmed insiders. This feature is represented as a binary value: 1 for confirmed insider activity and 0 for normal user behaviour.

6.4 Experiments

This section details the experiments conducted in Python's Jupyter Notebook environment to evaluate the effectiveness of the proposed insider threat prediction framework. We perform comparative analyses using various feature combinations: standalone features Xm, sequential features Xs, standalone features with ground truth $Xm \| Xg$, and all features combined $Xm \| Xs \| Xg$- details in section 6.4.1. Furthermore, section 6.4.2 explores the framework's performance using various RNN models with the combined features $Xm \| Xs \| Xg$. Furthermore, section 6.4.2 explores the framework's performance using various RNN models

Table 6.2: Sequential activities Encoding

Nature of the activity	Code for working hours	Code for off-duty hours
Logging onto PC	1	13
Logging off from PC	2	14
Connecting a USB drive	3	15
Disconnecting a USB drive	4	16
Opening a .doc file	5	17
Opening a .exe file	6	18
Opening a .jpg file	7	19
Opening a .pdf file	8	20
Opening a .text file	9	21
Opening a .zip file	10	22
Sending an email to an internal address	11	23
Sending an email to an external address	12	24

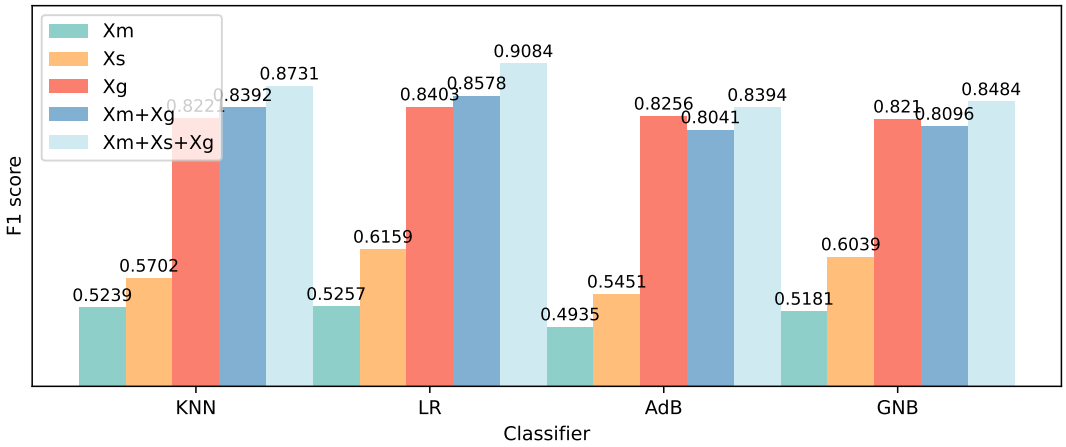


Fig. 6.2. Performance of various sequential features on supervised learning algorithms

with the combined features $Xm\|Xs\|Xg$. Section 6.4.3 investigates the influence of predictive length on performance, identifying the optimal length for improved BiLSTM results. Section 6.4.4 explores the effect of different embedding sizes on BiLSTM performance, using a predictive length of 5 and the combined features $Xm\|Xs\|Xg$.

6.4.1 Comparison of Insider Threat Prediction Models on Various Feature Configurations

Table 6.3 compares various supervised learning algorithms for insider threat prediction using different feature configurations. The evaluated supervised learning algorithms include KNN, LR, AdB, and GNB. Five distinct feature sets are considered: standalone features Xm, sequential features Xs, ground truth of the day Xm, standalone with ground truth of the day $Xm\|Xg$, and a combination of all features $Xm\|Xs\|Xg$. The assessed metrics include Accuracy (Acc), Precision (Pre), Recall (Rec), and the F1 score.

The KNN model achieved its most robust performance (F1 score: 0.8730) by utilising all three feature sets of $Xm\|Xs\|Xg$, which includes individual user

actions Xm, sequential activity patterns Xs, and explicit ground truth information Xg. This combination yielded the highest accuracy (Acc: 0.8766) and F1 score. Even standalone features Xm provided a baseline for prediction (F1 score 0.5240), demonstrating the model's ability to identify potential threats based on user actions alone. Adding sequential features Xs further improved performance (F1 score 0.5702), highlighting the importance of considering the order of activities. Furthermore, Xg achieved an F1 score of 0.8221. Notably, incorporating ground truth information for specific days $Xm\|Xg$ significantly boosted performance (F1 score 0.8391), underlining the value of such information for accurate prediction.

Similarly, focusing exclusively on Xm in the LR model results in an F1 score of 0.5257. Adding Xs improves performance, yielding an F1 score of 0.6159. Furthermore, Xg achieved a significantly higher F1 score of 0.8403. Including ground truth information for specific days, $Xm\|Xg$, significantly improves LR's predictive capabilities, achieving an F1 score of 0.8578. The most resilient performance is achieved when LR integrates all the features $Xm\|Xs\|Xg$, yielding the peak F1 score of 0.9084.

For AdB, relying solely on Xm leads to an accuracy of 0.4961 and an F1 score of 0.4935. However, incorporating Xs or utilising only the ground truth of the day $Xm\|Xg$ enhances AdB's performance, resulting in improvement, although Xg is higher than these features. AdB demonstrates its most resilient performance when using a combination of standalone, sequential, and ground truth features $Xm\|Xs\|Xg$, achieving an accuracy of 0.8425 and an F1 score of 0.8394.

Likewise, GNB's Xm produces an F1 score of 0.5182 and an accuracy of 0.5433. When combining sequential features Xs or standalone with ground truth of the day $Xm\|Xg$, GNB exhibits enhanced accuracy and F1 score, with F1 score values of 0.6039 for Xs and 0.8096 for $Xm\|Xg$. However, Xg achieved a significantly higher F1 score of 0.8210. The combination of standalone, sequential, and ground truth features $Xm\|Xs\|Xg$ delivers the best results; GNB achieves an accuracy of 0.8504 and an F1 score of 0.8484.

In Table 6.3, the feature $Xm\|Xg$ was introduced in [190]. When comparing our proposed work $Xm\|Xs\|Xg$ to existing approaches, our method achieves superior accuracy and F1 score in insider threat prediction. Figure 6.2 shows that

Table 6.3: Performance of various sequential features on supervised learning algorithms

Classifier	Features	Acc	Pre	Rec	F1 score
KNN	Xm	0.5276	0.5255	0.5260	0.5240
	Xs	0.5932	0.5781	0.5717	0.5702
	Xg	0.8241	0.8207	0.8262	0.8221
	$Xm \| \| Xg$	0.8425	0.8392	0.8392	0.8392
	$Xm \| \| Xs \| \| Xg$	**0.8766**	**0.8766**	**0.8705**	**0.8731**
LR	Xm	0.5879	0.5694	0.5470	0.5257
	Xs	0.6457	0.6396	0.6184	0.6159
	Xg	0.8451	0.8446	0.8375	0.8403
	$Xm \| \| Xg$	0.8609	0.8581	0.8575	0.8578
	$Xm \| \| Xs \| \| Xg$	**0.9108**	**0.9108**	**0.9065**	**0.9084**
AdB	Xm	0.4961	0.4963	0.4962	0.4935
	Xs	0.5512	0.5453	0.5459	0.5451
	Xg	0.8294	0.8258	0.8253	0.8256
	$Xm \| \| Xg$	0.8058	0.8034	0.8094	0.8041
	$Xm \| \| Xs \| \| Xg$	**0.8425**	**0.8389**	**0.8399**	**0.8394**
GNB	Xm	0.5433	0.6357	0.5885	0.5182
	Xs	0.6115	0.6037	0.6041	0.6039
	Xg	0.8241	0.8200	0.8223	0.8210
	$Xm \| \| Xg$	0.8110	0.8092	0.8155	0.8096
	$Xm \| \| Xs \| \| Xg$	**0.8504**	**0.8467**	**0.8515**	**0.8484**

overall, LR outperforms other models in the performance comparison, achieving the highest accuracy of 0.9108 and an F1 score of 0.9084 when utilising a combination of standalone, sequential, and ground truth features $Xm \| \| Xs \| \| Xg$.

6.4.2 Performance of Various RNNs

This section presents KNN, LR, AdB, and GNB performance metrics across various RNN models, including RNN, LSTM, and BiLSTM, focusing on $Xm \| Xs \| Xg$ configurations.

Table 6.4: F1 score of various RNN on $Xm \| Xs \| Xg$

Model	Architecture	Acc	Pre	Rec	F1 score
KNN	RNN	0.8425	0.8392	0.8392	0.8392
	LSTM	0.8661	0.8706	0.8559	0.8609
	BiLSTM	**0.8766**	**0.8766**	**0.8705**	**0.8731**
LR	RNN	0.8609	0.8581	0.8575	0.8578
	LSTM	0.9081	0.9093	0.9027	0.9055
	BiLSTM	**0.9108**	**0.9108**	**0.9065**	**0.9084**
AdB	RNN	0.7795	0.7755	0.7795	0.7767
	LSTM	0.8241	0.8203	0.8247	0.8218
	BiLSTM	**0.8530**	**0.8494**	**0.8514**	**0.8503**
GNB	RNN	0.8163	0.8135	0.8193	0.8145
	LSTM	0.7375	0.7556	0.7544	0.7375
	BiLSTM	**0.8504**	**0.8467**	**0.8515**	**0.8484**

Table 6.4 shows that the BiLSTM architecture proves most effective for KNN, achieving the highest accuracy (0.87664) and surpassing both RNN (0.8425) and LSTM (0.8661). Similarly, BiLSTM achieves the best F1 score (0.8730) within the $Xm \| Xs \| Xg$ feature setting. For LR, accuracy increases from 0.8609 (RNN) to 0.9081 (LSTM), and the F1 score improves from 0.8578 to 0.9055. Finally, BiLSTM achieves the highest accuracy (0.9108) and F1 score (0.9084).

Similarly, the AdB model, under the RNN architecture, attains an accuracy of 0.7795 and an F1 score of 0.7767. Switching to LSTM improves the F1 score to 0.8218, followed by BiLSTM with an F1 score of 0.8503. This emphasises the effectiveness of AdaBoost in capturing temporal features. GNB achieves an

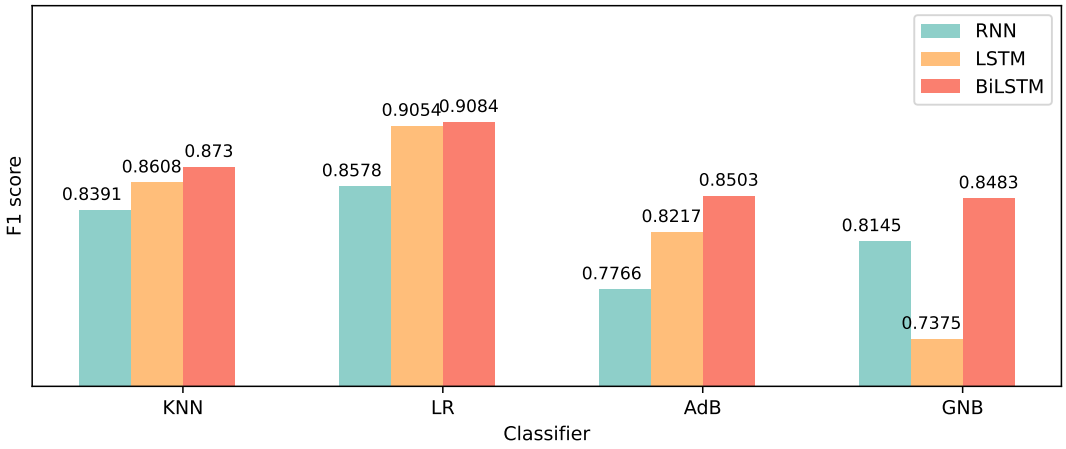


Fig. 6.3. Performance of various RNN on $Xm \| Xs \| Xg$

accuracy of 0.8163 and an F1 score of 0.8145 under RNN. Shifting to LSTM architecture slightly decreases GNB's performance (accuracy: 0.7375, F1 score: 0.7375), suggesting challenges in capturing long-term dependencies. However, the BiLSTM model revitalises GNB's effectiveness, resulting in an accuracy of 0.8504 and an F1 score of 0.8484, demonstrating the model's ability to leverage bidirectional information for improved predictive capabilities.

The Figure 6.3 indicates that the performance in the BiLSTM architecture consistently outperforms others, yielding the highest F1 scores across various supervised learning algorithms. Specifically, under the BiLSTM setting, LR and AdB exhibit F1 scores of 0.9084 and 0.8503, respectively, underscoring the effectiveness of bidirectional long short-term memory in achieving a balance between precision and recall. In contrast, the GNB model demonstrates relatively lower F1 scores, particularly under the LSTM configuration, with an F1 score of 0.7375. Overall, the BiLSTM architecture stands out for insider threat prediction because it understands complicated patterns and relationships over time in user activities.

6.4.3 F1 Score of Various Predictive Length on Bi-LSTM

This section comprehensively overviews the Bi-LSTM model's performance under various predictive lengths. This experiment evaluated four KNN, LR, AdB, and

Table 6.5: F1 score of various predictive lengths on Bi-LSTM

Model	1	2	3	4	5	6	7	8
KNN	0.8806	**0.9086**	0.8895	0.8520	0.8731	0.8522	0.8630	0.8602
LR	0.8689	**0.9138**	0.8898	0.8869	0.9084	0.8662	0.8843	0.8606
AdB	0.8269	0.8697	**0.8850**	0.7949	0.8503	0.8116	0.8308	0.7959
GNB	0.6154	0.7017	0.7682	0.8054	**0.8484**	0.7937	0.8030	0.7740

GNB algorithms across eight predictive lengths. This allows for exploring their performance in diverse temporal contexts with the $Xm \| Xs \| Xg$ feature combination.

The presented Table 6.5 illustrates the F1 scores of various predictive lengths on the Bi-LSTM model, with each model denoted as 1 to 8. In the KNN model, it consistently performs well across all predictive lengths. Notably, the predictive lengths of 2, 5, and 6 yield high F1 scores above 0.9, showcasing the robustness of KNN in various prediction scenarios. Similarly, LR exhibits strong performance, particularly at predictive lengths 2 and 5, with F1 score values of 0.9138 and 0.9084, respectively.

On the other hand, AdB shows a moderate performance, with scores ranging from 0.7949 to 0.8850. Fluctuating scores suggest limitations in handling sequences of different lengths. Furthermore, GNB consistently lags, with F1 scores from 0.6154 to 0.8484, implying challenges with the complexities of predictive lengths.

Figure 6.4 demonstrates that KNN and LR emerge as robust choices, displaying consistent high performance across various predictive lengths. AdB, while effective in a specific length of 3, shows sensitivity to changes in sequence length, and GNB appears less suitable for this particular task based on the observed F1 scores. Predictive length 2 is more suitable than other predictive lengths for achieving effective results in this experiment.

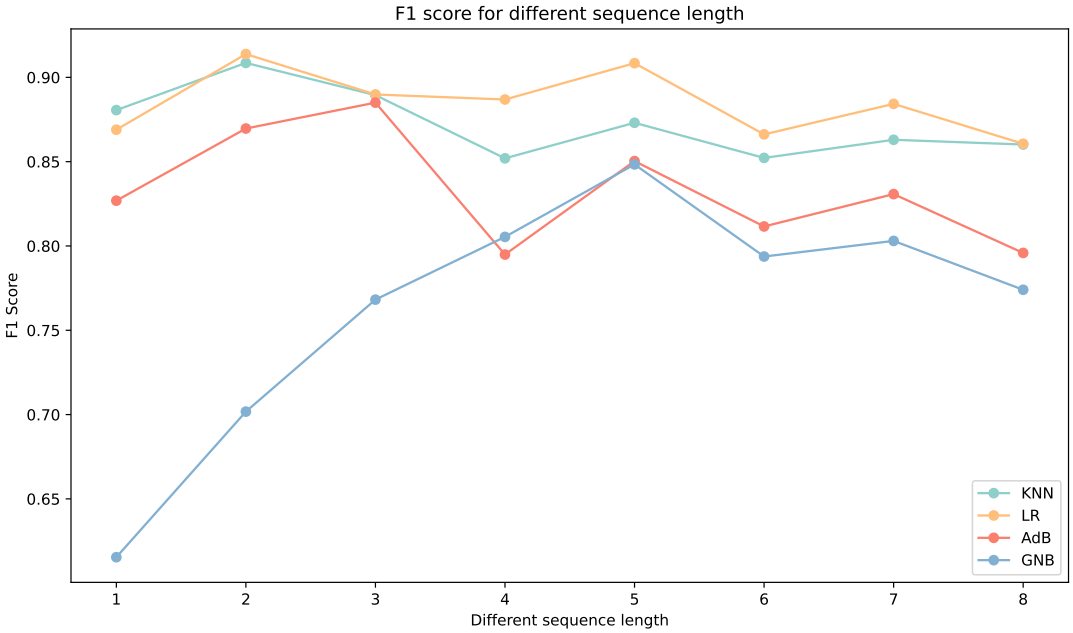


Fig. 6.4. Performance on various predictive lengths on Bi-LSTM

6.4.4 Impact of BiLSTM Embedding Size on Performance

In this experiment delves into the impact of various embedding sizes (16, 32, 64, and 128) on the performance of a BiLSTM architecture with a fixed sequence predictive length of 5. The input features for the BiLSTM model are a combination of Xm, Xs, and Xg, likely representing different data modalities that contribute to the overall prediction task.

The evaluation employs diverse machine learning algorithms: KNN, LR, AdB, and GNB. This allows us to compare the effectiveness of different learning paradigms when dealing with the interplay between embedding size and BiLSTM performance.

The results in Table 6.6 demonstrate a clear trend for most models. As the embedding size increases, so does the F1 score, indicating improved performance. For instance, KNN's F1 score steadily climbs from 0.8392 on an embedding size 16 to 0.8731 on an embedding size 128. Similarly, LR exhibits a consistent rise in F1 score with larger embedding sizes, achieving its peak performance at 128. LR outperforms all other models for all embedding sizes except 16, where KNN

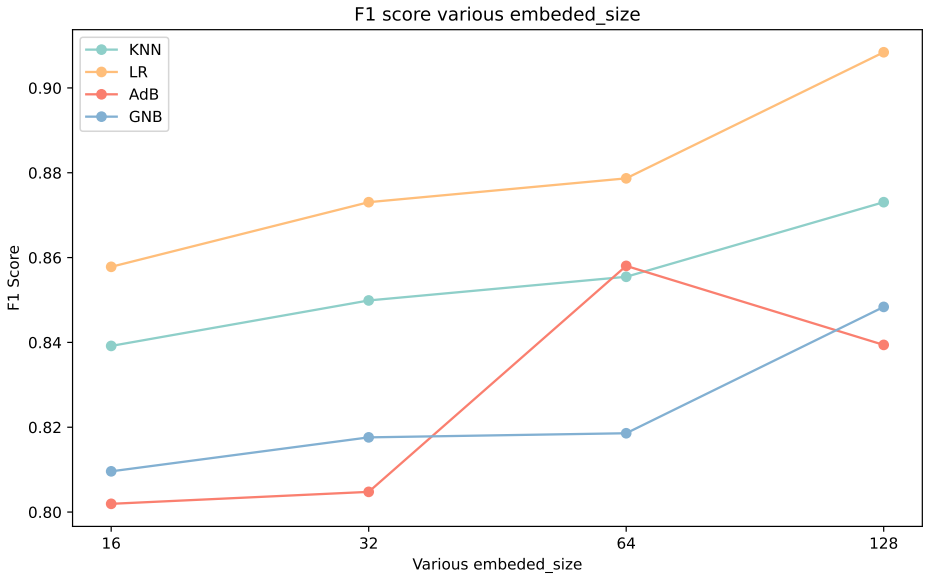


Fig. 6.5. F1 score various embeded_size on BiLSTM

Table 6.6: Performance various embeded_size on BiLSTM

Model	16	32	64	128
KNN	0.8392	0.8499	0.8555	**0.8731**
LR	0.8578	0.8731	0.8787	**0.9084**
AdB	0.8019	0.8048	**0.8580**	0.8394
GNB	0.8096	0.8176	0.8186	**0.8484**

takes a slight lead.

In contrast, AdB presents a contrasting pattern, reaching its peak F1 score at an embedding size of 64 before declining at 128. This suggests that for AdB, a sweet spot exists where a larger embedding size might introduce unnecessary complexity. GNB, on the other hand, follows the prevailing trend, attaining its highest performance with an embedding size of 128. These findings highlight the significance of selecting an optimal embedding size to maximise the BiLSTM model's effectiveness. While Figure 6.5 visually confirms the generally positive impact of larger embedding sizes, the extent of this influence varies across different machine learning algorithms.

6.5 Summary

This chapter tackles the critical issue of predicting insider threats to minimize damage to an organisation's sensitive assets. It proposes a novel approach using a BiLSTM model to proactively identify potential threats before they occur. This method analyses past user behaviour, considering both individual actions and sequences of actions, along with information about the current day. The researchers conducted experiments on the CERT r4.2 dataset, comparing the BiLSTM model's performance across different feature combinations (standalone, sequential, and combined) and against other RNN architectures. They also evaluated the BiLSTM's effectiveness with a specific feature set and its performance under various prediction horizons (1-8 days) and embedding layer dimensions. To ensure a comprehensive understanding, the chapter compared the BiLSTM model with various machine learning algorithms. These comparisons confirmed that the BiLSTM with the combined features achieved superior results in terms of F1 score across all prediction horizons and embedding sizes tested, outperforming other feature combinations and RNN architectures.

Chapter 7

Conclusion & Future Work

This chapter concludes the thesis by highlighting the study's main contributions, discussing its limitations, and exploring promising directions for future investigation.

7.1 Summary of Contributions

Insider threats, a growing worldwide problem, pose a significant risk as authorized users leverage their access for malicious or negligent purposes. Stolen data, sabotage, and difficulty detecting trusted insiders with legitimate access can cause financial loss, reputational damage, and operational disruptions. The global business landscape further intensifies the issue, with remote workforces and diverse locations challenging consistent security protocols. As reliance on technology increases and economic pressures fluctuate, organisations must prioritize measures such as employee screening, access control, data encryption, and incident response plans to mitigate these ever-present threats.

While conventional research on insider threats has primarily concentrated on developing detection algorithms, there has been a recent trend towards exploring the human aspect. Nowadays, cybersecurity experts are increasingly focusing on identifying behavioural patterns and motivations that might predict malicious intent. Additionally, research expands beyond technical solutions, exploring or-

ganisational culture, access control policies, and employee training programs as potential deterrents.

This holistic approach promises a more comprehensive understanding of insider threats, leading to more effective prevention and mitigation strategies. Many types of research are proposed to detect and predict insider threats, such as anomaly-based and behaviour-based. The findings of this research will contribute to the detection and prediction of insider threats with better performance. The first challenge in evaluating insider threat detection algorithms is the absence of standardized datasets and problem settings. Each detection approach uses different datasets and different feature extraction methods. This inconsistency hinders the comparison of different approaches, making it challenging to provide clear recommendations for decision makers.

The second challenge is that many existing works only use standalone activities or sequential activities. The third challenge lies in the limitations of existing research. While previous efforts have focused on identifying malicious insider activities after they occur, they offer minimal assistance in preventing these very risks. Based on the identified limitations of existing research, this study seeks to achieve the following objectives:

- to evaluate and conduct a comparative analysis of supervised machine learning algorithms to assess their suitability for insider threat detection in the same settings.
- to propose a novel method: bilateral insider threat detection to detect malicious insiders. This approach leverages RNNs to analyse both individual activities and their sequences, enhancing overall detection accuracy.
- to propose a novel method for predicting potential insider threats. The approach utilizes bidirectional LSTM networks to capture and analyse individual user actions and the sequences in which they occur. This focus on sequential patterns allows the model to predict the likelihood of an individual transitioning into a malicious insider.

This thesis presents a corresponding solution for each to address the aforementioned objectives. Machine learning and deep learning techniques have emerged

as promising tools for detecting insider threats. Additionally, various monitoring systems such as SIEM and UAM are crucial in user activity surveillance. However, current approaches to insider threat detection face two key challenges: imbalanced datasets, where malicious activities are far less frequent than normal behaviour, and limitations in capturing the sequential nature of user actions.

This thesis proposed a comparative analysis of supervised machine learning algorithms to assess their suitability for insider threat detection within a standardized experimental setting in Chapter 4. Many existing research studies analyse and compare different supervised learning algorithms with various feature extraction methods using diverse datasets. However, this lack of standardization in datasets and evaluation metrics makes it difficult to compare the effectiveness of these approaches and draw definitive conclusions about which method performs best. To address this issue, a comparative analysis of various supervised machine learning algorithms, including RF, XG Boost, KNN, GNB, DT, MLP, AdB, and QDA, was conducted. This analysis is performed using the balanced CERT r4.2 dataset. This technique utilizes the CERT r4.2 dataset and extracts features containing both text and numerical data.

To prepare these features for the machine learning algorithm, text features are converted into numerical values, with "1" representing the presence of a feature and "0" representing its absence. Moreover, no existing work researches the impact of hyperparameters in ML algorithms. This chapter compares the impact of hyperparameter modifications on the performance of the machine learning models KNN, DT, and AdB in the balanced dataset. Similarly, existing in its exploration of the impact of varying imbalanced dataset ratios. This study investigates the performance of various supervised machine learning methods in handling imbalanced datasets, a common challenge in real-world scenarios. We specifically examine how these methods perform under different levels of class imbalance, ranging from a high imbalance of 0.5% insider representation to a more balanced 40% insider representation. Compared with existing works, the results demonstrate a similar trend: performance degradation for all algorithms as the class imbalance level increases. However, this study uniquely identifies random forest as the most resilient algorithm across all imbalanced scenarios.

The aforementioned approaches only use standalone features. Similarly, traditional insider threat detection involves analysing user logs and developing classifiers to categorize individuals as malicious or not. However, these methods only consider individual activities or sequences. A more comprehensive approach is needed to capture the complex and nuanced nature of insider threats. To address the limitations of existing methods, Chapter 5 proposed a novel bilateral insider threat detection framework. This approach leverages both individual user activities and their sequential patterns, leading to a more comprehensive understanding of user behaviour and improved detection of malicious insiders. To capture these sequential patterns, the chapter employed a feature extraction method based on RNNs. RNNs extract sequential features from data, making them well-suited for extracting sequential data. Experiments on the CERT r4.2 insider threat detection benchmark evaluated the performance of various RNNs with differing capabilities for handling sequential data. The experiment compared a standard RNN, an LSTM network adept at learning long-term dependencies, and a BiLSTM that analyses data in both directions for richer context. This evaluation aimed to identify the model best suited to capture the sequential patterns in user activities indicative of malicious insider threats. In this approach, the feature extraction process involved two main steps. First, manual feature engineering extracted five behavioural features from the user activity data in "device.csv" and "logon.csv" files. These features capture daily activity patterns and rely on domain knowledge about suspicious behaviours. The second step focused on extracting sequential features. This step involved transforming user activity sequences into numerical representations suitable for learning algorithms. The experiments on the CERT 4.2 dataset yielded positive results for the proposed bilateral approach. This approach, which analyses both individual user activities and their sequential patterns, significantly outperformed algorithms that rely solely on standalone features. This highlights the importance of considering the order and context of user actions in identifying insider threats.

The aforementioned existing research is constrained by limitations that impede proactive prevention efforts. Previous investigations have predominantly focused on identifying insider threats post-occurrence, providing limited assistance in preemptively thwarting such incidents. To solve this issue, chapter 6 proposed

a comprehensive framework for insider threat prediction, leveraging user activity features, including the ground truth of each day. This framework addresses the challenge of accurately identifying potential insider threats by considering both standalone and sequential user activity data from previous days. Moreover, a systematic evaluation is conducted to assess the impact of integrating standalone features Xm, sequential features Xs, and the ground truth for a specific day Xg on insider threat prediction accuracy. This comprehensive assessment involves a comparative analysis of the performance of three different models: RNN, LSTM, and BiLSTM on $Xm \| Xs \| Xg$. Furthermore, the investigation delved into the impact of varying predictive lengths on BiLSTM's ability to predict threats. Additionally, exploration was undertaken on the impact of various embedding sizes (16, 32, 64, and 128) on a BiLSTM architecture with a fixed sequence predictive length (e.g., 5). All models were evaluated using the combined feature set $Xm \| Xs \| Xg$. In this chapter, the feature extraction process is divided into three parts: manual features, sequential features, and ground truth of the day. Manual features are extracted from device.csv and logon.csv files in the CERT r4.2 dataset. Five features (m1-m5) are derived to capture the user's daily routines. Sequential features capture the order of user activities in a day. Each unique activity is assigned a numerical code based on activity type and working/off-hour occurrence. These encoded activities are then arranged chronologically into sequences. Ground truth of the day Xg is a binary feature indicating a user's transition to insider activity. It is derived from timestamps of confirmed insider historical data and captures the specific day the user's behaviour deviates from normal patterns. The experiment results show that combining all features Xm, Xs, Xg with a BiLSTM model proved the most effective. This approach achieved the highest F1 scores across various testing conditions, consistently outperforming other feature sets and model types.

7.2 Study Limitations

7.2.1 Imbalanced Data

In insider threat detection, a significant hurdle is the class imbalance problem. This arises because most user activity represents legitimate actions (normal class). This creates a lopsided dataset to train detection models. These models inherently prioritize identifying the dominant class (normal activity), making them susceptible to overlooking rare instances of malicious insider activity (minority class). This can lead to a situation where the models perform well at identifying normal behaviour but fail to detect the crucial yet infrequent signs of malicious intent. The models become overly focused on the common and miss the rare, critical threats. Addressing this class imbalance is crucial for building effective insider threat detection systems.

7.2.2 High False Alarm

Insider threat detection systems are plagued by high false alarm rates when configured with excessive sensitivity. This oversensitivity leads to many false positives, where innocent activities are mistakenly flagged as malicious. Security teams become inundated with these false alarms, diverting their focus and resources from investigating genuine threats. The challenge lies in calibrating the system to effectively detect malicious activity without generating overwhelming false alerts.

7.2.3 Lack of Real-world Data

The lack of real-world data hinders the building of strong detection models for insider threats. While helpful, public datasets like CERT are often synthetically generated, creating a gap between simulated and real-world scenarios. These synthetic datasets might contain randomly generated activities that lack the complexity of true insider attacks. This dearth of real-world data makes it challenging to train models that can effectively handle the full range and intricacy of insider threats in the real world.

7.3 Future Work Directions

7.3.1 Imbalanced data

Insider threat detection remains challenging due to the inherent imbalance in available data. Future research should focus on improved data collection strategies. Furthermore, exploring machine learning algorithms suitable for imbalanced data, alongside deep learning and unsupervised learning techniques, holds promise. Human expertise will remain crucial, so integrating machine learning with human analysis and fostering explainable AI is vital. Finally, considering adversarial learning and broader behavioural analysis will be essential to stay ahead of evolving insider threats.

7.3.2 Dataset

The future of insider threat detection hinges on leveraging a wider range of data sources, each with its strengths and limitations. System logs offer a detailed history, but future systems need to glean context from user activity data (keystrokes, mouse movements) while addressing privacy concerns and storage costs. Network traffic analysis will require advanced algorithms to pinpoint unusual data transfers amidst the ever-growing volume. Content analysis, though valuable, needs anonymization techniques to ensure data privacy. Finally, exploring psychological data (surveys, biometrics) holds promise, but ethical considerations necessitate careful interpretation and validation methods. By overcoming these limitations and harnessing the power of diverse data, future insider threat detection can become more robust and proactive.

7.3.3 New Theories

Moving beyond traditional methods, exploring new theoretical approaches like hybrid techniques holds immense promise for insider threat detection. These techniques combine the strengths of different approaches to address the limitations of individual methods. Hybrid techniques offer the following advantages:

Combining Supervised and Unsupervised Learning: Supervised learning excels at identifying known threats, while unsupervised learning excels at detecting anomalies. Combining them allows for leveraging pre-defined threat indicators while also adapting to novel insider tactics.

Enhancing Anomaly Detection with Context: Traditional anomaly detection in system logs might miss insider threats with subtle behavioural changes. Hybrid approaches can integrate user context (department, access level) and psychological data to understand suspicious activities better.

Leveraging Content Analysis with Network Traffic Data: analysing content alone might miss exfiltration attempts via network channels. Hybrid approaches that combine content analysis with network traffic data can provide a more comprehensive picture of potential insider activities.

7.3.4 Federated Learning for Decentralized Training

Federated learning offers a promising approach to address the challenge of siloed and sensitive insider threat data within organisations. Traditional methods often require centralizing this data, raising privacy concerns. Federated learning allows training deep learning models directly on decentralized datasets across different departments. Each department trains a local model on its own data and only shares model updates, not the raw data. This collaborative approach leverages the collective power of the organisation's data for improved threat detection while maintaining departmental data privacy.

7.3.5 Practical Evaluation Metrics

Insider threat detection suffers because common evaluation metrics like accuracy don't account for the rarity of insider incidents. To address this, future research should explore the use of cumulative recall (CR-k) which considers a daily budget for investigating suspicious activities. CR-k prioritizes catching real threats even if they mean some false alarms, which better reflects the true cost of missing an insider attack.

Bibliography

[1] J. Yin, M. Tang, J. Cao, M. You, and H. Wang, "Cybersecurity applications in software: Data-driven software vulnerability assessment and management," in *Emerging Trends in Cybersecurity Applications*. Springer, 2022, pp. 371–389.

[2] X. Sun, M. Li, and H. Wang, "A family of enhanced (l, α)-diversity models for privacy preserving data publishing," *Future Generation Computer Systems*, vol. 27, no. 3, pp. 348–356, 2011.

[3] Y. Zhang, Y. Gong, Y. Gao, H. Wang, and J. Zhang, "Parameter-free voronoi neighborhood for evolutionary multimodal optimization," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 2, pp. 335–349, 2020.

[4] P. Manoharan, J. Yin, H. Wang, Y. Zhang, and W. Ye, "Insider threat detection using supervised machine learning algorithms," *Telecommunication Systems*, pp. 1–17, 2023.

[5] Scamwatch. Threats & extortion stats for 2023. Accessed: March 20, 2024. [Online]. Available: https://www.scamwatch.gov.au/research-and-resources/scam-statistics?scamid=32&date=2023

[6] D. C. Le, N. Zincir-Heywood, and M. I. Heywood, "Analyzing data granularity levels for insider threat detection using machine learning," *IEEE Transactions on Network and Service Management*, vol. 17, no. 1, pp. 30–44, 2020.

[7] B. Sharma, P. Pokharel, and B. Joshi, "User behavior analytics for anomaly detection using lstm autoencoder-insider threat detection," in *Proceedings of the 11th international conference on advances in information technology*, 2020, pp. 1–9.

[8] E. Kabir, "A role-involved purpose-based access control model," *Information Systems Frontiers*, vol. 14, pp. 809–822, 07 2012.

[9] X. Sun, H. Wang, J. Li, and J. Pei, "Publishing anonymous survey rating data," *Data Mining and Knowledge Discovery*, vol. 23, pp. 379–406, 11 2011.

[10] W. Hong, J. Yin, M. You, H. Wang, J. Cao, J. Li, and M. Liu, "Graph intelligence enhanced bi-channel insider threat detection," in *Network and System Security: 16th International Conference, NSS 2022, Denarau Island, Fiji, December 9–12, 2022, Proceedings*. Springer, 2022, pp. 86–102.

[11] X. Feng, X. Zhu, Q.-L. Han, W. Zhou, S. Wen, and Y. Xiang, "Detecting vulnerability on iot device firmware: A survey," *IEEE/CAA Journal of Automatica Sinica*, vol. 10, no. 1, pp. 25–41, 2022.

[12] J. Zhang, L. Pan, Q.-L. Han, C. Chen, S. Wen, and Y. Xiang, "Deep learning based attack detection for cyber-physical system cybersecurity: A survey," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 3, pp. 377–391, 2021.

[13] M. You, J. Yin, H. Wang, J. Cao, and Y. Miao, "A minority class boosted framework for adaptive access control decision-making," in *International Conference on Web Information Systems Engineering*. Springer, 2021, pp. 143–157.

[14] P. Institute and Proofpoint, "2022 cost of insider threats: Global report," PDF, 2022. [Online]. Available: https://www.proofpoint.com/au/resources/threat-reports/cost-of-insider-threats

[15] Gurucul, "2023 insider threat report," 2023, accessed: March 20, 2024. [Online]. Available: https://gurucul.com/2023-insider-threat-report

[16] M. You, J. Yin, H. Wang, J. Cao, K. Wang, Y. Miao, and E. Bertino, "A knowledge graph empowered online learning framework for access control decision-making," *World Wide Web*, vol. 26, no. 2, pp. 827–848, 2023.

[17] Deloitte. (2023) Impact of covid-19 on cybersecurity. [Online]. Available: https://www2.deloitte.com/ch/en/pages/risk/articles/impact-covid-cybersecurity.html

[18] J. Yin, M. Tang, J. Cao, H. Wang, M. You, and Y. Lin, "Adaptive online learning for vulnerability exploitation time prediction," in *Web Information Systems Engineering–WISE 2020: 21st International Conference, Amsterdam, The Netherlands, October 20–24, 2020, Proceedings, Part II 21*. Springer, 2020, pp. 252–266.

[19] J. Yin, M. Tang, J. Cao, M. You, H. Wang, and M. Alazab, "Knowledge-driven cybersecurity intelligence: Software vulnerability co-exploitation behaviour discovery," *IEEE Transactions on Industrial Informatics*, 2022.

[20] M. You, J. Yin, H. Wang, J. Cao, K. Wang, Y. Miao, and E. Bertino, "A knowledge graph empowered online learning framework for access control decision-making," *World Wide Web*, pp. 1–22, 2022.

[21] EkranSystem, "Insider threat statistics for 2024: Reports, facts, actors, and costs," PDF, 2024, accessed: March 20, 2024. [Online]. Available: https://www.ekransystem.com/en/blog/insider-threat-statistics-facts-and-figures

[22] V. Petkauskas. Tesla insider breach exposes thousands of employees. Updated on: November 27, 2023. [Online]. Available: https://cybernews.com/news/tesla-data-breach-thousands-exposed/

[23] J. Cox. Microsoft employees exposed own company's internal logins. Accessed: December 20, 2023. [Online]. Available: https://www.vice.com/en/article/m7gb43/microsoft-employees-exposed-login-credentials-azure-github

[24] H. Wang, J. Cao, and Y. Zhang, "A flexible payment scheme and its role-based access control," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 17, pp. 425– 436, 04 2005.

[25] J. Shu, X. Jia, K. YANG, and H. Wang, "Privacy-preserving task recommendation services for crowdsourcing," *IEEE Transactions on Services Computing*, vol. PP, pp. 1–1, 01 2018.

[26] Y. Zhang, Y. Shen, H. Wang, J. Yong, and X. Jiang, "On secure wireless communications for iot under eavesdropper collusion," *IEEE Transactions on Automation Science and Engineering*, vol. 13, pp. 1–13, 12 2015.

[27] H. Wang, Y. Zhang, and J. Cao, "Effective collaboration with information sharing in virtual universities," *IEEE Trans. Knowl. Data Eng.*, vol. 21, pp. 840–853, 06 2009.

[28] N. Baracaldo and J. Joshi, "An adaptive risk management and access control framework to mitigate insider threats," *Computers & Security*, vol. 39, pp. 237–254, 2013.

[29] N. Sun, J. Zhang, P. Rimba, S. Gao, L. Y. Zhang, and Y. Xiang, "Data-driven cybersecurity incident prediction: A survey," *IEEE communications surveys & tutorials*, vol. 21, no. 2, pp. 1744–1772, 2018.

[30] G. Lin, S. Wen, Q.-L. Han, J. Zhang, and Y. Xiang, "Software vulnerability detection using deep neural networks: a survey," *Proceedings of the IEEE*, vol. 108, no. 10, pp. 1825–1848, 2020.

[31] X. Chen, C. Li, D. Wang, S. Wen, J. Zhang, S. Nepal, Y. Xiang, and K. Ren, "Android hiv: A study of repackaging malware for evading machine-learning detection," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 987–1001, 2019.

[32] J. Qiu, J. Zhang, W. Luo, L. Pan, S. Nepal, and Y. Xiang, "A survey of android malware detection with deep neural models," *ACM Computing Surveys (CSUR)*, vol. 53, no. 6, pp. 1–36, 2020.

[33] D. C. Le, A. N. Zincir-Heywood, and M. I. Heywood, "Dynamic insider threat detection based on adaptable genetic programming," in *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2019, pp. 2579–2586.

[34] X. Sun, H. Wang, J. Li, and Y. Zhang, "Satisfying privacy requirements before data anonymization," *The Computer Journal*, vol. 55, no. 4, pp. 422–437, 2012.

[35] M. E. Kabir and H. Wang, "Conditional purpose based access control model for privacy protection," in *Proceedings of the Twentieth Australasian Conference on Australasian Database-Volume 92*, 2009, pp. 135–142.

[36] J.-Q. Yang, Q.-T. Yang, K.-J. Du, C.-H. Chen, H. Wang, S.-W. Jeon, J. Zhang, and Z.-H. Zhan, "Bi-directional feature fixation-based particle swarm optimization for large-scale feature selection," *IEEE Transactions on Big Data*, vol. PP, pp. 1–14, 01 2022.

[37] K. Cheng, L. Wang, Y. Shen, H. Wang, Y. Wang, X. Jiang, and H. Zhong, "Secure k-nn query on encrypted cloud data with multiple keys," *IEEE Transactions on Big Data*, vol. PP, pp. 1–1, 05 2017.

[38] J. Zhang, H. Li, X. Liu, Y. Luo, F. Chen, and H. Wang, "On efficient and robust anonymization for privacy protection on massive streaming categorical information," *IEEE Transactions on Dependable and Secure Computing*, vol. PP, pp. 1–1, 09 2015.

[39] E. Kabir, A. Mahmood, H. Wang, and A. Mustafa, "Microaggregation sorting framework for k-anonymity statistical disclosure control in cloud computing," *IEEE Transactions on Cloud Computing*, vol. PP, pp. 1–1, 08 2015.

[40] H. Wang, Y. Zhang, J. Cao, and V. Varadharajan, "Achieving secure and flexible m-services through tickets," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 33, no. 6, pp. 697–708, 2003.

[41] X. Sun, M. Li, H. Wang, and A. Plank, "An efficient hash-based algorithm for minimal k-anonymity," in *Conferences in Research and Practice in Information Technology (CRPIT)*, vol. 74, 2008, pp. 101–107.

[42] L. Sun, J. Ma, H. Wang, and Y. Zhang, "Cloud service description model: An extension of usdl for cloud services," *IEEE Transactions on Services Computing*, vol. PP, pp. 1–1, 08 2015.

[43] Y.-F. Ge, H. Wang, E. Bertino, Z.-H. Zhan, J. Cao, Y. Zhang, and J. Zhang, "Evolutionary dynamic database partitioning optimization for privacy and utility," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–17, 2023.

[44] C. Wang, B. Sun, K.-J. Du, J.-Y. Li, Z.-H. Zhan, S.-W. Jeon, H. Wang, and J. Zhang, "A novel evolutionary algorithm with column and sub-block local search for sudoku puzzles," *IEEE Transactions on Games*, vol. PP, pp. 1–11, 01 2023.

[45] X. Sun, H. Wang, and J. Li, "Injecting purpose and trust into data anonymisation," in *Proceedings of the 18th ACM conference on Information and knowledge management*, 2009, pp. 1541–1544.

[46] H. Wang, X. Yi, E. Bertino, and L. Sun, "Protecting outsourced data in cloud computing through access management," *Concurrency and computation: Practice and Experience*, vol. 28, no. 3, pp. 600–615, 2016.

[47] H. Wang, Y. Wang, T. Taleb, and X. Jiang, "Special issue on security and privacy in network computing," *World Wide Web*, vol. 23, pp. 951–957, 2020.

[48] N. Phruksahiran, "Improvement of source localization via cellular network using machine learning approach," *Telecommunication Systems*, pp. 1–9, 2023.

[49] S. Kok, A. Abdullah, N. Jhanjhi, and M. Supramaniam, "Ransomware, threat and detection techniques: A review," *Int. J. Comput. Sci. Netw. Secur*, vol. 19, no. 2, p. 136, 2019.

[50] H. Wang, Y. Zhang, J. Cao, and V. Varadharajan, "Achieving secure and flexible m-services through tickets," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 33, pp. 697 – 708, 12 2003.

[51] E. Kabir and H. Wang, "Conditional purpose based access control model for privacy protection." vol. 92, 01 2009, pp. 137–144.

[52] X. Sun, H. Wang, J. Li, and Y. Zhang, "Injecting purpose and trust into data anonymisation," *Computers & Security*, vol. 30, pp. 332–345, 07 2011.

[53] R. Nasir, M. Afzal, R. Latif, and W. Iqbal, "Behavioral based insider threat detection using deep learning," *IEEE Access*, vol. 9, pp. 143 266–143 274, 2021.

[54] J. Lu and R. K. Wong, "Insider threat detection with long short-term memory," in *Proceedings of the Australasian Computer Science Week Multiconference*, 2019, pp. 1–10.

[55] A. Gamachchi, L. Sun, and S. Boztas, "A graph based framework for malicious insider threat detection," *CoRR,arXiv preprint arXiv:1809.00141*, 2018.

[56] T. Rashid, I. Agrafiotis, and J. R. Nurse, "A new take on detecting insider threats: exploring the use of hidden markov models," in *Proceedings of the 8th ACM CCS International workshop on managing insider security threats*, 2016, pp. 47–56.

[57] L. Liu, O. De Vel, Q.-L. Han, J. Zhang, and Y. Xiang, "Detecting and preventing cyber insider threats: A survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 2, pp. 1397–1417, 2018.

[58] F. Yuan, Y. Cao, Y. Shang, Y. Liu, J. Tan, and B. Fang, "Insider threat detection with deep neural network," in *Computational Science–ICCS 2018: 18th International Conference, Wuxi, China, June 11–13, 2018, Proceedings, Part I 18*. Springer, 2018, pp. 43–54.

[59] P. Chattopadhyay, L. Wang, and Y.-P. Tan, "Scenario-based insider threat detection from cyber activities," *IEEE Transactions on Computational Social Systems*, vol. 5, no. 3, pp. 660–675, 2018.

[60] M. N. Al-Mhiqani, R. Ahmed, Z. Z. Abidin, and S. Isnin, "An integrated imbalanced learning and deep neural network model for insider threat detection," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 1, 2021.

[61] P. Manoharan, W. Hong, J. Yin, Y. Zhang, W. Ye, and J. Ma, "Bilateral insider threat detection: Harnessing standalone and sequential activities with recurrent neural networks," in *International Conference on Web Information Systems Engineering*. Springer, 2023, pp. 179–188.

[62] H. Wang, Y. Zhang, and J. Cao, "Ubiquitous computing environments and its usage access control," vol. 152, 01 2006, p. 6.

[63] J. Yin, M. Tang, J. Cao, and H. Wang, "Apply transfer learning to cybersecurity: Predicting exploitability of vulnerabilities by description," *Knowledge-Based Systems*, vol. 210, 10 2020.

[64] D. L. Costa, M. J. Albrethsen, and M. L. Collins, "Insider threat indicator ontology," Carnegie-Mellon Univ Pittsburgh Pa Pittsburgh United States, Tech. Rep., 2016.

[65] R. C. Brackney and R. H. Anderson, "Understanding the insider threat. proceedings of a march 2004 workshop," RAND CORP SANTA MONICA CA, Tech. Rep., 2004.

[66] A. Kim, J. Oh, J. Ryu, and K. Lee, "A review of insider threat detection approaches with iot perspective," *IEEE Access*, vol. 8, pp. 78 847–78 867, 2020.

[67] I. Lütkebohle, "Cybersecurity & infrastructure security agency (cisa)," https://www.cisa.gov/defining-insider-threats, [Online; accessed 20-June-2024].

[68] J. Predd, S. L. Pfleeger, J. Hunker, and C. Bulford, "Insiders behaving badly," *IEEE Security & Privacy*, vol. 6, no. 4, pp. 66–70, 2008.

[69] E. Schultz and R. Shumway, *Incident response: A strategic guide to handling system and network security breaches.* Sams, 2001.

[70] S. L. Pfleeger, J. B. Predd, J. Hunker, and C. Bulford, "Insiders behaving badly: Addressing bad actors and their actions," *IEEE transactions on information forensics and security*, vol. 5, no. 1, pp. 169–179, 2009.

[71] F. L. Greitzer and D. A. Frincke, "Combining traditional cyber security audit data with psychosocial data: towards predictive modeling for insider threat mitigation," in *Insider threats in cyber security.* Springer, 2010, pp. 85–113.

[72] J. Hunker and C. W. Probst, "Insiders and insider threats-an overview of definitions and mitigation techniques." *J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl.*, vol. 2, no. 1, pp. 4–27, 2011.

[73] M. Bishop, "Position: " insider" is relative," in *Proceedings of the 2005 workshop on New security paradigms*, 2005, pp. 77–78.

[74] M. Theoharidou, S. Kokolakis, M. Karyda, and E. Kiountouzis, "The insider threat to information systems and the effectiveness of iso17799," *Computers & Security*, vol. 24, no. 6, pp. 472–484, 2005.

[75] C. I. S. A. (CISA). Defining insider threats. [Online; accessed 22-04-2024]. [Online]. Available: https://www.cisa.gov/topics/physical-security/insider-threat-mitigation/defining-insider-threats

[76] M. James, "Mitigating the risk of insider threats," *CSO Online*, Mar. 2022. [Online]. Available: https://www.csoonline.com/network-security/

[77] I. T. Center, "The insider threat landscape report," Sep. 2023. [Online]. Available: https://www.sans.org/blog/decoding-insider-threat/

[78] E. Cole and S. Ring, *Insider threat: Protecting the enterprise from sabotage, spying, and theft.* Elsevier, 2005.

[79] How much data is created every day in 2021? [Online]. Available: https://techjury.net/blog/how-much-data-is-created-every-day/#gref.

[80] I. Proofpoint. Global cybersecurity study: Insider threats cost organizations $15.4 million annually, up 34 percent from 2020. [Online]. Available: https://www.proofpoint.com/us/resources/threat-reports/cost-of-insider-threats

[81] H. G. Goldberg, W. T. Young, A. Memory, and T. E. Senator, "Explaining and aggregating anomalies to detect insider threats," in *2016 49th Hawaii International Conference on System Sciences (HICSS)*. IEEE, 2016, pp. 2739–2748.

[82] W. Jiang, Y. Tian, W. Liu, and W. Liu, "An insider threat detection method based on user behavior analysis," in *Intelligent Information Processing IX: 10th IFIP TC 12 International Conference, IIP 2018, Nanning, China, October 19-22, 2018, Proceedings 10*. Springer, 2018, pp. 421–429.

[83] J. Jiang, J. Chen, K.-K. R. Choo, K. Liu, C. Liu, M. Yu, and P. 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)*. IEEE, 2018, pp. 1–6.

[84] C. Liu, Y. Zhong, and Y. Wang, "Improved detection of user malicious behavior through log mining based on ihmm," in *2018 5th International Conference on Systems and Informatics (ICSAI)*. IEEE, 2018, pp. 1193–1198.

[85] X. Wang, Q. Tan, J. Shi, S. Su, and M. Wang, "Insider threat detection using characterizing user behavior," in *2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*. IEEE, 2018, pp. 476–482.

[86] Y. Li, S. Wang, S. Xu, and J. Yin, "Trustworthy semi-supervised anomaly detection for online-to-offline logistics business in merchant identification," *CAAI Transactions on Intelligence Technology*, 2024.

[87] R. Nasir, M. Afzal, R. Latif, and W. Iqbal, "Behavioral based insider threat detection using deep learning," *IEEE Access*, vol. 9, pp. 143 266–143 274, 2021.

[88] S. Song, N. Gao, Y. Zhang, and C. Ma, "Britd: behavior rhythm insider threat detection with time awareness and user adaptation," *Cybersecurity*, vol. 7, no. 1, p. 2, 2024.

[89] F. Liu, Y. Wen, D. Zhang, X. Jiang, X. Xing, and D. Meng, "Log2vec: A heterogeneous graph embedding based approach for detecting cyber threats within enterprise," in *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*, 2019, pp. 1777–1794.

[90] S. Paul and S. Mishra, "Lac: Lstm autoencoder with community for insider threat detection," in *2020 the 4th International Conference on Big Data Research (ICBDR'20)*, 2020, pp. 71–77.

[91] X. Wang, J. Jiang, Y. Wang, Q. Lv, and L. Wang, "Uag: User action graph based on system logs for insider threat detection," in *2023 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2023, pp. 1027–1032.

[92] J. Xiao, L. Yang, F. Zhong, X. Wang, H. Chen, and D. Li, "Robust anomaly-based insider threat detection using graph neural network," *IEEE Transactions on Network and Service Management*, 2022.

[93] R. Gayathri, A. Sajjanhar, and Y. Xiang, "Adversarial training for robust insider threat detection," in *2022 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2022, pp. 1–8.

[94] D. C. Le and A. N. Zincir-Heywood, "Evaluating insider threat detection workflow using supervised and unsupervised learning," in *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2018, pp. 270–275.

[95] T. Al-Shehari, M. Al-Razgan, T. Alfakih, R. A. Alsowail, and S. Pandiaraj, "Insider threat detection model using anomaly-based isolation forest algorithm," *IEEE Access*, 2023.

[96] J. Jiang, J. Chen, T. Gu, K.-K. R. Choo, C. Liu, M. Yu, W. Huang, and P. Mohapatra, "Anomaly detection with graph convolutional networks for insider threat and fraud detection," in *MILCOM 2019-2019 IEEE Military Communications Conference (MILCOM)*. IEEE, 2019, pp. 109–114.

[97] G. Gavai, K. Sricharan, D. Gunning, R. Rolleston, J. Hanley, and M. 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*, 2015, pp. 13–20.

[98] D. C. Le and A. N. Zincir-Heywood, "Machine learning based insider threat modelling and detection," in *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*. IEEE, 2019, pp. 1–6.

[99] B. Böse, B. Avasarala, S. Tirthapura, Y.-Y. Chung, and D. Steiner, "Detecting insider threats using radish: A system for real-time anomaly detection in heterogeneous data streams," *IEEE Systems Journal*, vol. 11, no. 2, pp. 471–482, 2017.

[100] D. C. Le, N. Zincir-Heywood, and M. Heywood, "Training regime influences to semi-supervised learning for insider threat detection," in *2021 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2021, pp. 13–18.

[101] B. Bin Sarhan and N. Altwaijry, "Insider threat detection using machine learning approach," *Applied Sciences*, vol. 13, no. 1, p. 259, 2022.

[102] F. Janjua, A. Masood, H. Abbas, and I. Rashid, "Handling insider threat through supervised machine learning techniques," *Procedia Computer Science*, vol. 177, pp. 64–71, 2020.

[103] A. Tuor, S. Kaplan, B. Hutchinson, N. Nichols, and S. Robinson, "Deep learning for unsupervised insider threat detection in structured cybersecurity data streams," *arXiv preprint arXiv:1710.00811*, 2017.

[104] S. Yuan, P. Zheng, X. Wu, and Q. Li, "Insider threat detection via hierarchical neural temporal point processes," in *2019 IEEE international conference on big data (big data)*. IEEE, 2019, pp. 1343–1350.

[105] D. C. Le, N. Zincir-Heywood, and M. I. Heywood, "Analyzing data granularity levels for insider threat detection using machine learning," *IEEE Transactions on Network and Service Management*, vol. 17, no. 1, pp. 30–44, 2020.

[106] Z. Tian, W. Shi, Z. Tan, J. Qiu, Y. Sun, F. Jiang, and Y. Liu, "Deep learning and dempster-shafer theory based insider threat detection," *Mobile Networks and Applications*, pp. 1–10, 2020.

[107] A. Anju, K. Shalini, H. Ravikumar, P. Saranya, M. Krishnamurthy *et al.*, "Detection of insider threats using deep learning," in *2023 3rd International Conference on Pervasive Computing and Social Networking (ICPCSN)*. IEEE, 2023, pp. 264–269.

[108] A. Saaudi, Z. Al-Ibadi, Y. Tong, and C. Farkas, "Insider threats detection using cnn-lstm model," in *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE, 2018, pp. 94–99.

[109] H. Xiao, Y. Zhu, B. Zhang, Z. Lu, D. Du, and Y. Liu, "Unveiling shadows: A comprehensive framework for insider threat detection based on statistical and sequential analysis," *Computers & Security*, vol. 138, p. 103665, 2024.

[110] Q. Ma and N. Rastogi, "Dante: Predicting insider threat using lstm on system logs," in *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. IEEE, 2020, pp. 1151–1156.

[111] B. Sabir, F. Ullah, M. A. Babar, and R. Gaire, "Machine learning for detecting data exfiltration: a review," *ACM Computing Surveys (CSUR)*, vol. 54, no. 3, pp. 1–47, 2021.

[112] L. Liu, O. De Vel, Q.-L. Han, J. Zhang, and Y. Xiang, "Detecting and preventing cyber insider threats: A survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 2, pp. 1397–1417, 2018.

[113] R. A. Alsowail and T. Al-Shehari, "Techniques and countermeasures for preventing insider threats," *PeerJ Computer Science*, vol. 8, p. e938, 2022.

[114] I. Homoliak, F. Toffalini, J. Guarnizo, Y. Elovici, and M. Ochoa, "Insight into insiders and it: A survey of insider threat taxonomies, analysis, modeling, and countermeasures," *ACM Computing Surveys (CSUR)*, vol. 52, no. 2, pp. 1–40, 2019.

[115] F. Yuan, Y. Shang, Y. Liu, Y. Cao, and J. Tan, "Attention-based lstm for insider threat detection," in *Applications and Techniques in Information Security: 10th International Conference, ATIS 2019, Thanjavur, India, November 22–24, 2019, Proceedings 10.* Springer, 2019, pp. 192–201.

[116] E. Yilmaz and O. Can, "Unveiling shadows: Harnessing artificial intelligence for insider threat detection," *Engineering, Technology & Applied Science Research*, vol. 14, no. 2, pp. 13 341–13 346, 2024.

[117] O. Brdiczka, J. Liu, B. Price, J. Shen, A. Patil, R. Chow, E. Bart, and N. Ducheneaut, "Proactive insider threat detection through graph learning and psychological context," in *2012 IEEE Symposium on Security and Privacy Workshops.* IEEE, 2012, pp. 142–149.

[118] R. Sandhu, D. Ferraiolo, R. Kuhn *et al.*, "The nist model for role-based access control: towards a unified standard," in *ACM workshop on Role-based access control*, vol. 10, no. 344287.344301, 2000.

[119] N. Metoui, M. Bezzi, and A. Armando, "Risk-based privacy-aware access control for threat detection systems," *Transactions on Large-Scale Data- and Knowledge-Centered Systems XXXVI: Special Issue on Data and Security Engineering*, pp. 1–30, 2017.

[120] J. Yin, W. Hong, H. Wang, J. Cao, Y. Miao, and Y. Zhang, "A compact vulnerability knowledge graph for risk assessment," *ACM Trans. Knowl. Discov. Data*, jun 2024, just Accepted. [Online]. Available: https://doi.org/10.1145/3671005

[121] L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5–32, 2001.

[122] G. Biau and E. Scornet, "A random forest guided tour," *Test*, vol. 25, pp. 197–227, 2016.

[123] M. Belgiu and L. Drăguţ, "Random forest in remote sensing: A review of applications and future directions," *ISPRS journal of photogrammetry and remote sensing*, vol. 114, pp. 24–31, 2016.

[124] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[125] S. S. Dhaliwal, A.-A. Nahid, and R. Abbas, "Effective intrusion detection system using xgboost," *Information*, vol. 9, no. 7, p. 149, 2018.

[126] Y. Wang, Y. Shen, H. Wang, J. Cao, and X. Jiang, "Mtmr: Ensuring mapreduce computation integrity with merkle tree-based verifications," *IEEE Transactions on Big Data*, vol. 4, no. 3, pp. 418–431, 2016.

[127] F. Zhang, Y. Wang, S. Liu, and H. Wang, "Decision-based evasion attacks on tree ensemble classifiers," *World Wide Web*, vol. 23, pp. 2957–2977, 2020.

[128] B. Charbuty and A. Abdulazeez, "Classification based on decision tree algorithm for machine learning," *Journal of Applied Science and Technology Trends*, vol. 2, no. 01, pp. 20–28, 2021.

[129] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

[130] H. Wang and L. Sun, "Trust-involved access control in collaborative open social networks," in *2010 fourth international conference on network and system security*. IEEE, 2010, pp. 239–246.

[131] T. Huang, Y.-J. Gong, S. Kwong, H. Wang, and J. Zhang, "A niching memetic algorithm for multi-solution traveling salesman problem," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 3, pp. 508–522, 2019.

[132] M. C. Belavagi and B. Muniyal, "Performance evaluation of supervised machine learning algorithms for intrusion detection," *Procedia Computer Science*, vol. 89, pp. 117–123, 2016.

[133] L. E. Peterson, "K-nearest neighbor," *Scholarpedia*, vol. 4, no. 2, p. 1883, 2009.

[134] O. Kramer and O. Kramer, "K-nearest neighbors," *Dimensionality reduction with unsupervised nearest neighbors*, pp. 13–23, 2013.

[135] A. Tharwat, "Linear vs. quadratic discriminant analysis classifier: a tutorial," *International Journal of Applied Pattern Recognition*, vol. 3, no. 2, pp. 145–180, 2016.

[136] S. Srivastava, M. R. Gupta, and B. A. Frigyik, "Bayesian quadratic discriminant analysis." *Journal of Machine Learning Research*, vol. 8, no. 6, 2007.

[137] R. E. Schapire, "Explaining adaboost," in *Empirical Inference: Festschrift in Honor of Vladimir N. Vapnik*. Springer, 2013, pp. 37–52.

[138] T. Hastie, S. Rosset, J. Zhu, and H. Zou, "Multi-class adaboost," *Statistics and its Interface*, vol. 2, no. 3, pp. 349–360, 2009.

[139] J. Yin, G. Chen, W. Hong, H. Wang, J. Cao, and Y. Miao, "Empowering vulnerability prioritization: A heterogeneous graph-driven framework for exploitability prediction," in *International Conference on Web Information Systems Engineering*. Springer, 2023, pp. 289–299.

[140] M.-C. Popescu, V. E. Balas, L. Perescu-Popescu, and N. Mastorakis, "Multilayer perceptron and neural networks," *WSEAS Transactions on Circuits and Systems*, vol. 8, no. 7, pp. 579–588, 2009.

[141] J. Yin, M. You, J. Cao, H. Wang, M. Tang, and Y.-F. Ge, "Data-driven hierarchical neural network modeling for high-pressure feedwater heater group," in *Australasian Database Conference*. Springer, 2020, pp. 225–233.

[142] Y. Chen, S. Han, G. Chen, J. Yin, K. N. Wang, and J. Cao, "A deep reinforcement learning-based wireless body area network offloading optimization strategy for healthcare services," *Health Information Science and Systems*, vol. 11, no. 1, p. 8, 2023.

[143] J. C. Stoltzfus, "Logistic regression: a brief primer," *Academic emergency medicine*, vol. 18, no. 10, pp. 1099–1104, 2011.

[144] C.-Y. J. Peng, K. L. Lee, and G. M. Ingersoll, "An introduction to logistic regression analysis and reporting," *The journal of educational research*, vol. 96, no. 1, pp. 3–14, 2002.

[145] J. Yin, G. Chen, W. Hong, J. Cao, H. Wang, and Y. Miao, "A heterogeneous graph-based semi-supervised learning framework for access control decision-making," *World Wide Web*, vol. 27, no. 4, p. 35, 2024.

[146] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International conference on machine learning*. Pmlr, 2013, pp. 1310–1318.

[147] L. Medsker and L. C. Jain, *Recurrent neural networks: design and applications*. CRC press, 1999.

[148] X. Zhang, G. Zhang, X. Qiu, J. Yin, W. Tan, X. Yin, H. Yang, L. Liao, H. Wang, and Y. Zhang, "Radiomics under 2d regions, 3d regions, and peritumoral regions reveal tumor heterogeneity in non-small cell lung cancer: a multicenter study," *La radiologia medica*, vol. 128, no. 9, pp. 1079–1092, 2023.

[149] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: Lstm cells and network architectures," *Neural computation*, vol. 31, no. 7, pp. 1235–1270, 2019.

[150] S. Hesaraki. (2023) Long short-term memory (lstm). [04-Jan-2024]. [Online]. Available: https://medium.com/@saba99/long-short-term-memory-lstm-fffc5eaebfdc

[151] Z. Huang, W. Xu, and K. Yu, "Bidirectional lstm-crf models for sequence tagging," *arXiv preprint arXiv:1508.01991*, 2015.

[152] M. B. Salem and S. J. Stolfo, "Modeling user search behavior for masquerade detection," in *International Workshop on Recent Advances in Intrusion Detection*. Springer, 2011, pp. 181–200.

[153] M. B. Salem and S. J. Stolfo, "Modeling user search behavior for masquerade detection," in *International Workshop on Recent Advances in Intrusion Detection*. Springer, 2011, pp. 181–200.

[154] M. Schonlau, W. DuMouchel, W.-H. Ju, A. F. Karr, M. Theus, and Y. Vardi, "Computer intrusion: Detecting masquerades," *Statistical science*, pp. 58–74, 2001.

[155] S. Greenberg, "Using unix: Collected traces of 168 users," 1988. [Online]. Available: https://prism.ucalgary.ca/handle/1880/45929

[156] A. Harilal, F. Toffalini, J. Castellanos, J. Guarnizo, I. Homoliak, and M. 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*, 2017, pp. 45–56.

[157] T. E. Senator, H. G. Goldberg, A. Memory, W. T. Young, B. Rees, R. Pierce, D. Huang, M. Reardon, D. A. Bader, E. Chow *et al.*, "Detecting insider threats in a real corporate database of computer usage activity," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2013, pp. 1393–1401.

[158] S. Mathew, M. Petropoulos, H. Q. Ngo, and S. Upadhyaya, "A data-centric approach to insider attack detection in database systems," in *Recent Advances in Intrusion Detection: 13th International Symposium, RAID 2010, Ottawa, Ontario, Canada, September 15-17, 2010. Proceedings 13*. Springer, 2010, pp. 382–401.

[159] W. Wang, W. Wang, and J. Yin, "A bilateral filtering based ringing elimination approach for motion-blurred restoration image," *Current Optics and Photonics*, vol. 4, no. 3, pp. 200–209, 2020.

[160] R. S. Rao, A. Umarekar, and A. R. Pais, "Application of word embedding and machine learning in detecting phishing websites," *Telecommunication Systems*, pp. 1–13, 2022.

[161] X. Hu, W. Ma, C. Chen, S. Wen, J. Zhang, Y. Xiang, and G. Fei, "Event detection in online social network: Methodologies, state-of-art, and evolution," *Computer Science Review*, vol. 46, p. 100500, 2022.

[162] X. Zhu, S. Wen, S. Camtepe, and Y. Xiang, "Fuzzing: a survey for roadmap," *ACM Computing Surveys (CSUR)*, vol. 54, no. 11s, pp. 1–36, 2022.

[163] H. Kavusi, K. Maghooli, and S. Haghipour, "A novel and smarter model to authenticate and identify people intelligently for security purposes," *Telecommunication Systems*, vol. 82, no. 1, pp. 27–43, 2023.

[164] M. Humayun, N. Jhanjhi, M. F. Almufareh, and M. I. Khalil, "Security threat and vulnerability assessment and measurement in secure software development," *Comput. Mater. Contin*, vol. 71, pp. 5039–5059, 2022.

[165] D. 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)*, 2019, pp. 1–6.

[166] M. A. Haq, M. A. R. Khan, and M. Alshehri, "Insider threat detection based on nlp word embedding and machine learning," *Intell. Autom. Soft Comput*, vol. 33, no. 1, pp. 619–635, 2022.

[167] D. C. Le and N. Zincir-Heywood, "Exploring anomalous behaviour detection and classification for insider threat identification," *International Journal of Network Management*, vol. 31, no. 4, p. e2109, 2021.

[168] J. Yi and Y. Tian, "Insider threat detection model enhancement using hybrid algorithms between unsupervised and supervised learning," *Electronics*, vol. 13, no. 5, p. 973, 2024.

[169] F. Yuan, Y. Cao, Y. Shang, Y. Liu, J. Tan, and B. Fang, "Insider threat detection with deep neural network," in *Computational Science–ICCS 2018: 18th International Conference, Wuxi, China, June 11–13, 2018, Proceedings, Part I 18*. Springer, 2018, pp. 43–54.

[170] L. Lin, S. Zhong, C. Jia, and K. Chen, "Insider threat detection based on deep belief network feature representation," in *2017 International Conference on Green Informatics (ICGI)*. IEEE, 2017, pp. 54–59.

[171] M. Singh, B. Mehtre, and S. Sangeetha, "Insider threat detection based on user behaviour analysis," in *Machine Learning, Image Processing, Network Security and Data Sciences: Second International Conference, MIND 2020, Silchar, India, July 30-31, 2020, Proceedings, Part II 2*. Springer, 2020, pp. 559–574.

[172] M. N. Al-Mhiqani, R. Ahmad, Z. Zainal Abidin, W. Yassin, A. Hassan, K. H. Abdulkareem, N. S. Ali, and Z. Yunos, "A review of insider threat detection: classification, machine learning techniques, datasets, open challenges, and recommendations," *Applied Sciences*, vol. 10, no. 15, p. 5208, 2020.

[173] S. Yuan and X. Wu, "Deep learning for insider threat detection: Review, challenges and opportunities," *Computers & Security*, vol. 104, p. 102221, 2021.

[174] J. Yin, M. Tang, J. Cao, M. You, and H. Wang, "Cybersecurity applications in software: Data-driven software vulnerability assessment and management," in *Emerging Trends in Cybersecurity Applications*. New York City: Springer, 2022, pp. 371–389.

[175] H. Wang, J. Cao, and Y. Zhang, "Building access control policy model for privacy preserving and testing policy conflicting problems," *Access Control Management in Cloud Environments*, pp. 225–247, 2020.

[176] M. N. Al-Mhiqani, R. Ahmad, Z. Z. Abidin, K. H. Abdulkareem, M. A. Mohammed, D. Gupta, and K. Shankar, "A new intelligent multilayer framework for insider threat detection," *Computers & Electrical Engineering*, vol. 97, p. 107597, 2022.

[177] J. Yin, M. Tang, J. Cao, M. You, H. Wang, and M. Alazab, "Knowledge-driven cybersecurity intelligence: Software vulnerability coexploitation behavior discovery," *IEEE Transactions on Industrial Informatics*, vol. PP, pp. 1–9, 01 2022.

[178] J.-Y. Li, K.-J. Du, Z.-H. Zhan, H. Wang, and J. Zhang, "Distributed differential evolution with adaptive resource allocation," *IEEE transactions on cybernetics*, vol. PP, 03 2022.

[179] F. Liu, X. Zhou, J. Cao, Z. Wang, W. Tianben, H. Wang, and Y. Zhang, "Anomaly detection in quasi-periodic time series based on automatic data segmentation and attentional lstm-cnn," *IEEE Transactions on Knowledge and Data Engineering*, vol. PP, pp. 1–1, 08 2020.

[180] M. Aldairi, L. Karimi, and J. Joshi, "A trust aware unsupervised learning approach for insider threat detection," in *2019 IEEE 20th International Conference on Information Reuse and Integration for Data Science (IRI)*. IEEE, 2019, pp. 89–98.

[181] A. Sherstinsky, "Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network," *Physica D: Nonlinear Phenomena*, vol. 404, p. 132306, 2020.

[182] F. Meng, F. Lou, Y. Fu, and Z. Tian, "Deep learning based attribute classification insider threat detection for data security," in *2018 IEEE third international conference on data science in cyberspace (DSC)*. IEEE, 2018, pp. 576–581.

[183] P. Pal, P. Chattopadhyay, and M. Swarnkar, "Temporal feature aggregation with attention for insider threat detection from activity logs," *Expert Systems with Applications*, vol. 224, p. 119925, 2023.

[184] J. Glasser and B. Lindauer, "Bridging the gap: A pragmatic approach to generating insider threat data," in *2013 IEEE Security and Privacy Workshops*. IEEE, 2013, pp. 98–104.

[185] J. Zhang, X. Tao, and H. Wang, "Outlier detection from large distributed databases," *World Wide Web*, vol. 17, 07 2014.

[186] E. Kabir, A. Mahmood, H. Wang, and A. Mustafa, "Microaggregation sorting framework for k-anonymity statistical disclosure control in cloud computing," *IEEE Transactions on Cloud Computing*, vol. PP, pp. 408–417, 08 2015.

[187] R. C. Staudemeyer and E. R. Morris, "Understanding lstm–a tutorial into long short-term memory recurrent neural networks," *arXiv preprint arXiv:1909.09586*, 2019.

[188] A. Gamachchi and S. Boztas, "Insider threat detection through attributed graph clustering," in *2017 IEEE Trustcom/BigDataSE/ICESS*. IEEE, 2017, pp. 112–119.

[189] M. Kandias, A. Mylonas, N. Virvilis, M. Theoharidou, and D. Gritzalis, "An insider threat prediction model," in *Trust, Privacy and Security in Digital Business: 7th International Conference, TrustBus 2010, Bilbao, Spain, August 30-31, 2010. Proceedings 7*. Springer, 2010, pp. 26–37.

[190] F. Xiao, W. Hong, J. Yin, H. Wang, J. Cao, and Y. Zhang, "A study on historical behaviour enabled insider threat prediction," in *The Asia Pacific Web (APWeb) and Web-Age Information Management (WAIM) Join International Conference on Web and Big Data (APWeb-WAIM)*. Springer, 2024, pp. 1–13.

[191] Y. Wei, K.-P. Chow, and S.-M. Yiu, "Insider threat prediction based on unsupervised anomaly detection scheme for proactive forensic investigation," *Forensic Science International: Digital Investigation*, vol. 38, p. 301126, 2021.