AI-ENHANCED MODEL FOR DETECTING CYBERSECURITY THREATS ON SOCIAL MEDIA PLATFORMS

A Thesis submitted by

Omar Alsodi

BMIS, MPM

For the award of

Doctor of Philosophy

2024

ABSTRACT

The proliferation of cyber threats on social media requires the development of robust detection systems. X (formerly Twitter) is a prime example due to its extensive user base and real-time nature. Despite advancements in key enabling technologies such as Artificial Intelligence (AI), Machine Learning (ML), big data, blockchain, cloud computing, and the Internet of Things (IoT), the frequency and sophistication of cyberattacks escalate. Thus, cyber security remains a critical priority for X, as existing detection systems struggle to identify increasingly complex threats. This study addresses cybersecurity threats, the motivations behind them, and the primary challenges in detecting them on X. Current studies lack comprehensive evaluations of critical factors such as prediction scope, types of cyber security threats, feature extraction techniques, algorithm complexity, information summarisation levels, scalability over time, and performance measures. The surge in social media activities, particularly tweets, exacerbates the problem, making it imperative to develop more effective detection techniques. Traditional methods, including anomaly detection and rule-based approaches, are time-consuming, resource-intensive, and prone to inaccuracies. By contrast, AI, especially ML and DL, enhances the precision of cyber threat assessments. The research introduces a Subspace Random Ensemble Machine Learning Model (SREMLM) and a Voting Ensemble Deep Learning Model (VEDLM) designed for cyber threat detection and addressing cybersecurity challenges on social media. The findings demonstrate that the proposed SREMLM, and the proposed VEDLM, surpass individual DL and ML models in identifying cyber threats. Comparative analysis further confirms that this voting ensemble model, which incorporates multiple deep learning techniques, consistently delivers superior performance. The overall performance of the Voting Ensemble Deep Learning Model (VEDLM) exceeds the Subspace Random Ensemble Machine Learning Model (SREMLM), showcasing the advantage of using deep learning approaches. Furthermore, this research proposes a conceptual framework grounded in real-world datasets to enhance the practical applicability of the findings. The development of the novel proposed Ensemble ML, DL model advances threat detection capabilities, offering a pathway to bolstered security on the X platform. The study underscores the potential of ML, DL methods to revolutionise cyber threat detection, ensuring more robust and effective defense mechanisms against evolving cyber threats.

CERTIFICATION OF THESIS

I, Omar Alsodi, hereby declare that the thesis entitled *AI-Enhanced Model for Detecting Cybersecurity Threats on Social Media Platforms* does not exceed 100,000 words, including citations, excluding tables, figures, appendices, references, bibliography, and footnotes. This thesis does not contain any content, in whole or in part, previously submitted for the award of another degree or diploma. Unless otherwise noted, this thesis is my own work.

Date: September 24 , 2024

Principal Supervisor:
Professor Xujuan Zhou

Associate Supervisor:
Professor Raj Gururajan

Associate Supervisor:
Dr Anup Shrestha

Student and supervisors' signatures of endorsement are held at the University.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to those who have contributed immensely to the successful completion of this research.

First and foremost, I am deeply indebted to my supervisor, Professor Xujuan Zhou, for her unwavering support, guidance, and invaluable expertise throughout this journey. Her patience and encouragement have been instrumental in shaping this thesis.

I am equally grateful to my thesis associate supervisors, Professor Raj Gururajan, Doctor Anup Shrestha for their insightful feedback and dedicated support. Their constructive criticism has significantly enhanced the quality of this work.

I am truly thankful for the encouragement and camaraderie of my colleagues and friends. Their belief in me has been a constant source of motivation.

I would also like to thank the faculty and staff at UniSQ for creating an academic environment that fosters intellectual exploration and excellence.

To my beloved wife and family, I express my sincere gratitude for your boundless love, unwavering support, and selfless sacrifices. Your steadfast belief in me has been the cornerstone of my academic endeavours.

Finally, I acknowledge the support provided by the Research Training Program (RTP) by The Australian Commonwealth Government, whose support has made this research possible.

The completion of this thesis would not have been achievable without the collective support and contributions of everyone mentioned above. Thank you for believing in me.

KEYWORDS

Artificial Intelligence (AI); Twitter, X, Cyber Threats, Cyber Threat Intelligence, social media; Deep Learning, Machine Learning, Ensemble Learning, Cyber Security, Twitter API, Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM), Bidirectional Long Short-Term Memory (BiLSTM), Iterated Dilated Convolutional Neural Network (IDCNN). K-Nearest Neighbours (KNN), Logistic Regression (LR), Decision Tree (DT), Random Forest (RF).

TABLE OF CONTENTS

ABSTRACT i
CERTIFICATION OF THESIS ii
ACKNOWLEDGMENTS iii
KEYWORDS iv
TABLE OF CONTENTS v
LIST OF FIGURES viii
LIST OF TABLES x
LIST OF ABBREVIATIONS xi
CHAPTER 1 : INTRODUCTION 1
1.1 Background 1
1.2 Research problem 6
1.3 Aim of this PhD thesis 8
1.4 Significant contributions 10
1.5 Thesis organisation 11
CHAPTER 2: LITERATURE REVIEW 14
2.1 Cybersecurity challenges and threats to X 14
2.1.1 *Multimedia content threats* 16
2.2.1 *Traditional threats* 21
2.3.1 *Social threats* 23
2.2 Motivations for the cyber threats on X 24
2.3 Cyber threat detection on X: ML, DL-based solutions 27
2.3.1 Machine Learning (ML) 28
2.3.2 Deep Learning (DL) 35
2.3.3 X security: ML/DL solutions 38
2.3.4 Ensemble learning 50
2.4 Chapter summary 55
CHAPTER 3 : RESEARCH METHODOLOGY 56
3.1 Scientific approaches 56
3.2 Action research approach 56
3.3 Research design and approach 57
3.3.1 Literature review 58
3.3.2 Conceptual framework 58
3.3.3 Theoretical framework 59
3.3.4 Data selection 60
3.3.5 Experimental configuration and results 60

3.3.6 Evaluation and reflection ... 61
3.3.7 Interpretation and write-up ... 63
3.4 Chapter summary ... 63
CHAPTER 4 : THE PROPOSED SUBSPACE RANDOM ENSEMBLE MACHINE LEARNING MODEL (SREMLM) ... 64
4.1 The Collected Dataset ... 65
4.1.1 Data collection method ... 65
4.1.2 The collected dataset overview ... 67
4.1.3 Data cleaning ... 68
4.1.4 Feature scaling ... 69
4.1.5 Data features ... 70
4.1.6 Generating word clouds ... 71
4.1.7 Dataset balancing ... 72
4.1.8 Data preprocessing ... 72
4.2 Machine Learning techniques ... 76
4.2.1 K-Nearest Neighbours (KNN) ... 76
4.2.2 Logistic Regression (LR) ... 78
4.2.3 Decision Tree (DT) ... 79
4.2.4 Random Forest (RF) ... 80
4.3 The Proposed Subspace Random Ensemble Machine Learning Model (SREMLM) ... 82
4.4 Experimental results and discussion ... 85
4.4.1 Comparison between ML techniques before the ensemble approach ... 86
4.4.2 Comparison between ML techniques after ensemble approach ... 91
4.5 Chapter summary ... 94
CHAPTER 5 : THE PROPOSED VOTING ENSEMBLE DEEP LEARNING MODEL (VEDLM) ... 96
5.1 Deep Learning techniques ... 98
5.1.1 Long Short-Term Memory (LSTM) model ... 98
5.1.2 The Bidirectional Long Short-Term Memory (BiLSTM) model ... 102
5.1.3 IDCNN-BiLSTM MODEL ... 104
5.1.4 Convolutional Neural Network (CNN) Model ... 108
5.2 The proposed Voting Ensemble Deep Learning model (VEDLM) ... 111
5.3 Experimental results and discussion ... 117
5.3.1 Comparison between DL techniques before ensemble approach ... 117
5.3.2 Comparison between DL techniques after voting ensemble approach ... 122
5.4 Chapter summary ... 127

CHAPTER 6 : COMPARATIVE ANALYSIS 128
6.1 Comparative analysis: SREMLM and VEDLM on the same dataset 128
6.2 Comparing VEDLM to related works 130
6.2.1 Dataset 131
A. Dataset preprocessing 131
B. Dataset word clouds 132
C. Dataset balancing 132
D. The Train-Test Split datasets 133
E. Cross Validation (CV) 133
6.3 Finding 136
6.4 Discussion 142
6.5 Chapter summary 147
CHAPTER 7 : CONCLUSION AND FUTURE WORK 149
7.1 Concluding remarks 149
7.2 Current limitations 152
7.3 Future directions 153
REFERENCES 155
APPENDICES 168

LIST OF FIGURES

Figure 1: AI, Machine Learning, and Deep Learning: Understanding the Relationship, (Manakitsa et al. 2024). 2
Figure 2: Global cybersecurity spending from 2017 to 2024, (Research & Department 2023) 4
Figure 3: Categories of cyber threats on social media platforms (Rathore et al. 2017). 16
Figure 4: Example of the Cyber threats impact and motivation on social media, (Akoto 2024). 27
Figure 5: Cyber-Twitter architecture, Horrocks et al. (2004) 43
Figure 6: Running example, (Sapienza et al. 2017) 44
Figure 7: SYNAPSE's Architecture, (Alves et al. 2021) 45
Figure 8: SONAR Architecture, (Le Sceller et al. 2017) 45
Figure 9: Data mining situational awareness scheme, (Rodriguez & Okamura 2019) 46
Figure 10: DeepNN BiLSTM Architecture, (Dionísio et al. 2019) 48
Figure 11: Research design framework developed in this doctoral thesis 58
Figure 12: Proposed conceptual framework 59
Figure 13: Proposed Subspace Random Ensemble Machine Learning Model (SREMLM) 65
Figure 14: Number of each class for the collected dataset 68
Figure 15: Word Clouds for the collected dataset 71
Figure 16: Data Balancing for collected dataset 72
Figure 17: The datasets' preprocessing steps 76
Figure 18: ROC curve for K-Nearest Neighbours (KNN) 77
Figure 19: ROC Curve for Logistic Regression (LR) 79
Figure 20: ROC for Decision Tree (DT) 80
Figure 21: ROC for Random Forest (RF) 81
Figure 22: F1 score, Comparison the ensemble techniques. 85
Figure 23: ML Algorithms performance before applying ensemble techniques. 87
Figure 24: KNN Confusion Matrix 88
Figure 25: LR Confusion Matrix 88
Figure 26: RF Confusion Matrix 89
Figure 27: DT Confusion Matrix 89
Figure 28: F1 score for ML before applying ensemble techniques 91
Figure 29: Performance results after applying Subspace Random ensemble approach 93
Figure 30: F1 score for DL techniques after applying Subspace Random Ensemble 94
Figure 31: The proposed voting Ensemble model (VEDLM) 97
Figure 32: Architecture of a LSTM unit. 99
Figure 33: LSTM model structure 100
Figure 34: Architecture of the BiLSTM, (Graves & Schmidhuber 2005). 102
Figure 35: BiLSTM model structure. 103
Figure 36: IDCNN-BiLSTM model structure 106
Figure 37: General architecture of the convolutional neural network (CNN), (Graves & Schmidhuber 2005). 108
Figure 38: CNN model structure 110
Figure 39: DL Ensemble Methods and Algorithm Performance 113
Figure 40: F1 score for Ensemble Methods and Algorithm Performance 113
Figure 41. DL Ensemble models _F1 sore 114

Figure 42: DL Algorithms performance before applying proposed voting Ensemble Approach 119
Figure 43: LSTM confusion matrix 120
Figure 44: BLSTM confusion matrix 120
Figure 46: CNN confusion matrix. 121
Figure 45: BiLSTM confusion matrix 121
Figure 47: Performance results after applying voting Ensemble approach 125
Figure 48: F1 score for DL techniques after applying the proposed Voting Ensemble (VEDLM) approach 126
Figure 49: Confusion Matrix for the proposed VEDLM model 127
Figure 50: Comparison of the models in Chapter 4 and 5 - Performance. 130
Figure 51: Number of each class for the open-source dataset 131
Figure 52: Word Clouds for the open-source dataset 132
Figure 53: Data Balancing for open-source dataset 133
Figure 54: Comparison of the effectiveness of models based on the models' performance 145
Figure 55: Related works, model Performance: a comparative analysis 147

LIST OF TABLES

Table 1: Comparison of most popular attacks on online social networks, (Alsodi et al. 2021). 6
Table 2: Comparisons of previous studies on the detection of cybersecurity threats on X. ... 49
Table 3: A summary of related work ... 54
Table 4: The key characteristics of the datasets ... 60
Table 5: Performance measures ... 63
Table 6: Collected dataset keyword selection ... 66
Table 7. Data Extraction process algorithm ... 66
Table 8: Sample of the collected dataset ... 68
Table 9. Algorithm 4.2.1: K-Nearest Neighbours (KNN) ... 77
Table 10. Algorithm 4.2.2: Logistic regression (LR) ... 78
Table 11. Algorithm 4.2.3: Decision Tree (DT) ... 79
Table 12. Algorithm 4.2.4: Random Forest (RF) ... 81
Table 13: Algorithms performance on different ensemble techniques ... 85
Table 14: ML algorithms performance before applying ensemble techniques. ... 87
Table 15: Performance results after applying proposed Subspace Random ensemble. ... 93
Table 16. Algorithm 5.1.1: LSTM-Text-Classification algorithm ... 101
Table 17. Algorithm 5.1.2: BLSTM-Text-Classification algorithm ... 104
Table 18. Algorithm 5.1.3 : IDCNN-BiLSTM Text Classification algorithm ... 107
Table 19. Algorithm 5.1.4: CNN Text Classification algorithm ... 110
Table 20: DL Ensemble Methods and Algorithm Performance. ... 113
Table 21. Algorithm 5.2 : The proposed voting VEDLM Text Classification algorithm ... 116
Table 22: Algorithms performance before applying the proposed Ensemble approach. ... 119
Table 23: Performance results after applying the proposed voting ensemble approach ... 125
Table 24:Comparison of the Algorithms ... 138
Table 25: Comparison of the degree of information summarisation ... 139
Table 26: Comparison of scalability ... 140
Table 27: Evaluation of the performance of models ... 141
Table 28: Comparison of semantic characteristics ... 142
Table 29: Comparison of proposed VEDLM model performance with related studies. ... 145

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
AR	Action Research
BiLSTM	Bidirectional Long Short-Term Memory
CIA	Confidentiality, Integrity, and Availability
CNNs	Convolutional Neural Networks
CTI	Cyber Threat Intelligence
CVE-DB	Common Vulnerabilities and Exposures database
CVE-ID	Common Vulnerability and exposure identifier
CVEs	Common Vulnerabilities and Exposures
CVSS	Commonly vulnerability scoring system
DL	Deep Learning
DNN	Deep Neural Network
VEDLM	Voting Ensemble Deep Learning Model
EM	Expectation-maximisation algorithm
FN	False Negatives
FP	False Positives
GPS	Global Positioning System
IC	Intelligence Community
IDCNN	The Iterated Dilated Convolutional Neural Network
IoC	Indicator of Compromise
KB	Knowledge Base
LSTM	Long Short-Term Memory
ML	Machine Learning
NER	Named Entity Recognition
NLP	Natural Language Processing
NLTK	Natural Language Toolkit

KNN	K-Nearest Neighbours
RF	Random Forest
DT	Decision Tree
LR	Logistic Regression
SREMLM	Subspace Random Ensemble Machine Learning Model
NVD	National Vulnerability Database
PR	Precision-Recall
RNN	Recurrent Neural Network
ROC	Receiver operating Characteristic
SIEM	Security Information and Event management
SMOTE	Synthetic Minority Over-sampling Technique
SN	Social Network
SVCE	Security Vulnerability Concept Extractor
SVM	Support Vector Machine
SWRL	Semantic Web Rule Language
TN	True Negatives
TNR	True Negative Rate
TP	True Positives
TPR	True Positive Rate
TPR	True Positive Rate
UCO	Unified Cybersecurity Ontology

CHAPTER 1 : INTRODUCTION

This doctoral research thesis endeavours to develop a Subspace Random Ensemble Machine Learning Model (SREMLM) predictive modelling techniques and Voting Ensemble Deep Learning Model (VEDLM) introduced for the detection of cybersecurity threats on X (formerly Twitter). Detecting cybersecurity threats is crucial for social media platforms, benefiting both users and developers. It represents a vital method for interpreting and addressing threats and attacks in the digital landscape. By employing Machine Learning and Deep Learning algorithms within an artificial intelligence framework, I aim to create a robust system capable of detecting various threats and assessing their severity levels based on X data. Despite the demonstrated effectiveness of ML, DL models and computer vision technology in this domain, accurately detecting threats across multiple levels remains a challenging task that requires further refinement. This research focuses on developing and accessing an enhanced voting ensemble deep learning model and a Subspace Random Ensemble Machine Learning Model tailored to identify and detect threats users face through their tweets.

This chapter provides a foundational understanding of Cybersecurity Threats in X. It then delves into automated Cybersecurity threat systems developed using Machine Learning and Deep Learning. As a contemporary AI technique, Machine Learning and Deep Learning play a pivotal role in Cybersecurity threat detection. However, their application in automated systems presents unique challenges. Following this, I outline the research questions that will guide our investigation. The subsequent sections elaborate on the research aims, objectives, and significance. Finally, I conclude the chapter with a brief overview of the thesis' organization.

1.1 Background

The integration of artificial intelligence (AI) into the digital landscape has precipitated both significant advancements and formidable challenges in cybersecurity. While AI offers promising capabilities for threat detection, authentication, and incident response, it has also empowered adversaries to develop more sophisticated and evasive attacks. The complex interplay between AI algorithms, human behaviour, and malicious intent has created a rapidly evolving threat

landscape. AI-driven attacks, leveraging machine learning to bypass traditional defences, pose a serious risk to organisations across industries. Effective countermeasures demand a holistic approach that combines robust threat intelligence, adaptive defence mechanisms, and a strong ethical foundation. Leveraging AI for defensive purposes can enhance threat detection, automate response actions, and augment human analysts. However, challenges such as algorithmic bias, data privacy concerns, and the potential for AI-enabled attacks necessitate a comprehensive risk management strategy. To fully harness the potential of AI in cybersecurity, organisations must prioritise regulatory compliance, industry standards, and collaboration. Investing in cybersecurity education and training is essential to develop a skilled workforce capable of addressing emerging threats. By bridging the gap between theoretical understanding and practical implementation, we can effectively mitigate the risks associated with AI and build a more resilient digital (Familoni 2024). Artificial intelligence, machine learning, and deep learning are often confused with one another. To clarify their relationship, imagine them as nested circles. Artificial intelligence is the broadest concept, encompassing the entire field. Machine learning is a subset of artificial intelligence, representing a specific approach to achieving intelligent behaviour. Deep learning, a specialised form of machine learning, lies at the core and is driving much of today's AI advancements (Manakitsa et al. 2024). Figure 1 illustrates the relationship between artificial intelligence, machine learning, and deep learning.

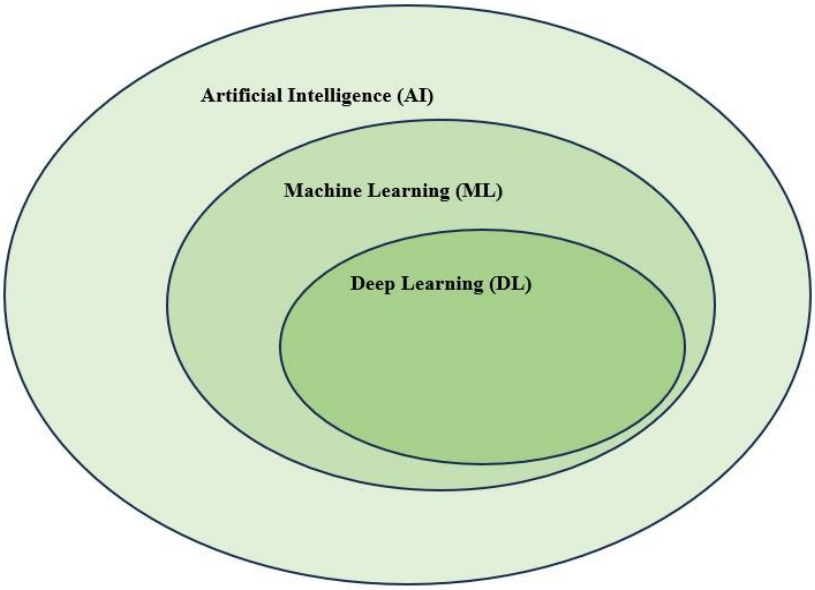


Figure 1: AI, Machine Learning, and Deep Learning: Understanding the Relationship, (Manakitsa et al. 2024).

Cybersecurity is a specialised field within information security that concentrates on safeguarding the confidentiality, integrity, and availability (CIA) of digital information assets. This protection is specifically targeted against threats that exploit the internet to compromise these assets (Von Solms & Von Solms 2018). The rise of information technology (IT) has fundamentally changed social media. But with this evolution, new dangers have also appeared. Social media platforms like Twitter, which store a vast amount of user data, have become prime targets for hackers. This has made cybersecurity a top priority for these platforms. The biggest threat today comes from cybercrime. Hackers can use various methods to attack social media platforms, including phishing, spoofing, identity theft, spyware, spamming, and denial-of-service attacks (Creado & Ramteke 2020). Their motives can range from damaging reputations and causing political unrest to stealing money.

Cybersecurity threats have become a major concern for social media platforms in recent years. This coincides with a booming cybersecurity market, which has grown roughly 35-fold in the past decade. In 2019, global cybersecurity spending reached $40.8 billion, rising steadily to $71.1 billion by 2022 (Research & Department 2023), As of 2023, spending topped $80 billion, and forecasts predict it will exceed $87 billion in 2024. This surge in cybersecurity spending reflects the increasing threat landscape. The digital economy's growth has unfortunately been accompanied by a rise in digital crime. The explosion of online and social media applications has created more opportunities for attackers, leading to data breaches that endanger both users and social media platforms. At the current rate of growth, the financial damage caused by cyberattacks is projected to reach nearly $10.5 trillion annually by 2025, marking a threefold increase from the levels recorded in 2015 (Aiyer et al. 2022). Global cybersecurity spending from 2017 to 2024 is illustrated in Figure 2.

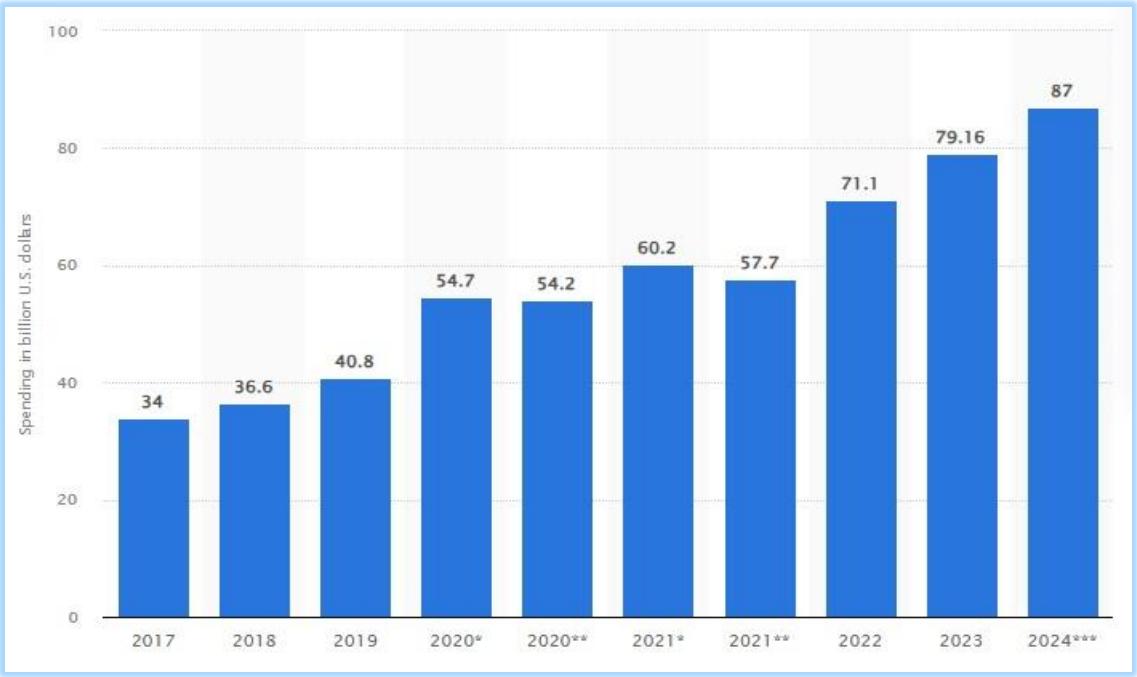


Figure 2: Global cybersecurity spending from 2017 to 2024, (Research & Department 2023) .

The surge of online social media platforms like X, Facebook, Instagram and TikTok reflects our evolving relationship with data sharing in the digital age. However, this convenience comes with a growing risk: cyber threats. Cyber threats involve criminals using technology to steal sensitive data, like users' information, through cyberattacks. This stolen data can then be used to conduct unauthorised activities online. Lost, stolen, or skimmed information can all be vulnerabilities for fraudsters. As the volume of social media platforms continues to climb, so does the threat of cyber threats, posing a serious challenge for both individuals and social media platforms (Kaur et al. 2024).

X comprises online services that enable users to establish a public or semi-public profile and connect with a list of other users to view and share their profiles and content. The association of X links differ from one service to another (Boyd & Ellison 2007). There is a growing range of X with several common features (Weir, Toolan & Smeed 2011). According to Boyd and Ellison (2007), social networks are online platforms where users can: 1) Create a public or partially public profile with limitations set by the platform, 2) Build a list of connections with other users they know, and 3) Browse their connections and connections of others to navigate the social network.

X offers a range of lucrative market prospects and advantages to businesses and organisations. As a result, over the past few years, the use of X has grown exponentially (Bello-Orgaz, Jung & Camacho 2016). X helps users to discover and extend their networks with new friends. An important characteristic of X is data sharing, where users can share their preferences, videos, images, events, and so on. SN such as X and Facebook, have been the dominant means of communication for billions of Internet users (Rathore et al. 2017). However, X continues to offer a rich environment for illegal activities that relate to cybersecurity incidents (Weir, Toolan & Smeed 2011). User information, including multimedia data, may be collected and inappropriately exploited by unauthorised users and third-party companies to maximise their income (Rathore et al. 2017). Cyber-attacks can occur at any point, and severely affect peoples' lives and trigger social and economic disruptions ((Fang et al. 2020). There is a constant demand for a novel and innovative set of security defences for cyber-attacks (Tounsi & Rais 2018). Therefore, companies need to collect and share real-time cyber risk data and turn them into cyber threat intelligence that can prompt timely cyber security incident response (Tounsi & Rais 2018). Early identification of cyberattacks will effectively minimise damage to users and related organisations and provide timely data and evidence of trial proceedings (Noor et al. 2019).

For the successful development of threat intelligence, Deep Learning can be useful in two main ways: firstly, processing an enormous amount of data (big data) and secondly maintaining the veracity and relevance of such data (Miret, 2020). Deep learning (DL) is the most effective technique for detecting cyber threats and overcoming the limitations of traditional security systems (Firdausi, Erwin & Nugroho 2010). The rise of cyber threats on X, fuelled by technological advancements, necessitates robust detection methods. This research delves into applying machine learning (ML) and Deep learning (DL) techniques, particularly those effective in cybersecurity anomaly detection, to create a framework for identifying cyber threats on X. By analysing real-world datasets, I aim to uncover hidden patterns and infer characteristics of threat activity. Our approach involves an independent validation of chosen ML, DL methodologies to ensure their effectiveness in intelligence cyber threats detection. Inspired by successful applications of ML, DL in cybersecurity anomaly detection. I will utilise various techniques to distinguish cyber threats based

on tweet data. Enhanced ML and DL models were developed and evaluated to detect cybersecurity threats on X.

1.2 Research problem

Intrusion detection is the process that monitors a network or system for malicious activity or policy violations. Any intrusion activity or violation is typically reported either to an administrator or collected centrally using a security information and event management (SIEM) system. A SIEM system combines outputs from multiple sources and uses alarm filtering techniques to distinguish malicious activity from false alarms, and it focuses on hardware (Bace 2000). On the other hand, threat detection is the process of monitoring any circumstance or event with the potential to adversely impact a user through unauthorised access, destruction, disclosure, modification of data, and/or denial of service (Rathore et al. 2017). This study will focus on cyber security threat detection.

Various types of X users report different cybersecurity attacks against them that aim to steal the identity of users or undermine the privacy and trust of the network. These threats include activities such as hijacking, identity theft, spamming, social phishing, malware attacks, face image retrieval and analysis, impersonation, fake requests and Sybil attacks (Zhang & Gupta 2018). Table 1 presents a comparative analysis of these threats to X users.

Table 1: Comparison of most popular attacks on online social networks, (Alsodi et al. 2021).

Measure	Impact on user	Effectiveness of server side protection mechanism	Effectiveness of user side protection mechanism	Threat to data privacy	Threat to data integrity
Identity theft	Average to high	Poor	Poor	Yes	Yes
Spam attack	Small	Strong	Poor	No	No
Malware	High	Medium	Medium	Yes	Yes
Sybil attack	Average	Strong	Poor	No	Yes
Social phishing	High	Poor	Strong	Yes	Yes
Impersonation	High	Poor	Poor	Yes	Yes
Hijacking	High	Poor	Poor	Yes	Yes
Fake requests	Small	Poor	Strong	Yes	No
Image retrieval and analysis	Average to high	Medium	Medium	Yes	No

Attackers, driven by various motives including personal and political reasons, focus on X.

As cyber threats become increasingly sophisticated, there has been a surge in research and development, resulting in a diverse array of security solutions. Watermarking (Zigomitros, Papageorgiou & Patsakis 2012), Steganalysis (Li et al. 2015), and digital oblivion (Stokes & Carlsson 2013) are some of the solutions for protecting Twitter users against threats from compromised multimedia data.

Likewise, traditional solutions such as spam detection (Miller et al. 2014) and phishing detection (Lee & Kim 2013) mitigate the conventional risks. There are also some established security solutions such as mechanisms for authentication (Joe & Ramakrishnan 2017) and privacy settings (Ghazinour, Matwin & Sokolova 2016) as well as commercial solutions such as minor monitoring and social protection applications that offer safeguards against cyberthreats in X. Thus, the traditional information security solutions that focus on heuristics and digital signatures are predominantly static and do not offer full protection against the dynamic nature of the new generation of cyber security threats that are more evasive, resilient, and complex (Tounsi & Rais 2018). However, existing cybersecurity solutions are not robust in detecting cybersecurity threats on X. This issue arises from two primary causes. Firstly, since the tweets are limited to 140 characters and the writing patterns of people are flexible, the meaning and context of words are also used and are varied (De Souza & Da Costa-Abreu 2020).Secondly, the flood of diverse and confusing advertisements, along with the misuse of hashtags for attention, creates a chaotic social media landscape. For these reasons, it is extremely difficult to detect cybersecurity threats from tweets (Fang et al. 2020).

Cybersecurity threats have become a critical concern in recent years with the growing popularity of social networks. X-based event detection has become a popular method of communicating such threats, and researchers have been using X as an extensive database for event analysis and extraction. Various techniques have been proposed for the detection of cyber security threats to X, focusing on attributes, frequency, and multimodal X hashtags. However, the current studies lack comprehensive evaluations of critical factors such as prediction scope, type of cyber security threats, feature extraction technique, algorithm complexity, information

summarisation level, scalability over time, and performance measurements (Coyac-Torres et al. 2023).

This study explored the following research questions:

RQ 1. What is the most effective machine learning technique for extracting and selecting features from X datasets to build a cybersecurity threat intelligence model?

RQ 2. What is the most effective deep learning technique for extracting and selecting features from X datasets to build a cybersecurity threat intelligence model?

RQ 3. How can an ensemble learning model be developed by aggregating predictions from pre-trained ensemble machine learning and deep learning architectures to improve predictive performance?

RQ 4. How effective are the proposed VEDLM cybersecurity threat models using X datasets?

1.3 Aim of this PhD thesis

This PhD thesis aimed at developing, validating, and evaluating a novel machine learning (ML) and deep learning (DL) model for cyber threat detection in tweets. The goal is to address the previously mentioned problems and enhance ML, DL techniques used in cyber threat detection.

This thesis tackles the limitations of machine learning and Deep Learning (DL) for cyber threat detection on X. The thesis proposes a novel solution that leverages advanced statistical metrics and visual analysis for robust verification. By thoroughly examining past research, the thesis develops a new model to enhance both accuracy and efficiency in threat detection. This contribution lies in applying and improving ML, DL techniques specifically for the challenges of X data. The result provides valuable insights and solutions for safeguarding the platform from malicious activities.

This thesis emphasises the importance of Machine learning (ML), Deep Learning (DL) techniques in enhancing cyber threat detection on X. The main objective of this PhD thesis is to develop a novel model consisting of ML and DL algorithms to detect cyber threats on X.

Cybersecurity on X requires understanding the key features of tweets vulnerable to surveillance. This allows for real-time threat detection, identification, and reporting. This study addresses the challenge of improving threat detection in tweets, given the rising frequency and evolving nature of cyberattacks. Existing independent DL models for threat detection struggle with complex threat behaviour and class imbalance issues. To address these limitations and minimise cyber risks, I propose a novel ensemble deep learning model that leverages voting for effective cyber threat detections. Despite the surge in machine learning (ML), Deep Learning (DL) techniques for cyber threat detection, a multi-faceted approach to this critical issue remains underexplored. This research aims to explore novel methods that combine various approaches for enhanced cyber threat detection performance.

The research aims to achieve the following objectives, which were identified to address the problems discussed previously:

1. To critically review the literature on cybersecurity, Machine Learning (ML), Deep Learning (DL), Artificial Intelligence (AI), and cyber threats within the Twitter platform.
2. To Build a proposed Subspace Random Ensemble Machine Learning model via trained K-Nearest Neighbours (KNN), Logistic Regression (LR), Decision Tree (DT), and Random Forest (RF) architectures.
3. To Build a proposed Voting Ensemble deep learning model by voting pre-trained LSTM, BiLSTM, IDCNN-BiLSTM, and CNN architectures.
4. To train and test the proposed novel (SREMLM), (VEDLM) models to detect and indicate potential cyber threats on Twitter.
5. To evaluate the proposed novel (SREMLM), (VEDLM) models to detect cyber threats in X by applying measurement techniques and comparing them with baseline models.
6. Collect and curate a new dataset to comprehensively assess the performance of the proposed (SREMLM), (VEDLM) models.

To detect threats from user tweets, we developed and evaluated cyber threat detection algorithms using a novel feature extraction model. As detailed in subsequent sections, the proposed VEDLM model demonstrated effective threat detection and high efficiency, as evidenced by its accuracy and performance metrics.

1.4 Significant contributions

Social media platforms, in today's fast-changing digital world, must prioritise improving security. For X, this means leveraging the newest technology to strengthen their systems. As they build a more robust information security foundation, cybersecurity becomes an essential tool.

The rise of online threats necessitates leveraging Machine Learning (DL) and Deep Learning (DL) for threat detection in social media like Twitter. This research proposes a cyber threat identification system using ML, DL methods. By integrating cybersecurity principles and tweet analysis, the system aims to identify threats from both traditional security logs and tweet history. Furthermore, the model utilises ML, DL to learn hidden patterns in data, offering new insights into cyber threats. This research contributes by proposing an exploratory model that explores relationships within the data to enhance threat detection.

While Machine Learning (ML) and Deep Learning (DL) have proven effective in detecting cyber threats, a critical gap remains in our understanding of the features redacted from real-world datasets. Past research has prioritised evaluating individual ML, DL methods, overlooking the intricacies of the cyber threats themselves within these datasets. As data breaches and cyberattacks rise alongside advancements in technology, addressing this knowledge gap is crucial (Fang et al. 2020).

Current cyber threat detection models often struggle with accuracy and false positives. This research proposes a novel approach utilising a voting ensemble deep learning model. This model aims to significantly improve threat detection accuracy while reducing false alarms. The findings of this study will contribute valuable knowledge to the field of cyber threat detection, aiding in the development of better mitigation strategies. Additionally, the proposed AI model has the potential to be implemented, leading to increased efficiency in threat detection, Reduced time spent investigating false positives, enhanced accuracy in safeguarding tweets from cyber threats.

X is a double-edged sword. It offers great ways to connect but also raises security and privacy concerns. The information users share can be used for criminal activities like identity theft. This makes them vulnerable to social engineering scams. Machine

and Deep learning can help Twitter identify these threats faster and more accurately, improving overall cybersecurity. This research will not only refine our understanding of cybersecurity threats on Twitter but also provide Twitter with practical tools to combat them. Twitter can significantly improve its cyber threat detection by utilising ML, DL algorithms. These algorithms offer an accurate, automated, and secure solution that saves time and resources. This empowers X to enhance its threat detection capabilities, streamline user service processes, and ultimately, safeguard its users from cybercrime and malicious activity.

This research aims to resolve the issue outlined in Section 1.2 and address the related research questions. The following outcomes have been achieved:

1. Development of a novel and effective machine deep learning classifier for automatic cyber threat activity detection.
2. Development of an innovative and effective joint deep learning classifier for automatic cyber threat activity detection.
3. Achievement of significantly improved performance in cyber threat classification through a proposed ensemble technique that combines individual machines and deep learning models into a new ensemble model.
4. A cyber threat detection framework that can be easily implemented as an AI algorithm on the X platform such as mobile applications and web portals to automatically manage the level of threat activities and can be practically applied to cyber threat detection tasks.
5. New datasets were collected in this research, creating a novel, large-scale X dataset meticulously curated to address the specific research objectives. This dataset constitutes a significant contribution to the field, providing a rich resource for future investigations.

1.5 Thesis organisation

This thesis is structured as follows:

Chapter 1 delves into the background and problem statement underlying the current research. It starts with an introduction to the histories of threat detection, emphasising the imperative need for automated threat detection tools. Furthermore, I explore the challenges associated with deep learning algorithms for threat detection

from users' tweets, finding key research questions. The chapter concludes by delineating the aims and original contributions of this thesis in detail.

Chapter 2 supplies an overview of various Deep Learning models for threat detection. A comprehensive literature review is presented, encompassing automated threat detection from tweets, Deep Learning feature extraction techniques, classifiers, and pertinent databases. The insights garnered from this chapter offer the groundwork to address research question 1, facilitating the development of enhanced deep learning models for threat detection and the identification of suitable databases for algorithm training and evaluation.

Chapter 3 expands upon the research method devised for this thesis. A research method serves as a roadmap for problem-solving and research execution. Here, I amalgamate the strengths of both scientific inquiry and action research to reach our research objectives. A framework is introduced for automatic threat detection, elucidating the phases and their interrelations. Furthermore, I detail the datasets employed for training and testing the algorithms, along with the evaluation metrics, validation approaches, and experimental configurations,

Chapter 4 introduces the Subspace random ensemble machine learning model (SREMLM) for cybersecurity threat detection on X. This model combines K-Nearest Neighbours (KNN), Logistic Regression (LR), Decision Tree (DT), and Random Forest (RF) algorithms to enhance accuracy and effectiveness. The chapter explores the model's architecture, including its machine learning techniques, a novel feature extraction method, and the SREMLM's unique design.
It also provides a detailed analysis of the results and compares the SREMLM's performance to existing state-of-the-art models.

Chapter 5 presents the Voting Ensemble Deep Learning Model (VEDLM), which combines LSTM, BiLSTM, IDCNN-BiLSTM, and CNN models to detect cybersecurity threats effectively and accurately on Twitter. This chapter delves into the models' architectures, including the employed deep learning techniques, a novel feature extraction algorithm, and the proposed VEDLM. Additionally, it presents a comprehensive analysis of the obtained results and a comparison with existing state-of-the-art models.

Chapter 6 provides a detailed comparison between the proposed SREMLM, VEDLM, and studies previously discussed in Section 2.3.3, all focused on cyber threats detection, these models were evaluated using the dataset described in Section 6.2.1. and shows that the proposed VEDLM integrates these models through a voting mechanism, achieving an exceptional and competitive F1 score of 99%. This integration illustrates that combining individuals' deep learning techniques within a voting framework significantly enhances predictive capabilities, yielding promising results for cyber threat detection.

Chapter 7 summarises the key findings of the thesis, discusses limitations, and suggests avenues for future research.

CHAPTER 2: LITERATURE REVIEW

In this digital age, social media platforms such as X have become an integral part of our daily lives. With millions of users sharing and exchanging information, the risk of cyber threats has increased significantly. In this chapter, the comprehensive background and state of current methods will be detailed and reviewed. The chapter will be divided into four sections. Section 2.1 will explore the various cyber security challenges and threats on X. Section 2.2 will examine the motivations behind these threats, including financial gain and political influence. Section 2.3 will present the current solutions being implemented to mitigate these threats including DL, ML and Ensemble Learning. Finally, Section 2.4 will summarise the chapter. This chapter supplies a comprehensive overview of the state of X cyber threats and will be valuable for researchers and practitioners working in the field of cyber security.

2.1 Cybersecurity challenges and threats to X

Cyber security is a tool to detect unwanted access to the property of individuals and organisations (Humayun et al. 2020). The cybersecurity community has established the field of Cyber Threat Intelligence (CTI). Cyber Threat Intelligence (CTI) has been receiving increasing attention from both academic and CTI researchers in security operating centres and security services providers as a component of cyber security (Dionísio et al. 2020). The primary objective of CTI is to develop a knowledge advantage over cyber threat actors. At the tactical and operational levels, CTI expedites early detection of malicious behaviours, preferably before a malicious actor gains a foothold in the network. On a strategic level, CTI provides sense-making and insight into the relevant threat environment to decision-makers. Effectively, CTI is the civilian, First-sector alternative to defensive counter-intelligence executed by the established Intelligence Community (IC) (Oosthoek & Doerr 2020).

Sharing Cyber Threat Intelligence (CTI) promises to be a new way of raising awareness among stakeholders about the situation (Sigholm & Bang 2013). The key idea behind risk intelligence sharing is to raise awareness among stakeholders of the situation by sharing information on current threats and vulnerabilities and to rapidly implement solutions (Wagner et al. 2019).

X has recently appeared as one of the most popular CTI sources that is used to gather information on vulnerabilities, threats and incidents, Several surveillance systems have been used for detecting CTI by collecting and analysing security vulnerability tweets and using supervised machine learning techniques (Koloveas et al. 2021).

X is a global platform with over 340 million users. It boasts 186 million daily active users who contribute to the staggering figure of 500 million tweets generated every day. X is an online service that allows users to create a public or semi-public profile and connect with other users to view and share their profiles and content. The way X links are associated with other services varies (Boyd & Ellison 2007). X is expanding its offerings and now includes a variety of features that are widely shared among users (Weir, Toolan & Smeed 2011). Boyd and Ellison (2007) defined X (then known as Twitter) as a web-based platform that enables users to create a public or semi-public profile within a specific context. Additionally, it allows users to create a list of connections with other individuals and to view and navigate through their connections and those of others.

X offers a variety of profitable market opportunities and benefits to businesses and organisations, leading to a significant increase in its usage over the past few years (Bello-Orgaz, Jung & Camacho 2016). X allows individuals to expand their connections and make new friends through its platform. One key feature of X is the ability to share various types of content, including preferences, videos, images, and events. Along with other social media platforms like Instagram and Facebook, X has become a major means of communication for a vast number of Internet users worldwide (Rathore et al. 2017).

Despite this, X remains a fertile ground for illegal activities related to cybersecurity threats (Weir, Toolan & Smeed 2011). User information, including multimedia data, may be collected and inappropriately exploited by unauthorised users and third-party companies to maximise their income (Rathore et al. 2017). Cyberattacks can happen at any time and can have severe consequences on individuals and society, causing disruptions in both social and economic areas (Fang et al. 2020). The need for new and creative security measures to defend against cyber threats is ongoing (Tounsi & Rais 2018).

X, with its various features such as tweets, video and image sharing, and e-commerce capabilities, has become an integral aspect of the daily routines of a vast number of internet users. However, this widespread utilisation of the platform also exposes individuals to a plethora of cyber threats and security concerns. The following section will outline these potential threats. As illustrated in Figure 3, there are several categories of security threats on social media.

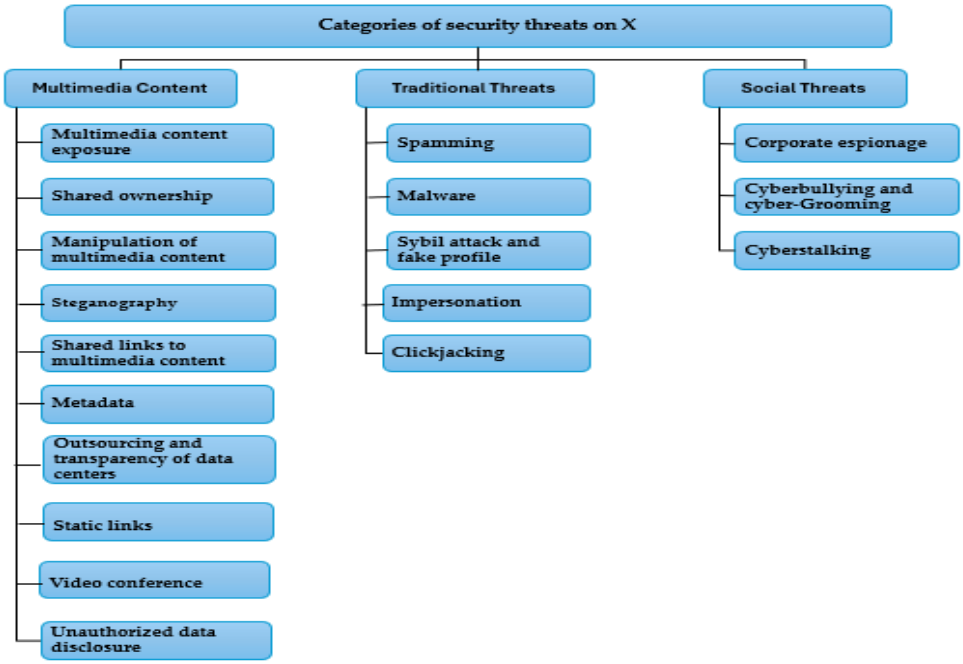


Figure 3: Categories of cyber threats on social media platforms (Rathore et al. 2017).

Cybersecurity threats occur more frequently with the popularity of today's use of X. Consequently, these threats may seriously impact the lives of individuals and cause social and financial unrest. Researchers have been using Twitter at least since 2010 as an extensive, publicly available database for analysing and extracting cybersecurity threats (Fang et al. 2020).

2.1.1 *Multimedia content threats*

X allows users to share various forms of data, including multimedia content, which has been improved by the integration of high-definition videos and images. However, multimedia search technologies, such as geotagging and facial recognition, can increase the potential for illegal use of shared data, putting sensitive user information at risk. This section focuses on the multimedia threats that attackers could exploit to

obtain sensitive user information from multimedia data shared on X (Rathore et al. 2017).

- **Multimedia content exposure**

It has been noted that individuals utilising social media platforms, such as X, exhibit a level of caution when it comes to exposing text-based information, such as their identification and home address. However, research indicates that users tend to be less cognisant of the potential implications of sharing multimedia data, which can reveal a significant amount of confidential information. For instance, the posting of an image of one's residence could potentially provide an intruder with the ability to locate the physical address of the user. Additionally, updates on the status of one's whereabouts (e.g., vacation, concert, etc.) may indicate that the user's home is unoccupied, thereby increasing the likelihood of intrusion. Furthermore, images shared on social media platforms can also reveal the current location of the user and their proximity to a given location, which can provide an added advantage to potential intruders. It is also worth noting that multimedia data shared on social media platforms may also draw unwanted attention to valuable assets or objects. Additionally, the sharing of photos or videos without the consent of the individuals depicted may compromise their privacy. Advances in technology, such as facial and voice recognition, have further complicated matters by enabling the identification of multiple individuals without their permission or knowledge. (de Andrade, Martin & Monteleone 2013).

- **Shared ownership**

Multimedia data shared on Twitter may relate to multiple users (González-Manzano et al. 2014). An illustration of this scenario would be two individuals who are friends attending an event together and subsequently capturing an image together. Subsequently, one of the friends may choose to upload the image to the X platform, without obtaining the consent of the other friend. This may result in the exposure of the other friend's privacy, as the image belongs to both individuals. It is important to note that the preferred privacy settings for multimedia data that pertain to multiple users are determined by a single individual, as opposed to being determined by the intersection of the privacy settings of each individual user, which would be a logical approach (Kaur et al. 2024).

- **Manipulation of multimedia content**

Twitter offers a medium for users to disseminate and access a plethora of multimedia content. However, the veracity and integrity of this content may be compromised by malicious actors who employ various digital tools to manipulate and distort multimedia data. This can lead to the unauthorised alteration of personal images, resulting in potential harm or defamation of legitimate users (Rathore et al. 2017).

- **Steganography**

Steganography is a technique that involves the concealment of data within other forms of media. As technology and scientific advancements have progressed, the utilisation of steganography has gained significant popularity and has been applied in a variety of legitimate contexts. Viejo, Castella-Roca and Rufián (2013) conducted a research study that revealed the existence of a distinct form of covert communication among X users, where they conceal their messages within images that are uploaded to the platform. This research demonstrates that not only is the technique of steganography feasible within the context of X but that it can also be executed with a relatively low level of technical complexity. Despite its many benefits, the utilisation of multimedia data on the social media platform X can also be exploited for nefarious purposes. The ability to conceal malicious intent within multimedia content can enable malicious actors to spread disinformation and potentially harm the reputation of the platform. Furthermore, this behaviour can also implicate innocent users, who may inadvertently come into contact with such maliciously embedded content, resulting in association with criminal activity. An example of this could be the dissemination of a seemingly innocent image, which upon closer inspection, contains embedded malicious code, by a malicious actor on X, and subsequently downloaded by an unsuspecting user (Gurunath et al. 2021).

- **Shared links to multimedia content**

The vast array of multimedia formats (e.g., JPEG, PNG) presents a hurdle in developing a universal framework. Many formats, especially interactive ones like Flash, are vulnerable to attacks or require manual validation. Social platforms, like X, often have limited multimedia support, restricting sharing options. For instance, X only accepts JPEG or PNG images and has limited GIF support. While users can share

unsupported content via links, this can be exploited by malicious actors. They might replace the linked content with harmful material, potentially redirecting users to malware or phishing sites. (Lee & Kim 2013).

- **Metadata**

Metadata is information about data. On platforms like X, multimedia content (like photos and videos) serves as metadata, revealing insights like user identities and locations. While this can be beneficial, it also poses risks. For instance, geotagging—embedding location data in images—can expose a user's privacy. This can reveal personal details like religious beliefs, political affiliations, or health conditions. Geotagged images can even lead to physical harm (Dressler, Bronk & Wallach 2015). Different social media platforms handle metadata differently. Facebook removes all metadata before uploading images, while Google+ retains most except for GPS coordinates. Flickr, by default, shares GPS coordinates to connect users with images from the same location (Van Laere, Schockaert & Dhoedt 2013).

- **Outsourcing and transparency of data centres**

Social network users face significant privacy risks due to the lack of encryption for stored multimedia data. This vulnerability allows malicious actors to access unencrypted content directly, bypassing any authorization processes. Additionally, the data stored on social networks is accessible by the service providers themselves, raising concerns about data privacy. While larger networks may operate their own data centers, smaller platforms often rely on third-party cloud storage, introducing additional risks due to potential data breaches and privacy violations (Singh, Jeong & Park 2016). While users may trust Twitter, they often worry about how their data is used by third parties. This data can be shared with authorities for investigations or used by businesses for marketing (Fang et al. 2020).

- **Static links**

While static links are a popular method for sharing multimedia content, they pose significant privacy risks. Anyone with a static link can access and distribute the content without the owner's permission. This can lead to unauthorized sharing on unintended platforms and jeopardize user privacy (Rathore et al. 2017).

- **Tagging linkability from shared multimedia data**

X has a feature where you can tag multimedia content, such as videos and images, to increase interaction between users and make searching easier. People can label their own content and add more details, but this can also be a threat to their privacy. For instance, some X users may not want to share their own photos, but a friend could tag their photo to reveal their identity (Squicciarini, Shehab & Wede 2010). The primary concern is that tagging can connect an individual who doesn't have an X account and doesn't wish to reveal any personal details on the platform (González-Manzano et al. 2014). Additionally, a spammer or an individual with malicious intent can tag a substantial number of individuals in a single post, such as an image or video, to disseminate harmful content to a wide audience with minimal effort (Ahmed & Abulaish 2013).

- **Unauthorised data disclosure**

X offers its users the ability to share data. Sharing data involves making it available to a specific group of users. However, there is a risk that one of the members of the group may disclose the shared information (Viejo, Castella-Roca & Rufián 2013). This type of disclosure is often considered illegal as it can be manipulated. The same goes for multimedia data, such as pictures. When a user shares a picture with a group, any member of the group can download it and change the privacy settings, potentially causing the picture to be publicly shared even though the original uploader only wanted it to be seen by a certain group of people.

- **Video conference**

Today, X offers both text messaging and video conferencing capabilities. The added benefit of video conferencing is that it allows for greater interaction between users. However, this also opens up the possibility of more sensitive information being shared. A malicious user can access the video stream by exploiting any vulnerabilities in the communication infrastructure (Ramzan, Park & Izquierdo 2012). Additionally, someone participating in the video conference can record it and use it to blackmail or manipulate others. The attacker may also be able to access the webcam of the target by utilising malware and taking advantage of weaknesses in the communication protocols.

2.2.1 *Traditional threats*

In the context of X, there are specific types of traditional threats that involve utilising various attack methods, such as phishing and malware, to acquire a user's personal details. This information can provide a significant advantage for the attacker, as they can obtain sensitive information like social security numbers, passwords, and bank information. With this information, the attacker can carry out further crimes such as phishing and identity theft (Lanza & Lodi 2024). This section outlines the different traditional threats that can be employed by attackers to access a user's personal information.

- **Spamming**

Spam attack attackers flood Internet users with unsolicited messages (spam). On X, this kind of attack appears to be more successful than traditional spam attacks that use email to spread spam. This is because the social connections between X users can be easily abused. Target users can easily be convinced to read spam information and trust it to be safe. Here, the attacker can somehow obtain communication details about the user and send spam or junk data. Obtaining communication details is not too difficult and can be extracted from legitimate user profiles. A large amount of spam emails sent causes network congestion and the cost of sending emails is mainly borne by the provider of the service and in some cases by the user (Zhang & Gupta 2018).

- **Malware**

This is harmful software made up of Trojan horses, virii, and worms. X operates by connecting different users' systems. As a result, malware can easily spread from one user's system to another through these connections.(Nauman, Azam & Yao 2016). X lacks the necessary tools to identify if a URL is dangerous or not. Dangerous URLs can steer users to fake websites which can then transmit malware to their computers and steal their confidential information. Faghani and Saidi (2009) looked at the spread of malware on X and determined which factors played a role in its spread. These factors include features of the social network graph such as the number of nodes, number of connections, highest degree, average shortest distance, and longest distance. The researchers also explained how each factor affects the rate at which malware spreads on X (Lanza & Lodi 2024).

- **Sybil attacks and fake profiles**

In a Sybil attack, attackers generate a significant number of fake identities to gain an advantage in distributed and peer-to-peer systems. This type of attack poses a significant threat to security as it has a large number of users connected as peers in a peer-to-peer network, allowing one entity to control multiple fake identities. By utilising these fake identities, attackers can override legitimate users and manipulate reputation values, corrupt information, and outvote legal Twitter users, such as by voting an account as the "best" (Noh et al. 2014).

- **Impersonation**

The goal of the attacker is to construct a false profile with the intention of pretending to be a real individual. This type of attack heavily relies on the authentication procedures that users encounter when creating a new account. Such attacks can have severe consequences for the person being impersonated (Zhang & Gupta 2018).

- **Clickjacking**

This is an escalating threat to X, where attackers hide harmful software behind sensitive user interfaces or buttons to manipulate user clicks for malicious intent. Clickjacking takes several forms, with the most notable being Likejacking and Cursorjacking. In Likejacking, the attacker embeds malicious scripts within X's "Retweet button" that appears on the user's profile. Cursorjacking uses interface redressing to alter the cursor's position, replacing the real cursor with a fake one to lead users to a malicious website. (Faghani & Nguyen 2014).

- **Social phishing**

This type of attack involves the attacker attempting to obtain confidential information from a target by using a fake website that appears authentic or by pretending to be someone the target knows. The severity of these attacks can be significantly reduced if the target is informed and cautious when reviewing the information received (Jagatic et al. 2007).

- **Hijacking**

Gaining control over another person's profile is referred to as hijacking. The attacker succeeds in this if they are able to guess or obtain the login password for the account. Choosing weak passwords is not recommended as it increases the risk of

hijacking. These passwords can easily be acquired through dictionary attacks. To prevent this, it is best to use strong passwords and change them frequently (Zhang & Gupta 2018).

2.3.1 *Social threats*

Online social threats are becoming increasingly sophisticated, often leveraging the interpersonal connections formed on platforms like X to target vulnerable groups, such as minors and corporate employees. Attackers may use trust-building tactics like empathy and material incentives to manipulate their victims. Their malicious intentions can range from blackmail and cyberbullying to espionage and the distribution of pornographic content. In this section, we will explore the diverse social engineering attacks that exploit online relationships for nefarious purposes.

- **Corporate espionage**

Corporate espionage can employ automated social engineering tactics through Twitter. By utilising X as a tool, a social engineer can obtain valuable information, such as the job title, email, and complete name of employees, without relying on traditional social engineering methods and infiltrating the company. A study by Krombholz et al. (2015) describes a method of using social networking sites (Twitter) to execute a social engineering attack. They demonstrated that by utilising Twitter, an attacker can gather information about an employee within a targeted organisation in an automated fashion, which can then be utilised for a successful social engineering attack (Shah, Varshney & Mehrotra 2024).

- **Cyberbullying and cyber-grooming**

Cyberbullying refers to the intentional and ongoing online harassment or harm towards an individual. Cyber-grooming, on the other hand, involves an adult using the internet to build an emotional connection with a child for the purpose of sexual abuse. Children are particularly vulnerable to these types of online threats and attacks due to their young age (Diomidous et al. 2016). Teenagers who experience cyberbullying are at risk of developing depression. Online predators may try to manipulate them by pretending to be compassionate, loving, and generous by spending a lot of time online and offering gifts, money, and other incentives. According to security experts, these predators have targeted thousands of students worldwide through threat activities. The Megan Meier case is one of the most shocking examples of cyberbullying, which

resulted in a teenage girl taking her own life. The perpetrator successfully created a fake online profile and used it for other forms of cyber-grooming (El Asam & Samara 2016).

- **Cyberstalking**

X users have the option to reveal their personal details such as contact information, home address, location, and schedule on their X profile. However, this information can be vulnerable to exploitation by malicious individuals for cyberstalking purposes. For example, an attacker can blackmail their victim through phone calls or instant messages on X. Additionally, users often share location information through their photos, which attackers can gather and use for harmful cyberstalking attacks. Dreßing et al. (2014) reviewed the effects of cyberstalking on German X users on StudiVZ. They emphasised that cyberstalking could harm the mental well-being of X users and should be regarded as a significant danger to ensure a safe and secure environment on the platform.

2.2 Motivations for the cyber threats on X

Attackers, commonly referred to as hackers, have become a major concern for X users in recent years. These individuals carry out various attacks on X with different motivations behind their actions. Some attackers may be seeking revenge or driven by negative emotions, while others may be motivated by financial gains. Some hackers may simply be looking for entertainment, while others may be part of hacktivist groups who use their skills to protest against certain issues. Additionally, there are those who engage in espionage or cyber warfare, using their hacking skills for political or military purposes. Regardless of their motivations, the impact of these attacks on X can be significant and cause major harm to individuals and organisations. It is important for X users to be aware of these motivations and take steps to protect their online presence (Kaur et al. 2024).

- **Financial benefits**

Financial benefits are the primary motivation behind cyber-attacks on X. These attacks are carried out by cybercriminals who aim to acquire sensitive information related to the bank accounts of users. Malicious access to these accounts allows the perpetrators to steal money and financial assets from the victims. Additionally,

business-related information can also be targeted in these attacks, with the intention of profiting from the information by rival companies. The ease of access to large amounts of personal and financial information on X makes it a prime target for cybercriminals looking to make quick and easy financial gains (Munk 2022).

- **Entertainment**

Entertainment can come in many forms and for some hackers, it lies in the excitement of hacking on social media. These individuals are driven by the thrill of showcasing their hacking skills to their peers and gaining recognition in the hacking community. They do not have any financial or political motives behind their actions, but simply do it for the enjoyment of the challenge. As the saying goes, some people just find pleasure in causing chaos and disruption. For these hackers, hacking is a form of entertainment that allows them to express their technical abilities and gain a sense of notoriety among their peers (Kaur et al. 2024).

- **Cyber spying**

Cyber espionage refers to the act of obtaining information without the permission of the owner using hacking techniques and malicious software. This type of espionage is becoming increasingly prevalent on social media, where individuals, competitors, and even foreign governments are targeting confidential information. This can range from personal data to sensitive business information and can have serious consequences for those affected. The rise of cyber espionage highlights the importance of taking necessary precautions to protect personal and business information online (Akoto 2024).

- **Expertise for the job**

The demand for expertise in the fields of cybersecurity and hacking is at an all-time high, as many IT experts lack these specific skill sets. The job market for these positions is extremely competitive, as organisations are eager to hire individuals who can help them evaluate their security and protect against cyber criminals. Having a specialist on their team allows companies to think and operate in the same way as the criminals, giving them a better chance at beating them. The need for these experts is crucial in today's world, as cyber threats continue to grow and evolve (Graham & Lu 2023).

- **Cyber warfare**

Cyber warfare is a new form of conflict that is fought through the use of technology and the Internet. It is a politically motivated attack on information and information systems, mainly targeting government websites. The goal of these attacks is to disrupt the communication and financial stability of the targeted country and to cause improper functioning of its government. Unlike traditional warfare, cyber warfare is fought from the comfort of a room rather than on the front lines. The use of social media has made it easier for individuals or groups to launch these attacks, making it a serious threat to national security (Dawson Jr 2021).

- **Revenge/feelings.**

Revenge and emotions can drive individuals to engage in cyber-attacks on X. Whether it's a dissatisfied customer or an unhappy employee, the desire for revenge can lead to the destruction of an organisation's reputation. These hackers aim to cause chaos and frustration by blocking services and leaving legitimate users without access. The impact of such attacks can be devastating, causing significant financial loss to the victim’s organisation. It's important to recognise the power of emotions and the potential consequences they can have in the digital world (Gadekar & Rakshit 2020).

- **Hacktivism**

Hacktivism is a form of activism that utilises technology to achieve political and social goals. The main objectives of hacktivism include promoting free speech, protecting human rights, and advancing information ethics. This type of activism involves publishing the views and aims of a political community or religious group and staging protests to support their beliefs. However, it can also involve vandalism of websites with political or religious messages. Hacktivism is a unique form of activism that combines technology and activism to bring attention to important political and social issues (Romagna & Leukfeldt 2024). Figure 4 illustrates the impact and motivation of cyber threats on social media.

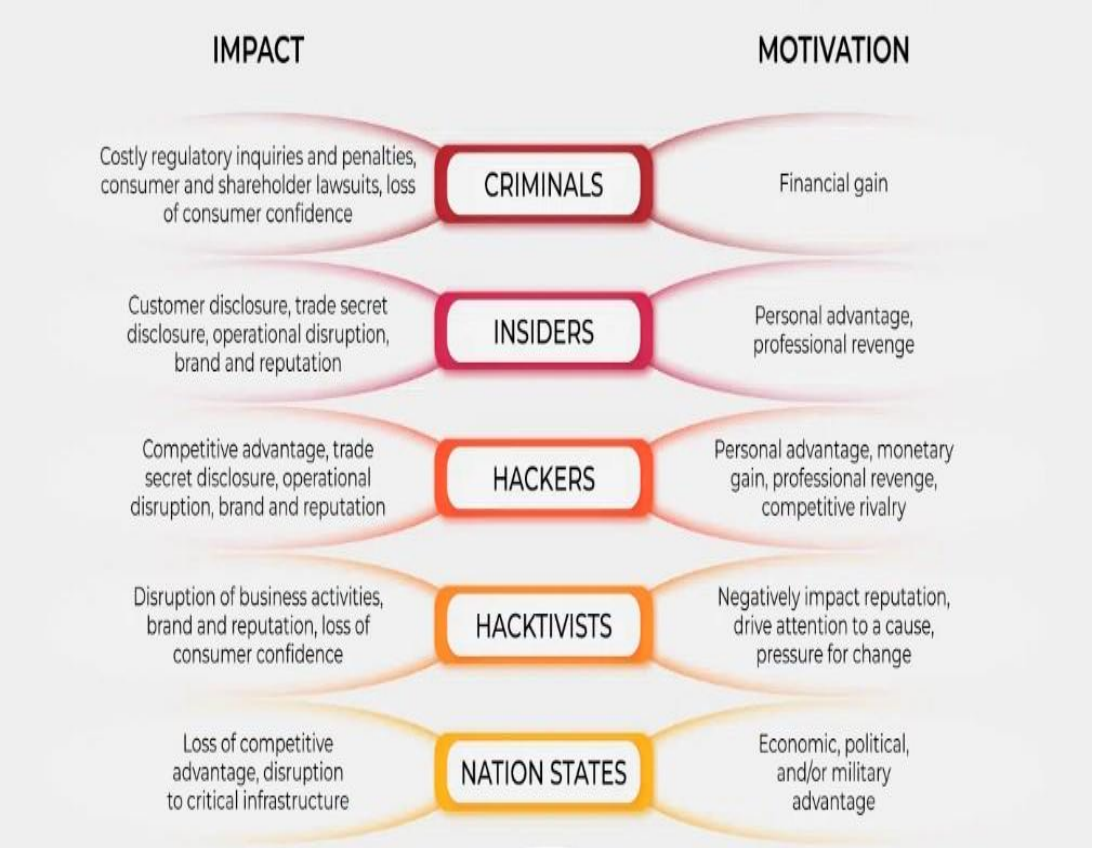


Figure 4: Example of the Cyber threats impact and motivation on social media, (Akoto 2024).

2.3 Cyber threat detection on X: ML, DL-based solutions

AI algorithms are instrumental in machine learning's pattern recognition capabilities. They can be broadly categorised into supervised and unsupervised types. Supervised algorithms, trained on labelled data to predict image classes, include parametric models like support vector machines and non-parametric methods such as K-Nearest Neighbours. Conversely, unsupervised algorithms, operating without labelled data, uncover patterns and structures through clustering and dimensionality reduction techniques. The selection of the optimal algorithm hinges on factors like accuracy, scalability, and the specific problem at hand. While initially met with scepticism, AI's potential benefits have gained widespread recognition. Aiming to replicate human intelligence, AI encompasses machine learning and deep learning, with computer vision as a critical component. Understanding this interplay is essential for appreciating the advancements in machine vision (Manakitsa et al. 2024).

X has become a significant platform for the dissemination of information, including cyber threats. This presents both challenges and opportunities for cybersecurity researchers. Deep learning offers powerful tools like Machine Learning to analyse this vast and dynamic data stream, enabling more effective threat detection, response, and prevention (Lughbi, Mars & Almotairi 2024). Machine Learning, Deep Learning and Ensemble Learning for cybersecurity threat detection are explored in the following sections.

2.3.1 Machine Learning (ML)

Machine Learning (ML) is a branch of Artificial Intelligence (AI) and computer science that focuses on using data and algorithms to enable AI to imitate the way that humans learn, gradually improving its accuracy (Btoush et al. 2023). Machine learning can create an effective model automatically based on initial training data. The motivation for this approach is the availability of the appropriate training data, or it can be obtained at least more conveniently compared to the effort required to define the model manually (Omar, Ngadi & Jebur 2013).

This section delves into the three primary methodologies in this field: supervised, semi-supervised, and unsupervised learning. It provides a contextual background and a comprehensive analysis of key research within each category.

2.3.1.1 Supervised learning

Supervised learning is a machine learning method that trains a model on a dataset of labeled examples, where each example comprises an input (typically a vector) and a corresponding output or label. By analyzing the patterns and relationships within the training data, the model learns to predict the output for any valid input. These predictions can take the form of regression, which involves predicting a continuous numerical value such as house prices, or classification, which involves assigning inputs to discrete categories, such as determining whether an email is spam or not. In order to do so, the learner must "reasonably" generalise from the data given to unseen circumstances (Ayodele 2010). In other words, in terms of predictor characteristics, the goal of supervised learning is to create a concise model for the distribution of class labels (Kotsiantis, Zaharakis & Pintelas 2007). Supervised learning is used, in particular, as a predictive mechanism in which a portion of the data is learned (or otherwise known as a training set), while another portion is used to test

a trained model (Cross-validation) and the remainder will be used to determine the accuracy and effectiveness of the forecast (Hatcher & Yu 2018).

Cyber threat analysis primarily relies on two types of features: behavioural and content. Behavioural features focus on user metadata, actions, and interactions, without deep content analysis. They examine factors like timestamps and basic text counts. Content features delve into the textual content itself to differentiate bots from real users.

A. Behaviour-based

The well-known BotOrNot (Ferrara et al. 2016) is an off-the-shelf system that leverages more than one thousand features to discriminate bots. It measures the ‘botness’ of a X account. The authors expanded on their previous work (Varol et al. 2017) by retraining the model on a new dataset (Lee, Eoff & Caverlee 2011) and disclosing their feature engineering process. Kantepe and Ganiz (2017) developed a feature set inspired by the DARPA competition (Subrahmanian et al. 2016) to distinguish between normal and bot accounts. David, Siordia and Moctezuma (2016) identified and evaluated the importance of features for Sybil detection on X, finding Random Forest to be the most effective classifier. Khaled, El-Tazi and Mokhtar (2018) combined Support Vector Machines and Neural Networks (SVM-NN) to detect fake accounts and bots, reducing the feature set from (Yang, Harkreader & Gu 2013) to improve efficiency.

(Velayutham & Tiwari 2017) developed a method to calculate a 'botScore' for X accounts, similar to the BotOrNot ‘botness’ score. They identified ten user profile attributes and tweet patterns to feed into their BotClassifier, a supervised classification algorithm. Compared to Naive Bayes, their model demonstrated superior performance in distinguishing between human and bot accounts.

CATS by Amleshwaram et al. (2013) uses a clever approach to spot X spam bots. By analysing just five tweets per user, they combine entropy, spammer behaviour, and a blacklist of spammy URLs. This helps them accurately identify spam accounts. The CATS team also introduced 15 new features for better spam detection. They tested different Machine Learning methods and even grouped spammers to understand how they operate.

Ji et al. (2016) conducted an empirical study on the evasion tactics used by social bots. They identified key characteristics of social bots and common evasion techniques and subsequently proposed a detection method that incorporated nine novel features alongside existing ones. Their approach was evaluated across multiple social media platforms.

Teljstedt, Rosell and Johansson (2015) introduced a hybrid approach combining human judgment and machine learning to identify X bots. This semi-automated method prioritises precision, making it suitable for creating large, high-quality datasets for bot detection models.

Gilani, Kochmar and Crowcroft (2017) developed a novel approach to classifying Twitter accounts by stratifying users based on account popularity. This strategy, centred on user social status, allowed them to identify distinct feature sets effective for distinguishing between human and automated accounts within each popularity tier. While their study focused on general account classification rather than specifically detecting malicious bots, the methodology, features, and dataset generated could serve as valuable foundations for future bot detection research.

Similarly, (Daouadi, Rebaï & Amous 2019) introduced a refined set of features focusing on user interaction levels and engagement. They combined these features with existing ones to detect X bots using deep learning. In a similar vein, Yang et al. (2014) identified bots in marketing campaigns by analysing user interactions on X. They compared various classifiers and found Back Propagation Neural Networks to be most effective with their feature set.

Pattern recognition techniques have been applied to classify X accounts. Chu et al. (2010) developed a model to categorise accounts as human, bot (spam bot), or cyborg. Their approach involved analysing account behaviour through entropy calculations for tweet timing patterns, machine learning for text-based spam detection, and statistical analysis of account properties. A decision-making component combined these analyses for final classification. In subsequent work by Chu et al. (2012), the model was refined with enhanced components and evaluated on a larger dataset.

Gurajala et al. (2015) developed a method to identify automated fake X profiles by analysing multiple profile attributes, screen name patterns, and tweet posting times.

While their model exhibited high precision in detecting fake accounts, its recall was relatively low. Nonetheless, due to its exceptional accuracy, the authors propose using it as a baseline or starting point for more complex graph-based detection methods.

Caruccio, Desiato and Polese (2018) developed an algorithmic approach to identify distinct behavioural patterns between real and fake X users.

Their method focuses on extracting Relaxed Functional Dependencies to differentiate between human and bot accounts. The researchers posit that the complex patterns exhibited by humans are inherently difficult for bots to replicate. Valliyammai and Devakunchari (2019) introduced a proactive method to identify Sybil accounts during their creation. By comparing private user data and images, their framework can prevent these fraudulent accounts from being established.

Additionally, Cai, Li and Zeng (2017) developed a model that represents social media users based on their behaviour and posting patterns. This model, utilising a CNN-LSTM algorithm, was employed to distinguish between human and bot accounts on X.

B. Content-based

Numerous studies have focused on content analysis and textual information to detect X bots. For instance, Kudugunta and Ferrara (2018) employed deep learning to identify bots using a single tweet and six account features. They addressed dataset limitations by applying oversampling techniques to a small training set.

Similarly, Wang, W. et al. (2018) leveraged tweet similarity to detect social bots, assuming similar botmaster objectives and technological constraints lead to comparable tweet content. Ping and Qin (2018) employed a CNN-LSTM algorithm on tweet content and metadata to identify evasive spam bots.

Given the role of bots in misinformation spread, Morstatter et al. (2016) proposed using topic analysis for bot detection. Their BoostOR algorithm optimised F1 score by balancing precision and recall. Notably, they introduced two publicly available labelled X datasets.

Igawa et al. (2016) classified Twitter accounts into human, bot, and cyborg using pattern recognition and a wavelet-based approach. Random Forest outperformed

Multilayer Perceptron, especially in the binary human/non-human classification. Jr et al. (2018) extended this work to distinguish between humans, legitimate bots, and malicious bots.

Presuming similar patterns in spam generated by the same botmaster; Bara, Fung and Dinh (2015) developed an iterative model to detect X spam and spam bots based on tweet similarity and closeness to known spam.

While not directly focused on bot detection, Gupta et al. (2017) classified X users into person and non-person. Their first step involved using Twitterati Main and Shekokhar (2015) to identify bots based on tweet properties like inter-tweet delay, spam detection, near-duplicate tweets, Klout score, and tweeting device.

Dickerson, Kagan and Subrahmanian (2014) hypothesised that sentiment differences could distinguish humans from bots. They introduced sentiment-based features alongside other tweet and user characteristics. Loyola-González et al. (2019) also used sentiment analysis with a Contrast Pattern-Based classifier. Andriotis and Takasu (2018) employed content and metadata information to detect social spam bots.

In a different approach, (Beskow & Carley 2019) adopted a different approach by focusing on usernames rather than user posts. They categorised usernames as either random or non-random, creating a dataset of 235,000 X accounts with random usernames, which they labelled as automated. An analysis of a 100-account sample from this dataset led them to conclude that it is accurate and diverse, making it a valuable resource for improving bot detection on social media.

2.3.1.2 Unsupervised learning

Unsupervised learning of these techniques does not require training data. They are based, as alternatives, on two fundamental assumptions. Firstly, they assume that daily traffic is the majority of network connections and that only a very small percentage of traffic is abnormal. Secondly, malicious traffic is calculated to be statistically different from regular traffic (Omar, Ngadi & Jebur 2013). According to these two assumptions, daily traffic is typically presumed to be data groups of similar instances, although occasionally instances that differ significantly from most instances are considered to be malicious (Zhang, Zhang & Sun 2009). Data sets provided as machine learning input in unsupervised learning are not labelled in any way that defines the correct or

incorrect outcome. Instead, the result may achieve a larger desired target, be measured on the ability to find something readily discernible by humans or provide a nuanced application of the statistical function to obtain the intended value (Hatcher & Yu 2018). Similar to supervised learning articles, those employing unsupervised methods are categorised into behaviour-based and content-based approaches. A review of these unsupervised techniques follows.

A. Behaviour-based

Several models have been proposed to detect social media bots using unsupervised Machine Learning techniques.

DeBot by Chavoshi, Hamooni and Mueen (2016) identifies bots based on correlated activity patterns. Assuming human users exhibit less correlated behaviour over time, DeBot flags accounts tweeting frequently (at least 40 tweets/hour) with high activity correlation as potential bots.

Cresci et al. (2016) introduced the Digital DNA model, which analyses the sequence of online actions to identify bot campaigns. Accounts with similar action sequences (Longest Common String) are classified as potential spam bots. In their subsequent work, Cresci et al. (2017) applied this model in both supervised and unsupervised settings, favouring the latter.

Minnich et al. (2017) employed a similar approach in their BotWalk model. By constructing vector representations of user features, BotWalk utilises seed bots and these vectors to identify social bots on Twitter. Seed bots are discovered using DeBot (Chavoshi, Hamooni & Mueen 2016), and the model then expands its search to connected users to detect anomalous accounts. The dataset used for this research is publicly accessible.

B. Content-based

To disseminate information effectively, bots typically exhibit openness and content duplication. Chew (2018) exploited these characteristics to identify patterns of similarity and subsequently detect automated X accounts, commonly referred to as Influence bots. By analysing tweet data, the author discovered emerging patterns among groups of accounts, positing that these patterns alone suffice to classify accounts as

automated without requiring additional ground truth verification. Building on this concept, Chen et al. (2017) developed a method to detect spam bot campaigns on X by examining patterns in URL shortening services and comparing content similarity between tweets. In a subsequent study, Munschauer et al. (2018) designed a system capable of identifying spam bot campaigns on the X platform. The system identifies groups of accounts sharing identical tweets by monitoring top trending URLs on X's real-time stream. Accounts within these groups are flagged as potential bots if they exhibit similar recent tweeting behaviour. A classifier is then employed to distinguish spam bot campaigns based on shared tweet content. Finally, the system links each identified campaign to the email address associated with the URL it promotes.

Abu-El-Rub and Mueen (2019) employed a content-based approach to identify Small and Medium-Sized Businesses (SMBs) within the BotCamp dataset. Their model capitalised on trending topics to detect social threats campaigns focused on political discourse. By gathering trending hashtags, the model employs DeBot by Chavoshi, Hamooni and Mueen (2016) to identify synchronised bots that exploit popular hashtags. Subsequently, graph-based techniques are used to represent topological relationships between these bots and group them into clusters. A supervised model is then applied to categorise user interactions as either agreeing or disagreeing with specific sentiments. Ultimately, the identified clusters serve as indicators of bot-driven campaigns within political discourse.

2.3.1.3 Reinforcement learning / Semi-supervised learning

Semi-supervised learning is a learning technique dealing with the study of how machines and natural systems, such as humans, learn in the presence of both labelled and unlabelled data. Traditionally, learning has been studied either in the unsupervised paradigm where all data is unlabelled (e.g., clustering, outlier detection) or in the supervised paradigm where all data is labelled (e.g., classification, regression) (Zhu & Goldberg 2009). In recent years, interest in SSL has increased, especially because of application domains in which unlabelled data such as images, text, and bioinformatics are abundant (Chapelle, Schlkopf & Zien 2010). The aim of the reinforcement learning approach is to maximise the reward of each change of state by learning the best behaviour to be performed in each state (Hatcher & Yu 2018).

Shi, Zhang and Choo (2019) introduced clickstream sequences as a robust feature to differentiate human users from social bots. By employing semi-supervised clustering, they leveraged the dynamic nature of clickstream data to unveil subtle behavioural patterns that are challenging for bots to replicate. This approach assumes that clickstream sequences encapsulate both the evolving aspects of user behaviour and underlying, consistent characteristics. Leveraging the principle of homophily in social networks, Dorri, Abadi and Dadfarnia (2018) developed SocialBotHunter, a model that detects spam bots on X by analysing user behaviour and interactions. Requiring only a seed set of labelled legitimate users, the model effectively identifies spam accounts.

2.3.2 Deep Learning (DL)

Deep Learning, an advanced form of machine learning, surpasses shallower models by utilising intricate algorithms that mimic human thought processes, allowing for deeper understanding and analysis of complex data (Manakitsa et al. 2024). Deep learning enables computational models consisting of several layers of processing to learn data representation at multiple abstraction levels. These methods have greatly improved state-of-the-art speech recognition, visual object recognition, object detection, and many other domains such as drug discovery and genomics (LeCun, Bengio & Hinton 2015). The use of deep learning technology for cybersecurity research and intrusion detection is highly important since most attacks use invasive software families that can be detected and classified (Hatcher & Yu 2018). Deep learning is commonly used in pattern recognition. Furthermore, the issue of classification, such as text classification and image classification, has also shown efficiency when deep learning is used (Fang et al. 2020). The following is a brief exploration of deep learning techniques.

- **Convolutional Neural Networks (CNNs)**

ConvNets are built to handle data represented as multiple arrays,such as a colour image consisting of three 2D pixel-intensity arrays in three colour channels. There are many data modalities in the form of multiple arrays and 1D for signals. Sequences, like language; 2D images or audio spectrograms; and 3D images, either video or volumetric. The four key ideas behind ConvNets that take advantage of the characteristics of natural signals are local connections, shared weights, pooling, and

the use of multiple layers (LeCun, Bengio & Hinton 2015). Convolutional networks integrate three architectural ideas to ensure a certain degree of transition, size, and distortion invariance: 1) local receptive fields; 2) shared weights (or duplication of weights), and 3) spatial or temporal subsampling (LeCun et al. 1998). Xu et al. (2016) also suggested CNN strive for image recognition. The core concept of Convolutional Neural Networks (CNNs) involves using a kernel, or convolution matrix, to scan across different regions of an image, effectively representing the data function. Unlike traditional neural networks that might lose spatial information, CNNs maintain it by moving the kernel over the entire image. This concept can be extended to Natural Language Processing (NLP) by applying the convolutional layer of CNNs to the vector space representation of a text corpus. Each kernel can learn to identify patterns within specific regions, such as sentences, capturing both semantic and structural features. This ability makes CNNs particularly effective for text classification tasks. Fang et al. (2020) proposed a multi-task learning approach based on the natural language processing technology and machine learning algorithm of the Iterated Dilated Convolutional Neural Network (IDCNN) and Bidirectional Long Short-Term Memory (BiLSTM) to establish a highly accurate network model. Their results show that the proposed model operates well to predict cyber hazard incidents from tweets and greatly outperforms a variety of baselines.

- **Recurrent Neural Networks (RNNs) and Long Short-Term Memory Networks (LSTMs)**

A recurrent neural network (RNN) is a recurrent structure where a directed graph along a chain is generated by node associations. This helps the RNN to view time dynamic behaviour for a time series applied to natural language processing (NLP). RNNs can use their internal state to process input agreements and may do so only for a limited period of time, i.e. they cannot remember long-term information (Simran et al. 2019). In other words, RNN is a neural network that simulates a complex system of discrete time that has an input *xt*, an output *yt*, and a hidden state *ht*. The subscript *t* represents time in our notation. RNNs have a very elegant way of dealing with sequential (time) data that embodies connections between data points similar to the sequence (Schuster & Paliwal 1997). Elman (1990) proposed recurrent neural networks (RNNs) for sequential data processing such as voice and text processing.

The defining characteristic of RNNs, which is distinct from that of RNNs, the general neural networks is the introduction of the hidden state vector. The secret state represents the description of the previous input data which is modified once the new input is reached. Finally, after processing all input results, the secret state is the summarisation of all sequences, which is similar to the processing of a sequence done by a human being. Of course, RNN has the benefit of reading sentences that are read by a human. However as the layer deepens, gradient explosions and vanishing problems occur, which can degrade performance (Shin, Kwon & Ryu 2020). Hochreiter and Schmidhuber (1997) proposed the long-term memory (LSTM) technique to avoid this. In order to prevent the gradient from bursting and causing disappearing problems, LSTM adds the cell state to change the previous knowledge. LSTM has been commonly used for text classification because it can learn high-level representation using a deeper layer due to the cell status while maintaining the sequence of representations given by RNN. Wang, J.-H. et al. (2018) applied LSTM to the emotion classification of short texts on social media. Ding et al. (2018) suggested a densely connected Bi-LSTM composed of several Bi-LSTM layers, which shows improved efficiency than Bi-LSTM.

- **Deep Neural Networks (DNNs)**

A neural network can be a Deep Neural Network (DNN) with many layers that make it very mind-boggling. DNN contains one layer of data, at least one hidden layer, and one layer of output. A rectilinear unit (ReLU) is contained in a hidden sheet. ReLU is a mechanism for activation which has specified the positive part of its argument. It has fewer gradient problems and is efficient in terms of computation. As each neuron in a single layer is connected with each neuron in the next layer, the secret layer is also called a fully linked layer (Simran et al. 2019). A typical neural network (NN) consists of a number of neurons called simple, interconnected network processors, each producing a series of activations of real value. Sensors that sense the environment activate input neurons, and weighted connections from previously active neurons trigger other neurons (Schmidhuber 2015). Dionísio et al. (2019) presented a new tool for analysing information obtained from X using deep neural networks to process cybersecurity. The subsequent section offers a thorough comprehensive

review of cybersecurity threat detection on X through Machine Learning and Deep Learning techniques.

2.3.3 X security: ML/DL solutions

In this section, the focus is on identifying key vulnerability characteristics and conducting a comprehensive literature review of prior research studies that have utilised X data for detecting cyber threats using ML and DL techniques. After providing a brief overview of vulnerability detection and exploitation, will delve into a detailed examination of these previous studies.

A. Detection of vulnerabilities and exploits on X

Vulnerabilities and exploits are problematic security weaknesses. Vulnerabilities are typically found within software systems, while exploits arise because of these vulnerabilities. In other words, exploits are the actual manifestation of the vulnerabilities within software systems.

To better understand these weaknesses, further research is necessary to identify the root cause of these security issues and develop effective mitigation strategies (Ruohonen, Hyrynsalmi & Leppänen 2020). To prioritise the protection of systems, this section examines the marginal variance between various weaknesses concepts. The scope of this investigation is centred on utilising X data as a source of information to identify any new vulnerabilities or to assess the presence of exploits targeting known vulnerabilities. Given that the majority of security breaches are subject to temporal constraints, it is imperative to have effective mechanisms in place for detecting such incidents in a timely manner. By doing so, it becomes possible to prioritise the allocation of resources towards rectifying the vulnerability, thereby saving valuable time and effort (Campiolo et al. 2013). Researchers utilise common vulnerability and exposure identifiers (CVE-ID) as a feature to predict the likelihood of exploitation for known vulnerabilities. A novel approach was presented by Sabottke, Suciu and Dumitraș (2015) for the generation of early warnings for real-world exploits against known vulnerabilities.

The prediction is grounded in an analysis of tweets that mention these vulnerabilities, along with their associated CVE-IDs, in the context of malicious intent. To achieve this, they utilised X's Streaming API to monitor occurrences of the keyword

"CVE". Additionally, they employed the SVM algorithm, a supervised machine learning technique, to develop a classifier that leverages user and tweet-related features to identify emerging cyber-attacks. The results of this approach demonstrated superiority over the Commonly Recommended Vulnerability Scoring System (CVSS), with a reduced rate of false positives. Furthermore, the method was capable of detecting exploits with a median lead time of two days ahead of existing datasets.

Trabelsi et al. (2015) proposed a method of utilising social media analysis for software vulnerability monitoring in the HANA (SMASH) architecture. The SMASH process involves conducting a search for security and vulnerability terminologies as well as software components from sources such as Twitter and the National Vulnerability Database (NVD) and storing the information in a local database. Subsequently, tweets are grouped together through the utilisation of a modified K-mean clustering algorithm, which takes into account the context of each tweet. The NVD serves as a reference point to differentiate between old and new information regarding vulnerabilities. Currently, this process is conducted manually. The results of this research showed that 100% of the NVD weaknesses were mentioned on X, with 41% of Common Vulnerabilities and Exposures (CVEs) being published on X prior to the official NVD release, with an average of 20 days in advance. Additionally, approximately 75% of Linux-Kernel zero-day vulnerabilities were disclosed on X before the official disclosure, with an average lead time of 19 days.

Kergl, Roedler and Rodosek (2016) proposed a novel crowd-sourced vulnerability detection system that utilises X as the main source of real-time information. The system employs the use of security-specific keywords to identify tweets that pertain to potential security incidents or anomalies in online services or accounts. Subsequently, the proposed model compares these tweets with the vulnerability descriptions present in the Common Vulnerabilities and Exposures database (CVE-DB) to determine whether the detected behaviour constitutes a new vulnerability or a zero-day exploitation of a previously known vulnerability.

Queiroz, Keegan and Mtenzi (2017) utilised a corpus of tweets posted by security experts to construct a Support Vector Machine (SVM) classifier with the objective of segregating tweets that contained security alerts and software patch/fix information from general security discussions. The classifier was developed utilising

three sets of word frequency features: unigram, bigram, and a combination of both. The study found that the proposed model had an accuracy rate of 94% when classifying tweets over a one-year time period (Queiroz, Keegan & Mtenzi 2017). However, the authors noted that the methodology, which is based solely on word appearance in tweets, can result in the misclassification of tweets as false positives. This occurs when security-related words appear in both security-related and non-security-related tweet phrases, leading to the misclassification of general discussions as useful alerts, and vice versa.

Behzadan et al. (2018) introduced a cascaded Convolutional Neural Network (CNN) framework for identifying and categorizing cyberattack-related events on X. This approach involves two CNN models: a binary classifier to distinguish cyber-related from irrelevant tweets, followed by a multi-class classifier to assign specific threat labels (DDoS, zero-day vulnerabilities, ransomware, data leaks, or marketing/general) to the identified cyber tweets. The model was trained on a dataset of approximately 21,000 annotated tweets. It achieved an average F1-score of 0.82 in classifying cyber threats. Arora, Sharma and Khatri (2019) developed a Random Forest model to automatically classify cyber threats using X data, achieving an accuracy of 80%.

Le et al. (2019), the authors addressed the discrepancy between the CVE-DB and the findings of Sabottke, Suciu and Dumitraș (2015) by developing a method for identifying security-related tweets that contain information about vulnerabilities, even if the specific vulnerability ID is not mentioned. To do this, they propose a model that leverages the CVE-DB to learn the features of vulnerabilities through the use of a centroid classifier. The model is trained using descriptions of vulnerabilities as positive samples. The pipeline begins by collecting tweets from specialised security accounts and extracting TF-IDF features for each tweet. The tweets are then passed through the trained model and classified as normal or not based on their distance from the centroid and a specified threshold value.

The performance of the model was evaluated using a manually labelled dataset, yielding an F1 score of 64%, surpassing the results of SVM, MLP, and CNN baseline models.

Mahaini and Li (2021) developed Machine Learning models to categorise cybersecurity-related X accounts. They collected cybersecurity-related tweets using

X's Sampling API and manually labelled them. A baseline model was trained to identify general cybersecurity accounts, followed by sub-models for classifying accounts into individuals, hackers, or academia. Four machine learning models (Decision Tree, Random Forest, Support Vector Machine, and Logistic Regression) were compared using various account features. Random Forest achieved the highest performance, with 93% accuracy for the baseline model and 88-91% accuracy for the sub-models.

Deshmukh et al. (2022) introduced Darkintellect, a machine learning-based approach for identifying cyberattacks through X data, The researchers collected approximately 21,000 cyberattack-related tweets using the Tweepy3 Python library. To prepare the data for analysis, they employed NLP techniques to preprocess the tweets by removing irrelevant information and special characters. Feature extraction was performed using TF-IDF to represent the text data numerically. Five machine learning algorithms—SVM, RF, DT, XGBoost, and AdaBoost—were evaluated for their ability to classify tweets as cyberattacks or not. The results demonstrated that Decision Trees (DT) outperformed the other methods, achieving a classification accuracy of 87.54%.

Coyac-Torres et al. (2023) developed a hybrid NLP and CNN model to identify and categorise four cyberattack types (malware, phishing, spam, and attacks) within social network messages. This method uniquely focuses on textual analysis, making it adaptable across different platforms. Evaluated on real-world data, the model underwent a two-phase process. First, it detected the presence of any cyberattack, followed by classification into a specific category. The model achieved an overall accuracy of 82%.

The efficiency of utilising information disclosed on X to detect zero-day vulnerabilities and exploits has been established through several studies. However, it is noted that these solutions have limitations in terms of vulnerabilities, where a more comprehensive approach to detecting security content is necessary. This is because the retrieved tweets may not include specific vulnerability numbers, unlike the advanced counting-based secret-sharing security technique.

B. Detection of security content

X is a popular social media platform used by millions of users worldwide to share information. However, there is a concern about the spread of false information or

propaganda by malicious actors with security implications. To address this, researchers have proposed various methods for detecting security-related content on X, including natural language processing, machine learning algorithms, and human expert judgement. These proposals aim to provide a means for detecting malicious content and ensuring the security and reliability of information disseminated on X. In this section, we will examine a body of literature that has put forth proposals aimed at detecting security-related content that is disseminated on X.

In their study, Abdelhaq, Sengstock and Gertz (2013). introduced a novel framework for detecting localised events by analysing the presence of bursty keywords and the spatial distribution of documents. The authors emphasised the importance of incorporating both temporal and spatial aspects in event detection. The framework they presented is based on the assumption that the occurrence of bursty keywords and their spatial distribution can provide useful clues in identifying localised events. This framework is designed to identify localised events in real time and can provide valuable information to decision-makers in various domains. The authors' innovative approach to event detection has the potential to revolutionise the way analyse and understand events, providing a more comprehensive and accurate picture of what is happening on the ground.

The Cyber Twitter framework Mittal et al. (2016) is a profile-based system that utilises X API to retrieve tweets that pertain to security-related topics and system profiles. The Security Vulnerability Concept Extractor (SVCE) is then employed to extract terms that are related to security vulnerabilities, with only tweets that contain two or more terms being retained for further analysis.

These terms are tagged with technical descriptors, such as means of attack, consequences, affected software and hardware, and version numbers. The output from the SVCE is mapped to real-world concepts using DBpedia and YAGO ontologies, as well as the Unified Cybersecurity Ontology (UCO) to provide context-specific knowledge.

The resulting data is stored as triples in a local Cybersecurity Knowledge Base (KB) and a set of Semantic Web Rule Language (SWRL) (Horrocks et al. 2004) rules are added to the system to reason over the KB and generate alerts based on potential threats and vulnerabilities, as shown in Figure 5. The rules interpret the relationships

between elements in the KB and update the system's state when new tweets arrive, triggering alerts when threats are detected. These alerts are reviewed by cybersecurity experts and used to inform their security policies as necessary. In a ten-day experimental evaluation, it was found that 13 out of 15 alerts were considered useful by assessors. However, the framework is limited in its ability to detect the context of tweets, which is crucial in differentiating tweets that discuss emerging attacks from those that discuss security-related topics in general.

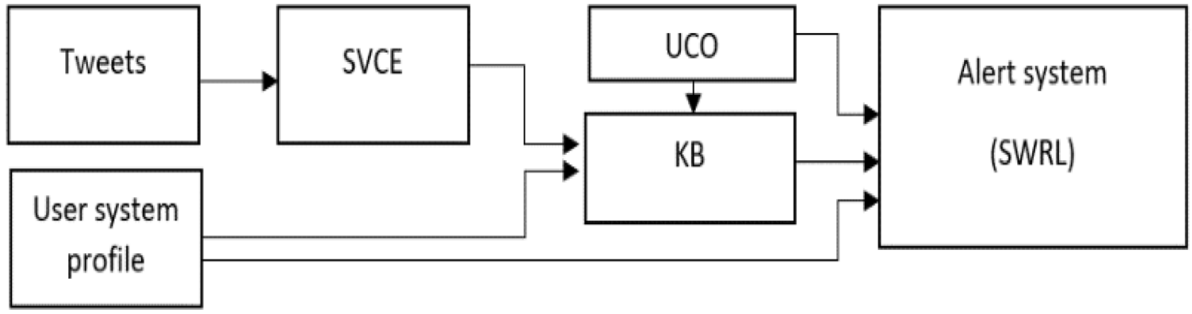


Figure 5: Cyber-Twitter architecture, Horrocks et al. (2004)

In the study by Sapienza et al. (2017), a real-time text mining approach was utilised to detect unfamiliar security terms from tweets posted by a predetermined set of 69 security experts. The process involved retrieving tweets every 60 minutes, filtering unique words and excluding words that appeared in dictionaries of English, stop-words, technical terms, and Italian terms. The remaining words were considered new security threats if they were mentioned in the same tweet with terms contained in a threat dictionary. The generated information included the new term, its volume of mentions in the past 60 minutes, the contents of the posts, and related words. The entire process from retrieving tweets to generating warnings took approximately 0.6 seconds, resulting in an accuracy rate of 84%.

One notable example was the detection of the Mirai term 49 days prior to the actual attack in October 2016. However, the solution is limited in its ability to detect new attack terms, as it relies on the knowledge of experts who may not be aware of evolving security threats until they become public, or the attack does not have an unfamiliar name prior to occurrence. Figure 6 in the study provides a visual representation of the proposed algorithm.

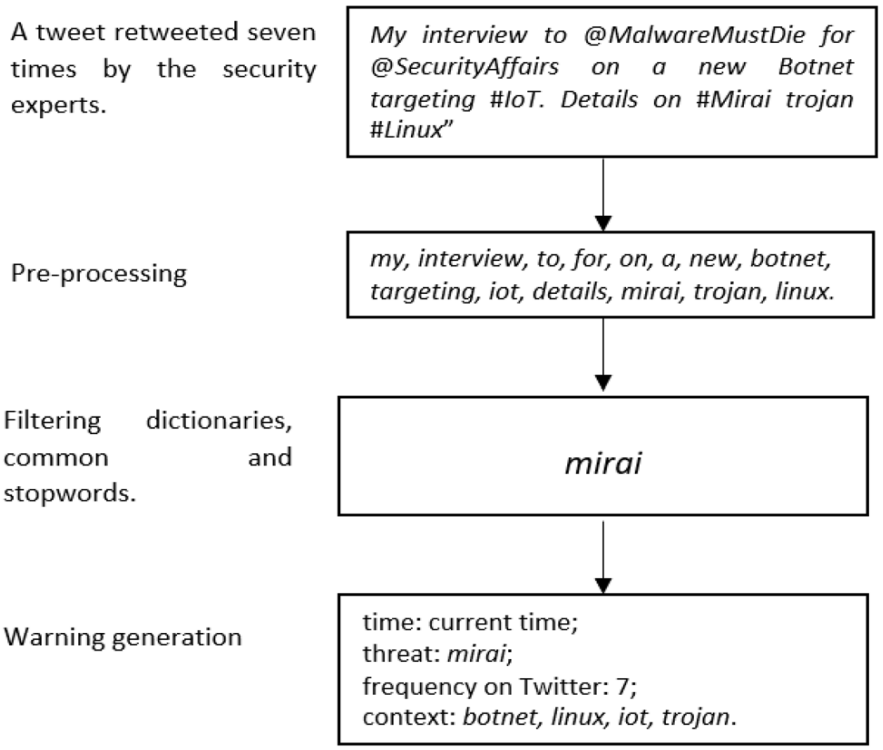


Figure 6: Running example, (Sapienza et al. 2017)

Le Sceller et al. (2017) introduced a real-time security event detection system that utilises a taxonomy of cybersecurity events and corresponding seed keywords to identify security-related tweets. Over a nine-month period, the system collected 47.8 million tweets utilising seed keywords ranging from unigrams to 6-grams. To further refine the results, a blacklist of phrases was added to reduce false positives. SONAR groups similar tweets into clusters using cosine similarity and locates the geographic area of high discussion through the use of Google Map Geocoding API. Additionally, it includes a keyword finder that continuously updates the keyword list to remain relevant. The system utilises GloVe embedding to find semantically related words, allowing for scalability while still relying on the analyst's final decision. The efficiency of SONAR was evaluated with positive results, showing that approximately 25% of the detected security events were relevant. The architecture of SONAR is presented in Figure 7.

The SYNAPSE system Alves et al. (2021) is a real-time security event detection tool for IT infrastructure. It utilises a dataset of over 195,000 tweets, retrieved from two sets of X accounts publishing security-related tweets, designated as S1 and S2. The tweets from the S1 accounts form the training set, while the validation and testing sets

are comprised of tweets from both S1 and S2 X accounts, allowing for the possibility of adding more accounts in the future. The tweets are filtered based on security keywords representing three different infrastructures and are then processed using TF-IDF features, a supervised machine learning approach using MLP and SVM algorithms, and a dynamic stream clustering methodology to group similar tweets into events. The final output of the model is presented in the form of an indicator of compromise (IoC) for manual inspection or integration with threat intelligence tools. The evaluation results showed that SVM achieved a better balance between true positive and true negative classification rates, approximately 90%, and the IoC was evaluated for relevance and modernity, demonstrating the efficiency of the end-to-end model. Figure 8 illustrates the main steps of the SYNAPSE system.

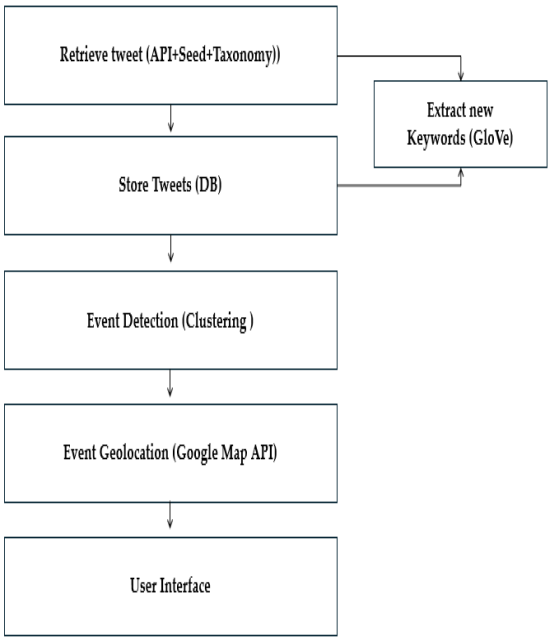


Figure 8: SONAR Architecture,
(Le Sceller et al. 2017)

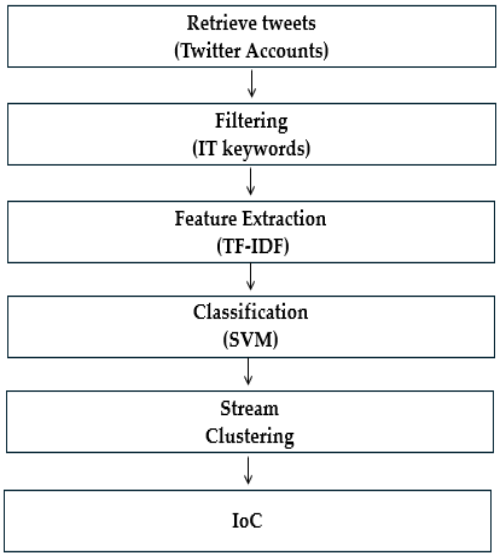


Figure 7: SYNAPSE's Architecture,
(Alves et al. 2021)

In the study conducted by Rodriguez and Okamura (2019) titled DataFreq, the authors present a novel approach for tracking the sentiment score of a particular company in relation to the probability of a potential security breach as shown in Figure 9. To gather relevant information, 70,475 tweets from a set of security expert accounts were collected and filtered based on a security keyword list. A machine learning model was developed using the logistic regression algorithm with n-gram feature extraction technique to classify the sentiment of the tweets. The authors also aimed to update the keyword list regularly to ensure that the system can adapt to new security-related

terms. Through their experiments, the model successfully identified three new words that indicated potential security breaches. The model's performance was evaluated over a four-week period, during which an increase in phishing attacks was observed during the third week (a holiday). The results indicate the effectiveness of the proposed method, with an 85% precision, 84% recall, and F1-score, reflecting the real-world scenario of an increased number of phishing attacks during holidays. The results are presented to the end-user in an easily interpretable format in real time, allowing security analysts to easily monitor the average sentiment score of the company and the most frequently mentioned security issues.

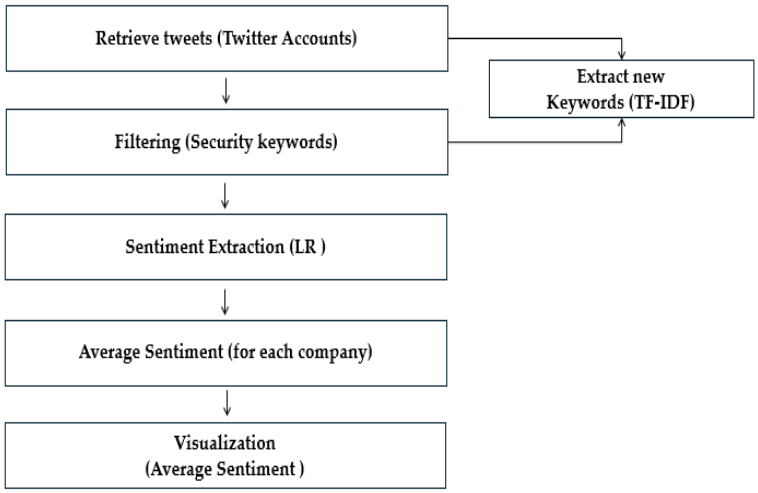


Figure 9: Data mining situational awareness scheme, (Rodriguez & Okamura 2019)

Nazir et al. (2019) combined the moving threshold average algorithm with the Gaussian tweet sentiment signal detection algorithm and the top hashtag. analysis algorithm to develop a new sentiment analysis model for X data. The proposed model was able to effectively identify the sentiment of tweets, and the hashtags associated with them. The results showed that the proposed model outperformed traditional sentiment analysis algorithms in terms of accuracy and efficiency. The study also showed that the combination of the moving threshold average algorithm and the Gaussian tweet sentiment signal detection algorithm improved the accuracy of sentiment analysis by detecting the sentiment signal more effectively. The top hashtag analysis algorithm helped to identify the hashtags associated with the sentiment and provide a more comprehensive analysis of the sentiment expressed in the tweets. This

study highlights the importance of combining multiple algorithms to improve the accuracy of sentiment analysis in social media data.

Dabiri and Heaslip (2019) presented an X-based framework for detecting traffic incidents as a supplementary method for monitoring traffic conditions. This framework involved the collection of a large volume of tweets using X API endpoints, which were then labelled through a systematic and efficient process that incorporated shortcuts for speed without sacrificing data quality. The labelled data was used to develop a deep-learning model for detecting traffic-related events from X streams. The tweets were transformed into numerical feature matrixes using word-embedding models, and CNN and RNN architectures were utilised to distinguish traffic-related tweets. The results of the experiments showed that the proposed model outperformed existing models in the field. However, to fully implement the framework, a geocoder must be developed to identify the location of traffic events and disseminate the relevant information to users in real time. This system can provide benefits to travellers by helping them choose the most efficient routes, as well as assisting traffic management agencies in restoring smooth traffic flow by detecting unexpected changes in traffic flow characteristics.

The study by Dionísio et al. (2019) investigates the real-time detection of security information relevant to IT infrastructure, as shown in Figure 10. In this work, tweets from two sets of security-related accounts are collected and processed in three-time intervals. The tweets are filtered based on keywords and transformed into numerical representations using word2vec word embedding. These embedded tweets are then input into three parallel Convolutional Neural Network (CNN) layers for classification. The output of this model is a binary classification of each tweet as security-related or not, followed by Named Entity Recognition (NER) to extract key entities such as company, asset, vulnerability, or IDs using a bidirectional long short-term memory network (BiLSTM). The extracted entities are then utilised to generate Indicator of Compromise (IoC) alerts. The classification performance was evaluated and achieved a True Positive Rate (TPR) of 94% and a True Negative Rate (TNR) of 91%, while the NER achieved an F1-score of 92% in specifying the correct labels. A comparison between the discovered IoCs and traditional security databases such as the National Vulnerability Database (NVD) revealed that the model was able to detect vulnerability information 1 to 149 days ahead of NVD.

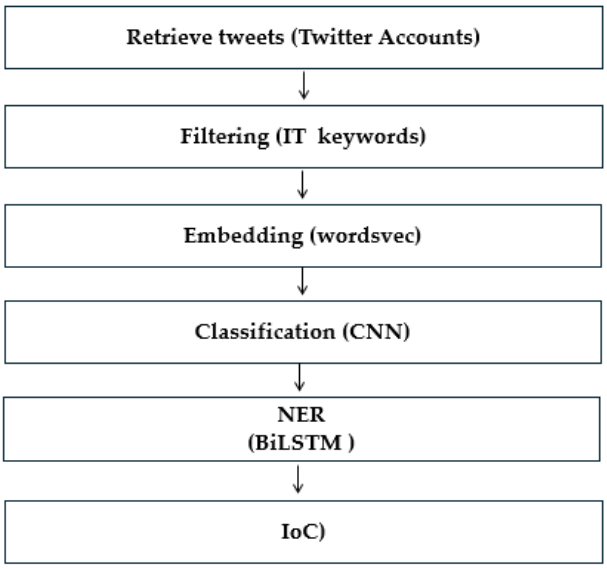


Figure 10: DeepNN BiLSTM Architecture, (Dionísio et al. 2019)

Fang et al. (2020) proposed a multi-task learning approach based on the Iterated Dilated Convolutional Neural Network (IDCNN) and Bidirectional Long Short-Term Memory (BiLSTM) natural language processing technology. The machine learning algorithm was presented to set up a highly accurate network model. Sani and Moeini (2020) used locality-sensitive hashing to roughly find related items and incremental clustering to implement a realistic, real-time event detection algorithm.

Researchers are trying to define the features of tweets and use suitable algorithms to solve the problems they are researching. Rodriguez and Okamura (2020) show a framework for classifying OSINT data into cyber-security-related to be introduced and analysed, and the accuracy of those data was subsequently improved using an unsupervised method. Table 2 summarises the comparisons of previous studies on the detection of cybersecurity threats to X. All previous studies indicate the focus on the usage of effective classifiers to improve detection accuracy.

Table 2: Comparisons of previous studies on the detection of cybersecurity threats on X.

Study	Focus	Methodology	Data Sets
Ritter et al. (2015)	Events from X that require only minimal supervision DoS attacks, data breaches, and account hijacking.	Weakly supervised learning	Tweets containing "DDoS"
(Le Sceller et al. 2017)	An automatic, self-learned framework that can detect, geolocate, and categorise cybersecurity events in near-real time over the X stream	First story detection	Streaming tweets
Rao et al. (2018)	Machine learning techniques by considering user behaviour, content of tweets, social relationships, etc., to detect different types of cyberthreats.	SocialKB	- Tweets containing "URLs" - Streaming Tweets
(Behzadan et al. 2018)	Cybersecurity events	Deep learning model; cascaded CNN architecture	Labelled 21,000 tweets collected using Tweepy
Chambers, Fry and McMasters (2018)	A novel application of NLP models to detect denial of service attacks using only social media as evidence.	Basic neural network	Tweets written on attack day
Yılmaz and Hero (2018)	Treat the event detection problem in a multimodal X hashtag network.	Expectation-maximisation (EM) algorithm	Tweets containing hashtags
Dionísio et al. (2019)	A novel tool that uses deep neural networks to process cybersecurity information received from X.	SVM, MLP, CNN, BiLSTM	Tweets filtered by keywords
Zong et al. (2019)	Analyse the severity of cybersecurity threats based on the language that is used to describe them online.	Supervised ML models	Tweets containing "ddos" and "vulnerability"
Ghankutkar et al. (2019)	Cybersecurity related data	Three supervised ML models; SVM, MNB, and RF	Real-time cyber-attack Data from HuffPost News Site
Arora, Sharma and Khatri (2019)	Cybersecurity threats relevant data	RF	Filtered tweets collected using X's streaming API
Le et al. (2019)	Collection method of Cyber threat tweets	Centroid, One-class SVM, CNN, LSTM	Streaming tweets
Dionísio et al. (2020)	A multi-task learning approach combining two Natural Language Processing tasks for cyberthreat intelligence.	Multi-Task Learning (MTL)	Streaming tweets
Shin, Kwon and Ryu (2020)	A novel word embedding model, called contrastive word embedding, that enables to maximise the difference between base embedding models.	CNN, RNN and LSTM	Curated data, OSINT data, and background knowledge.
Fang et al. (2020)	Detection of cyber threat events on tweets. Named Entity Recognition (NER) for tweets	Multitask learning NLP, IDCNN, BiLSTM	Streaming Tweets
Mahaini and Li (2021)	Cybersecurity related discussions	Four supervised machine learning models; Decision Tree, Random Forests, SVM, and Logistic Regression	Labeled tweets collected using the X Sampling API
Deshmukh et al. (2022)	Cybersecurity threats relevant data	Five ML models: SVM, random forest, decision tree, XGBoost and AdaBoost	Labeled 21,000 tweets were collected using a python package Tweepy
Coyac-Torres et al. (2023)	Cybersecurity related data	Deep learning model; CNN architecture	Social network messages

2.3.4 Ensemble learning

Ensemble learning is a powerful machine learning technique that improves model performance by combining the predictions of multiple individual models. The key principle is to leverage the diversity and strengths of these models to enhance prediction accuracy and robustness. Ensemble methods typically involve training several base models independently and then aggregating their predictions to arrive at a final output (Berahman et al. 2024).

Shukla, Jagtap and Patil (2021) developed a framework for identifying X bots using profile metadata. This study optimised the framework by comparing techniques for data preprocessing, feature selection, and model combination. The best results were achieved using Weight of Evidence encoding, Extra Trees for feature selection, and Random Forest blending, resulting in an impressive 93% AUC. While this approach offers rapid threat detection due to its reliance on static profile data, it is less effective than methods incorporating behavioural analysis.

Shahnawaz Ahmad and Mehraj Shah (2022) developed a novel unsupervised ensemble learning method to detect previously unseen attacks in IoT networks using unlabeled data. The system generates labelled data for training a deep learning model to identify IoT attacks. Additionally, it employs feature selection to optimise attack detection. The proposed model effectively recognised attacks in unlabeled IoT data, with a Deep Belief Network (DBN) achieving a 97.5% detection accuracy and a 2.3% false alarm rate when trained on the generated labelled dataset.

Khanday et al. (2022) conducted a study focused on detecting hate speech using machine learning and ensemble learning techniques during the COVID-19 pandemic. The research utilised X data, which was extracted via the platform's API with the aid of trending hashtags relevant to the pandemic. To facilitate analysis, tweets were manually annotated into two distinct categories based on various factors. Feature extraction was performed using methods such as TF-IDF, Bag of Words, and tweet length. The study identified the Decision Tree classifier as particularly effective, achieving a precision of 98%, recall of 97%, an F1-Score of 97%, and an accuracy of 97%. However, the Stochastic Gradient Boosting classifier demonstrated superior performance overall, with a precision of 99%, recall of 97%, an F1-Score of 98%, and an accuracy of 98.04%.

Ahmad et al. (2022) explored the potential of deep learning for detecting novel cyber threats—those unseen during model training. The study also examined the role of bias in identifying these unknown attacks. Traditional machine learning models, limited by single datasets, often struggle with unforeseen threats, exhibiting high accuracy in familiar scenarios but failing to recognise the unfamiliar. To address this, the research proposed a more adaptable Intrusion Detection System (IDS) using an ensemble of deep learning classifiers. Trained on multiple benchmark datasets, this ensemble aimed to detect unknown attacks without prior knowledge of specific threat patterns. By combining proven classifiers for sequential data, the research sought to create a robust IDS capable of identifying a wide range of cyber threats. The results demonstrated the effectiveness of this approach, offering promising performance and advancing practical IDS solutions.

Muneer et al. (2023) developed a novel ensemble stacking learning approach to detect cyberbullying on X. The method integrates multiple Deep Neural Networks (DNNs) and introduces a modified BERT model, BERT-M. The study employed a preprocessed Twitter dataset and utilised word2vec embeddings generated by Continuous Bag of Words (CBOW) to extract features. Convolutional and pooling layers processed these features to capture offensive language patterns. The proposed stacked model and BERT-M achieved exceptional performance, surpassing existing NLP cyberbullying detectors. With an F1-score of 0.964, precision of 0.950, and recall of 0.92, the stacked model demonstrated high accuracy in detecting cyberbullying within three minutes. The ensemble approach yielded a detection accuracy of 97.4% on the Twitter dataset and 90.97% on a combined X and Facebook dataset, emphasising its effectiveness in combating cyberbullying across platforms.

Siddiqui et al. (2023) employed an ensemble approach to accurately classify crime-related tweets. Data was collected using Tweepy and Twint libraries, and processed with TF-IDF vectorisation. The ensemble combined Logistic Regression, Support Vector Machine, K-Nearest Neighbours, Decision Tree, and Random Forest classifiers (weighted 1, 2, 1, 1, and 1 respectively) using a soft-weighted Voting classifier. This methodology achieved an impressive 96.2% accuracy on the test dataset, demonstrating the effectiveness of the ensemble for crime tweet classification.

Arora, Gupta and Yadav (2024) identified and classified spam URLs on X and developed multiple models using a combination of URL content, user profile information, and hybrid features. A large X dataset was analysed to create comprehensive feature sets for training various ensemble learning models. Our models achieved high accuracy, often exceeding 90%, particularly when using K-Nearest Neighbours within bagging and random forest ensembles. Results indicate that combining user profiles, content, and hybrid data significantly enhances spam detection accuracy.

Krishna et al. (2024) research delves into real-time public opinion by analysing tweets across a wide range of topics, including COVID-19, crime, spam, Flipkart, migraine, and airlines. The study harnessed the X API to collect a substantial dataset of tweets, which were then meticulously cleaned and preprocessed using natural language processing (NLP) techniques. To gauge public sentiment, a comparative analysis was conducted using both traditional machine learning (ML) algorithms (Naïve Bayes, Decision Trees, Random Forest, Logistic Regression) and advanced deep learning (DL) models (Recurrent Neural Networks, Long Short-Term Memory, Gated Recurrent Units). While these models were evaluated independently, the core contribution of the research lies in a novel ensemble approach that combines ML and DL models.

Alqahtani and Ilyas (2024) focused on automating the detection of binary labels in aggressive tweets, a novel system has been developed, demonstrating exceptional performance relative to previous studies conducted on the same dataset. The study employed a stacking ensemble machine learning approach, integrating four distinct feature extraction techniques to enhance performance within this framework. By combining five machine learning algorithms—Decision Trees, Random Forest, Linear Support Vector Classification, Logistic Regression, and K-Nearest Neighbours—into an ensemble model, the researchers were able to achieve significantly improved results over traditional machine learning classifiers. The stacking classifier attained an impressive accuracy rate of 94.00%, which not only surpassed the performance of conventional models but also outperformed the results of earlier experiments using the identical dataset. The findings highlighted the system's effectiveness, achieving an accuracy rate of 94.00% in correctly classifying tweets as either aggressive or non-aggressive.

Olaitan, David and Michael (2024) developed a sophisticated Deep Learning model tailored for cyberbullying detection in tweets. Leveraging the labelled twitter_parsed_dataset.csv, the model extracted keywords and entities using Maximum Entropy. A 1D-CNN architecture was then applied to classify tweets as truculent or non-truculent. The study compared four preprocessing methods (Unigram, Bigram, Trigram, and N-gram) and achieved impressive results: 96.1% accuracy, 93.6% precision, 73.7% recall, and an F1-Score of 83.8% across different evaluations.

Vaiyapuri et al. (2024) research introduces a novel cybersecurity approach, IRSO-EDLCS, to bolster cyberattack detection in Industrial Internet of Things (IIoT) environments. This technique leverages an Improved Reptile Search Optimization (IRSO) algorithm for feature selection, optimising feature relevance for enhanced detection accuracy. An ensemble of Deep Belief Network (DBN), Bidirectional Gated Recurrent Unit (BiGRU), and Autoencoder (AE) models is then employed to identify cyber threats. To further refine the model, a Modified Gray Wolf Optimiser (MGWO) is integrated for hyperparameter tuning, maximising the ensemble's performance. Rigorous simulations on a benchmark database demonstrate IRSO-EDLCS's superior performance compared to existing methods, highlighting its potential to significantly advance IIoT cybersecurity.

Table 3 provides a summary of comparisons from previous studies on detecting cybersecurity threats using ensemble learning.

Table 3: A summary of related work

Authors	Year	Focus Area	Techniques/Models	Results
Shukla, Jagtap and Patil (2021)	2021	X bot detection	Weight of Evidence encoding, Extra Trees (feature selection), Random Forest (blending)	93% AUC; rapid threat detection with static profile data but less effective than behavioral analysis methods
Shahnawaz Ahmad and Mehraj Shah (2022)	2022	IoT network attack detection	unsupervised ensemble learning, Deep Belief Network (DBN)	97.5% detection accuracy, 2.3% false alarm rate
Khanday et al. (2022)	2022	Hate speech detection during COVID-19	Decision Tree, Stochastic Gradient Boosting, TF-IDF, Bag of Words, tweet length	Stochastic Gradient Boosting: 99% precision, 97% recall, 98% F1-Score, 98.04% accuracy
Ahmad et al. (2022)	2022	Cyber threat detection	Ensemble of deep learning classifiers	Effective detection of novel cyber threats, adaptable Intrusion Detection System (IDS)
Muneer et al. (2023)	2023	Cyberbullying detection on X	Ensemble stacking, Deep Neural Networks (DNNs), BERT-M, word2vec, CBOW	97.4% accuracy on X dataset, 90.97% on combined X and Facebook dataset, F1 score: 0.964, precision: 0.950, recall: 0.92
Siddiqui et al. (2023)	2023	Crime-related tweet classification	Logistic Regression, Support Vector Machine, K-Nearest Neighbours, Decision Tree, Random Forest, TF-IDF	96.2% accuracy using soft weighted Voting classifier
Arora, Gupta and Yadav (2024)	2024	Spam URL detection on Twitter	K-Nearest Neighbours, bagging, random forest, URL content, user profile, hybrid features	High accuracy (>90%) when using combined feature sets
Krishna et al. (2024)	2024	Real-time public opinion analysis	Naïve Bayes, Decision Trees, Random Forest, Logistic Regression, RNN, LSTM, GRU, ensemble of ML and DL models	Comparative analysis of ML and DL models; novel ensemble approach combining ML and DL
Alqahtani and Ilyas (2024)	2024	Aggressive tweet detection	Stacking ensemble, Decision Trees, Random Forest, Linear SVC, Logistic Regression, K-Nearest Neighbours	94.00% accuracy in classifying tweets as aggressive or non-aggressive
Olaitan, David and Michael (2024)	2024	Cyberbullying detection in tweets	1D-CNN, Maximum Entropy, Unigram, Bigram, Trigram, N-gram	96.1% accuracy, 93.6% precision, 73.7% recall, 83.8% F1-Score
Vaiyapuri et al. (2024)	2024	Cyberattack detection in IIoT environments	IRSO algorithm, Deep Belief Network (DBN), BiGRU, Autoencoder (AE), Modified Gray Wolf Optimiser (MGWO)	Superior performance in IIoT cybersecurity, effective feature selection and hyperparameter tuning

2.4 Chapter summary

This chapter explores the cybersecurity threats prevalent on X, delving into the motivations behind these threats and highlighting the primary challenges in detecting them. In this chapter, the related literature about feature extraction methods was discussed. The Machine Learning techniques and Deep Learning methods, as well as ensemble learning were explained. Both Machine learning techniques and deep learning classifiers applied for cybersecurity threat detection were reviewed and their strength and weaknesses, the number of recognised classes, the applied databases, and measurement metrics were elaborated and compared. Then the popular and available databases in Cybersecurity threats detection in X to train and evaluate the models were discussed. This review also shows that Deep Learning offers more powerful tools than Machine Learning to analyse this vast and dynamic data stream, enabling more effective threat detection, response, and prevention. Additionally, there is a lack of comprehensive studies in the field of cybersecurity threats detection on X, which evaluates all the important factors such as prediction scope, type of cyber security threats, feature extraction technique, algorithm complexity, information summarisation level, scalability over time, and performance measurements. The current studies are not adequate in addressing all the critical issues in cybersecurity threat detection. In this research, a new enhanced machine and deep learning models for this task were developed and evaluated.

The following sections outline and explain the methodology used in this research.

CHAPTER 3 : RESEARCH METHODOLOGY

This chapter introduces and explains the applied research methodology and the proposed research design and approaches used in this thesis. It describes the research design framework and its components, including conducting a literature review, developing conceptual and theoretical frameworks, selecting databases, conducting experiments, evaluating the developed frameworks, and writing and documentation. Various research approaches are recognised as legitimate within the fields of knowledge discovery and information systems. These methods include case studies, action research, prototyping, and scientific approaches such as experimental methods. Given the focus of this research on developing robust mechanisms within the knowledge discovery system, these mechanisms or proposed theories must be validated through the classic scientific method of experimentation. This research involves diagnosing the problem and developing a solution, as identified in action research. Therefore, an integrated approach combining scientific experimentation and action research is chosen as the research method. The key features of each approach (scientific method and action research) are outlined and justified below.

3.1 Scientific approaches

Scientific approaches can be described as those that come out of a scientific tradition – that are characterised by repeatability, reductionism and refutability – inferred by objectively and rigorously studying the phenomenon under study (Lyytinen & Klein 1985; Checkland & Holwell 1998). The scientific method is popular in many areas. “The study methods of most academic areas follow the dictates of the scientific method. In many cases, only the tools of study are different. The biologist collects data by way of the microscope; the sociologist does likewise through a questionnaire. From there on the basic strategy of each is the same: to process the data, explain them, and reach a conclusion based on factual evidence” (Leedy & Ormrod 2005).

3.2 Action research approach

This research adopts the general action research (AR) methodology which is viewed as a cyclical process (Susman 1983). There are four main stages in this process: prepare, act, observe and reflect. This research methodology attempts to merge theory and practice to achieve both realistic and research goals (Susman 1983).

As one of the primary research methods, I chose action research because (1) it offers a general guide and method to conduct research activities in a rational and effective manner; (2) The power of action research is assessment and reflective learning (Susman 1983). Evaluating and analytical learning involves being able to reverse and objectively evaluate an action, decision or product based on what has been done or done, including learning to apply it to a new situation (Susman 1983). I assert that rigorous, reflective thinking and good research documentation are the means to learn well, particularly in developing a prototype system process.

3.3 Research design and approach

A research methodology outlines the approach taken to address a problem and describes the procedures for conducting the research (Leedy & Ormrod 2005). This is an operational plan derived from a research design. It offers a detailed description of the methodology used in conducting the research, including data characteristics, data collection instruments, and the data collection process. For instance, it covers the sample size and data origin, outlines the data collection instruments, and provides context for these instruments (Gable 1994). The research design guided in this research can be defined as (1) exploratory, (2) observational, (3) experimental, and (4) descriptive. The research methodology described by the research design brings together the power of the experimental (empirical) approach as well as action research to attain the research goals. It involves eight primary stages. (1) Literature review; (2) Constructing a conceptual framework; (3) Developing theoretical models; (4) data selection; (5) experimental configuration including building a prototype system and carrying out several experiments; (6) undertaking laboratory evaluation and reflection; (7) interpreting and analysing results, and thesis writing. Figure 11 presents the research design.

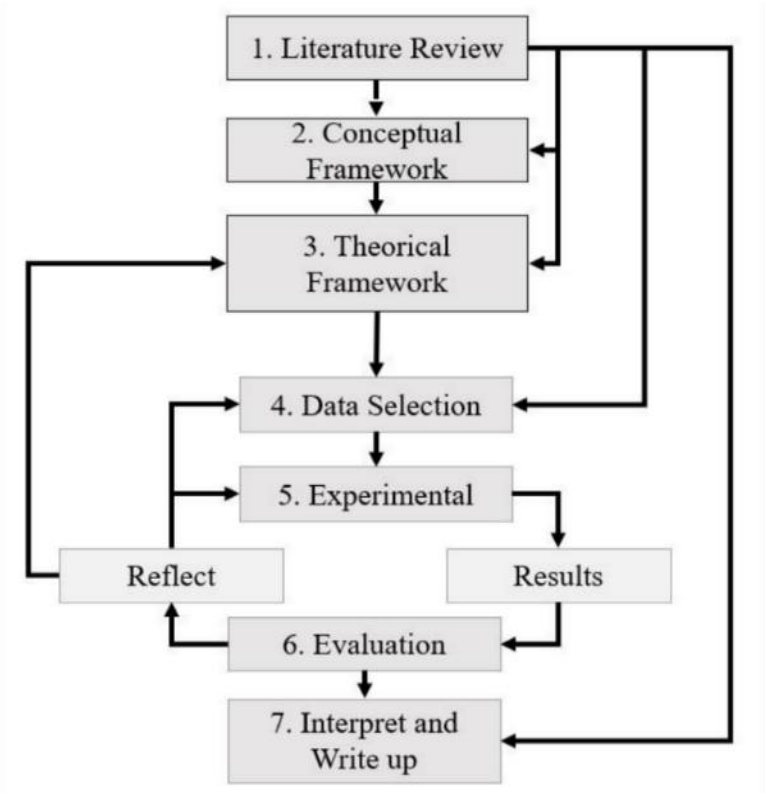


Figure 11: Research design framework developed in this doctoral thesis

3.3.1 Literature review

The literature review strategy involves the researcher exploring the literature to assess the status quo, formulating an issue or research investigation, defending the importance of following the investigation line identified, and comparing the results and ideas with his or her own (Bruce 1994). This effort involves translating the work of others in a form which shows how the exploratory process is accomplished. This phase examines possible significant problems/problems, interactions and related hypotheses found from previous research and reflects on the new fields of research. Critical research and assessment of literature e to answer research question one. Several critical related areas: Cyber Security Threats; Threat Intelligence; Social Networking Service; X; Deep Learning; and Machine Learning were discussed in the literature review.

3.3.2 Conceptual framework

After the literature review, an overall roadmap for this research can be a conceptual model for detecting cyber security threats.

After deciding on the issue to be dealt with, a preliminary literary review serves to establish a conceptual framework for developing a X-based, cyber threat detection model using machine and deep learning techniques. A preliminary cyber security threats detection framework, as the conceptual framework for this research, can be designed as shown in Figure 12.

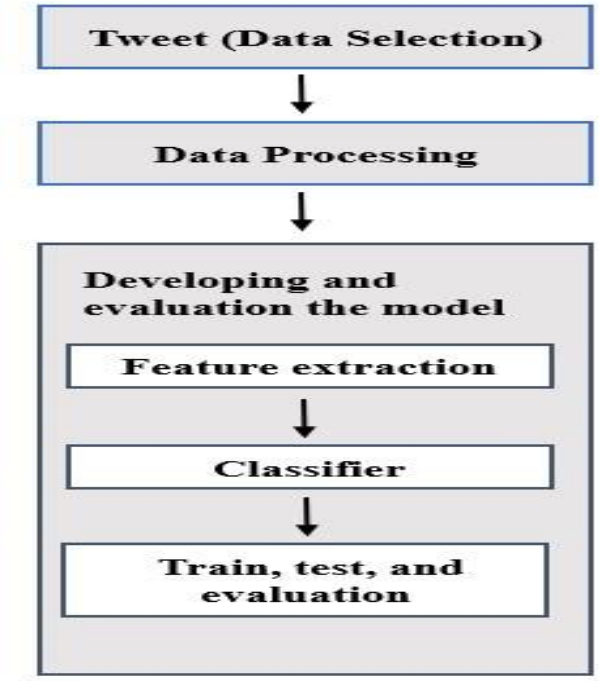


Figure 12: Proposed conceptual framework

3.3.3 Theoretical framework

This section will develop a new model based on prior research to address the research questions. The goal of this section is to develop and design a novel mechanism based on a literature review in order to address the challenges associated with detecting cyber threats on X. This model needs to be validated to answer all research questions defined in the Introduction section. As mentioned in the literature review section, recent research has found that DL techniques such as CNN, LSTM, and BiLSTM are used because of their capacity to minimise overfitting and uncover underlying cyber threat tendencies, and the capability to manage enormous datasets and the strength in short term data sequence learning (Nguyen et al. 2020; Benchaji, Douzi & El Ouahidi 2021).

3.3.4 Data selection

Datasets are essential for ML and DL. Getting data can be difficult, especially for tasks like identifying cyber threats in tweets. To assess the effectiveness of new ML, DL models for detecting cyber threats on X, a well-established dataset was chosen for training and testing. The collected Dataset consists of tweets collected from X. The Open-source Dataset, containing a collection of tweets, is available on the following link: https://www.kaggle.com/datasets/syedabbasraza/suspicious-tweets/data

To prepare the text data for analysis, a number of preprocessing steps, including removing emojis, URLs, punctuation, and stop words, converting text to lowercase, and applying stemming were implemented. These steps aimed to standardise the data, eliminate irrelevant information, and enhance the quality of the analysis by reducing noise and ensuring consistency. Details regarding the two datasets' characteristics and the specific portions used for the experiment will be covered in the following sections. The key characteristics of the dataset are illustrated in Table 4. A more detailed description of the datasets will be provided in Chapters 4 and 6.

Table 4: The key characteristics of the datasets

Datasets	**Total tweets**	**Threat tweets (label 1)**	**Non-threat tweets (label 0)**
Collected Dataset (Chapter 4)	48,783	40,838	7,945
Open-source Dataset (Chapter 5)	60,000	53,855	6,145

3.3.5 Experimental configuration and results

A prototype modelling system is developed to train, test, and evaluate the enhanced deep learning cyber threats detection model to answer research questions two and three. Modelling experiments were conducted to validate the effectiveness and efficiency of the models in Chapter 4. The developed algorithms were executed under Intel Core i7 @ 3.3 GHz and 16 GB memory computer. Python software is used for model construction and prototyping (Ketkar & Ketkar 2017), since it has freely available libraries suitable for Deep Learning such as Keras (Ketkar & Ketkar 2017), TensorFlow (Abadi et al. 2016), Scikit-learn (Pedregosa et al. 2011) and Matplotlib

(Hunter & Dale 2007). Keras allows for easy and fast prototyping and supports both convolutional networks and recurrent networks. Matplotlib is a Python 2D plotting library that is used for plotting and statistical analysis of modelling data.

This study utilises Python version 3.11.4 to build various learning models. The choice of Python stems from its popularity in cybersecurity, particularly for cyber threat analysis. Jupyter Notebook serves as the development environment. This web-based application uses JSON documents that combine code execution cells (inputs) with output cells displaying results like data, text, or visualisations. Jupyter Notebook fosters user interaction by allowing code execution in sections or as a whole program.

3.3.6 Evaluation and reflection

Evaluation of the results is important in the study to determine the effectiveness and efficiency of the models and to achieve Research Objectives 2 and 3. Validity refers to how accurately the metrics in this study's datasets capture and quantify the intended cyber threat prediction concepts. Validity is the degree to which the study's approach effectively evaluates the constructs it is intended to measure. In this study, I use ML, DL techniques to evaluate the effectiveness of the model in identifying cyber threat cases in the datasets. I use a dataset-balancing approach to obtain accurate cyber threat predictions.

In this study, several measuring systems were employed to calculate the performance of the models. In assessing ML, DL models, it is common to first train the models using a designated set of training data. Subsequently, these models are subjected to testing using a separate set of data, known as testing data, to ascertain their capacity to generalise beyond the training data. The study primarily examined the evaluation criteria related to the binary classification task of cyber threats, taking into consideration the inherent imbalance in the X datasets. Initially, to evaluate the algorithms, training and testing datasets were divided by the train-test Split. In the evaluation of ML, DL algorithms, the train-test split is a fundamental technique for assessing model performance and generalisation. This process involves dividing the available dataset into two distinct subsets: the training set and the test set. The training set is used to train the ML, DL model, allowing it to learn patterns and features from the data. After the training phase, the model's performance is evaluated using the test set, which consists of data that was not seen by the model during training. This

separation ensures that the evaluation metrics reflect the model's ability to generalise to new, unseen data rather than its proficiency on the training data alone. By using this approach can gauge how well their ML, DL algorithms perform in real-world scenarios and identify any issues such as overfitting or underfitting.

Performance metrics including confusion matrix, accuracy, recall, precision, F1-score, and PR curve are employed to assess the efficacy of different DL algorithms. A confusion matrix was employed to depict various metrics that consider the trade-off between selectivity and specificity, with the aim of minimising the time required for both Type I and Type II mistakes. The computation of the ML, DL algorithm`s output involves the utilisation of a confusion matrix to allocate input data to distinct labels. The confusion matrix is widely regarded as a straightforward and effective method for assessing outcomes. It facilitates the representation of the number of accurately classified data instances by comparing expected and observed data, utilising four different values: true positive (TP), true negative (TN), false positive (FP), and false negative (FN). True positive (TP) refers to a situation in which both the observed and expected values are positive, such as in the case of cyber threats. The scenario TN refers to a situation in which both the observed and predicted numbers indicate negativity, such as the absence of cyber threats. False Positive (FP) refers to a scenario in which the observed value is negative, but the anticipated value is positive. A false negative (FN) is the antithesis of a false positive (FP).

Accuracy, which is the reciprocal of the error rate, is a widely employed performance measure for evaluating algorithms in classification tasks. However, it may not be the most appropriate statistic for unbalanced datasets. The efficacy of the classifier in accurately identifying genuine instances of cyber threat is evidenced by the recall metric. The concept of recall pertains to the proportion of occurrences inside a given group that the model is capable of correctly identifying. There is a positive correlation between recall and the number of occurrences recovered from the minority group. The efficacy of the classifier in accurately identifying genuine instances of cyber threat is exemplified by the measure of recall. Precision is a quantitative measure that evaluates the efficacy of a hypothesis inside a hypothetical scenario where the expected outcome is positive. The F1-score is a metric that integrates both recall and accuracy, and it is commonly denoted as the harmonic mean. Assessing the

performance of the model on a minority class is beneficial. The F1-score is a widely employed metric for evaluating information retrieval systems.

The Area Under the Curve of the Receiver Operating Characteristic (AUC-ROC) is a widely employed assessment tool in situations involving unbalanced data. The training examples may be evaluated using the F1-score metric, which combines precision and recall. Additionally, a curve can be generated to illustrate the relationship between the recall and the false positive rate. As the area under the curve (AUC) value approaches, the model's performance improves. Boyd, Eng and Page (2013) provide further details on calculating and interpreting PR curves. I used precision, recall, and F1-score to measure the performance of the classifier. The specific calculation formula is shown in Table 5.

Table 5: Performance measures

Metrics	Description	Equation	Range
Accuracy (A)	Assess the number of TPs	$A = \frac{TN + TP}{TN + FN + TP + FP}$	[0-1]
Recall	The ratio of TP to a TP and FN	$R = \frac{TP}{TP + FN}$	[0-1]
Precision	The ratio of TP to a TP and FP	$P = \frac{TP}{TP + FP}$	[0-1]
F1- Score	Combines precision and recall	$FI = 2\frac{P - R}{P + R}$	[0-1]
AUC	The area between two points bounded by the function and the x-axis.	$AUC = \int_a^b f(x)\,dx$	[0-1]

3.3.7 Interpretation and write-up

The overall findings from model development, experimental data, and the literature review were analysed, interpreted, and presented.

3.4 Chapter summary

This chapter outlines the research method employed in this thesis, detailing the research design framework, including the steps taken and the development of its components to address the research questions. It introduces and clarifies the databases used, configuration setups, and evaluation metrics utilised in the proposed models. It provides a roadmap for Chapters 4, 5 and 6, which delve deeper into the specific techniques. The following chapter explores and analyses the proposed Subspace Random Ensemble Machine Learning Model (SREMLM) for cybersecurity threat detection on X.

CHAPTER 4 : THE PROPOSED SUBSPACE RANDOM ENSEMBLE MACHINE LEARNING MODEL (SREMLM)

Machine Learning (ML) is a subset of artificial intelligence (AI) and computer science dedicated to leveraging data and algorithms to mimic human learning processes, thereby enhancing accuracy over time (Btoush et al. 2023). This chapter presents a proposed Subspace Random Ensemble Machine Learning Model (SREMLM) for detecting cyber threats on X. The approach integrates K-Nearest Neighbours (KNN), Logistic Regression (LR), Decision Tree (DT), and Random Forest (RF), using a subspace random ensemble technique. Subspace random combines multiple base models to enhance predictive performance by leveraging their individual strengths and mitigating their weaknesses.

The proposed model employs a correlation-based feature selection technique to identify and extract the most relevant features for threat prediction. To improve computational efficiency, Principal Component Analysis (PCA) is applied to reduce the dataset's dimensionality. The selected features are then used to train the ensemble ML model, which combines the base models using subspace random.

The novelty of this study lies in its effective feature extraction and selection, as well as its ability to accurately classify transactions as threats or non-threats. The key contributions of this research include:

1. Developing effective algorithms for feature extraction using correlation-based techniques; K-Nearest Neighbours (KNN); Logistic Regression (LR); Decision Tree (DT), and Random Forest (RF); and Permutation Feature Importance, along with PCA for dimensionality reduction.
2. Creating a novel Subspace Random Ensemble Machine Learning Model (SREMLM) that automatically classifies tweets as threats or non-threats.
3. Integrating feature extraction and classification algorithms into a single workflow for easy implementation of X cyber threats detection.

Figure 13 illustrates the diagram of the proposed SREMLM framework.

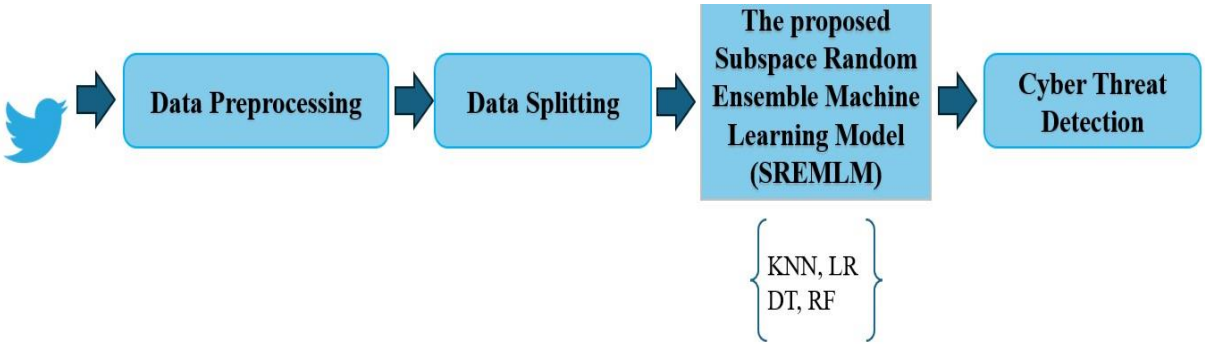


Figure 13: Proposed Subspace Random Ensemble Machine Learning Model (SREMLM)

4.1 The Collected Dataset

The dataset in this chapter was compiled by utilising the X API to collect relevant tweets. This dataset comprises 48,783 tweets, each labelled as either threats (1) or non-threats (0). The project involved preprocessing the text data, balancing the dataset, and implementing several Deep Learning models to achieve high classification accuracy.

4.1.1 Data collection method

This section provides a detailed account of the data collection process for our study on online behaviour and cybersecurity. The data, sourced from X, focuses on tweets containing keywords related to cybersecurity threats and general topics. The collected data was meticulously labelled and analysed to discern patterns in cyber threat-related and non-threat-related content.

A. **Tools and Libraries:** The primary tool used for data collection was the ***ntscraper*** library, specifically leveraging the ***Nitter*** class. This tool allows for efficient scraping of tweets without relying on X's API, which is subject to rate limits and other restrictions.

B. **Scraping Process:** The scraping process was conducted in several stages as follows:

1. **Keyword Selection**: I identified a comprehensive list of keywords relevant to our study. These keywords were chosen based on their association with cybersecurity threats and general topics. Table 6 presents the collected dataset keyword selection.

Table 6: Collected dataset keyword selection

Scammers	Bullying
Virus	Cybersecurity
Worm	Infosec
Hacker	Love
Ransomware	Security
Malware	News
Phishing	Get
DdoS	

2. **Data Extraction:** Using the ***Nitter*** class from the ***ntscraper*** library, tweets containing the specified keywords were extracted. The code snippet used for this process is as follows in table 7 .

Table 7. Data Extraction process algorithm

1	**Import** necessary libraries: ***ntscraper*** and pandas
2	Create a ***Nitter*** instance with caching disabled
3	Use Nitter to fetch tweets containing the keyword "**worm**"
4	Initialize a list to store **scraped** tweet data
5	Iterate over each tweet in the fetched tweets
6	**Extract** relevant information: URL, text, date, and comment count
7	**Append** the extracted data to the list
8	**Create** a Pandas DataFrame from the collected data
9	Save the DataFrame to a **CSV** file named "worm.csv"
10	**Print** a success message indicating completion

3. **Data Storage**: The collected data was stored in a CSV file for subsequent analysis. The file was named according to the keyword used during the scraping process to ensure traceability.

C. **Data Labelling:** After the tweets were collected, each tweet was manually annotated to classify the content as either related to cyber threats or not. This binary

classification helps in understanding the nature of the content and facilitates the development of models to detect harmful behaviour online.

D. Label Definitions:

- **Label 0**: Non-threat-related tweets: These include general discussions, informative posts, and benign content.
- **Label 1**: Cyber threat-related tweets: These comprise tweets containing mentions of cyber threats, cybersecurity incidents, and other forms of online threats.

E. Data Visualisation: To provide an intuitive understanding of the data, word clouds were generated for both classes (Label 0 and Label 1). Word clouds visually represent the frequency of words within each class, highlighting the most common terms used in non-threat-related and cyber threat-related tweets.

4.1.2 The collected dataset overview

The dataset comprises 48,783 tweets, each annotated with a label indicating whether the content is related to cyber threats or not. The dataset consists of 48,783 tweets with the following distribution, additionally, Finger 14 presents the number of each class for the Dataset:

- Total tweets: 48,783
- Threat tweets (label 1): 40,838
- Non-threat tweets (label 0): 7,945

The dataset initially included two columns: comments and labels

Figure 14: Number of each class for the collected dataset

The collected dataset was saved in a CSV file for ease of access and further analysis. Table 8 shows a sample of the collected dataset:

Table 8: Sample of the collected dataset

Comments	label
In other words #katandandre, your food was cra...	0
Why is #aussietv so white? #MKR #theblock #ImA...	0
@XochitlSuckkks a classy whore? Or more red ve...	0
@Jason_Gio meh. thanks for the heads up, b...	0
@RudhoeEnglish This is an ISIS account pretend...	0
...	...
â€œmatch my freakâ€• okay what about matching ...	1
Worms for brains	1
Been listening to this a bit more often and I ...	1
if a worm ate part of my brain and then the wo...	1
#Worms Wenn man als Halbnackter nicht in die ...	1

4.1.3 Data cleaning

To prepare the text data for analysis, I implemented a series of essential cleaning steps. These steps aimed to standardise the format and remove irrelevant information, ultimately improving the quality of our analysis.

1. Removing Emojis: I said goodbye to emojis using a custom regular expression. Emojis can add ambiguity and noise to the data, potentially skewing the results.

2. Removing URLs: I identified and removed URLs with regular expressions. They often don't contribute to the content itself and could lead to misleading analysis.
3. Removing Punctuation: All punctuation marks were removed to streamline the analysis process. Punctuation can disrupt techniques like tokenisation, so removing it helps standardise the text.
4. Converting to Lowercase: I converted all text to lowercase for uniformity. This reduces the impact of capitalisation variations, leading to more consistent analysis.
5. Removing Stop words: Using the NLTK library, we removed common English stop words like "the" and "is." These words don't carry much meaning and can be safely removed to focus on the important terms.
6. Stemming: I used the Snowball stemmer to transform words into their base forms. This helps identify and group similar words together during analysis.

These preprocessing steps were crucial in transforming the raw text data into a cleaner and more consistent format. This, in turn, improves the reliability and accuracy of our analysis. By carefully cleaning the data, I ensure it's in the best possible state for further processing and interpretation.

4.1.4 Feature scaling

To ensure that all features contribute equally to the model's learning process and to prevent any single feature from disproportionately affecting the model's performance, we applied feature scaling to the dataset using ***StandardScaler*** from scikit-learn. This scaler standardises the features by removing the mean and scaling them to unit variance. Feature scaling is a crucial preprocessing step in many Machine Learning algorithms. Without it, features with larger scales can dominate the learning process, leading to biased or suboptimal models. By standardising the features, we ensure that each feature is on the same scale, allowing the model to learn more effectively from the data. ***StandardScaler*** works by computing the mean and standard deviation for each feature during the fitting process and then transforming the data to have a mean of zero and a standard deviation of one. This transformation is particularly important for algorithms that rely on distance measurements, such as k-nearest neighbours and, Logistic Regression (LR), Decision Tree (DT), and Random Forest (RF)., as it ensures that all features are treated equally in terms of their impact on the

model. Additionally, standardising the data can improve the convergence speed of gradient descent-based algorithms. When features are on a similar scale, the optimisation process can proceed more smoothly, potentially leading to faster and more stable convergence.

In summary, by applying feature scaling with ***StandardScaler***, we enhance the learning process of our machine learning model, ensuring that no single feature disproportionately influences the model's performance. This step is essential for creating a robust and accurate model that can generalise well to new data.

4.1.5 Data features

Feature extraction is a crucial preprocessing step in machine learning, playing a fundamental role in transforming raw, high-dimensional data into a set of meaningful features that can significantly enhance model performance, efficiency, and interpretability. By selecting and transforming only the most relevant data points, feature extraction helps reduce noise, mitigate the risk of overfitting, and decrease computational costs. Common techniques for this process include Principal Component Analysis (PCA), which performs dimensionality reduction by identifying patterns in the data and projecting them onto a smaller, more manageable set of components. Through this transformation, PCA not only helps simplify the data but also retains the underlying structure, enabling the model to focus on the most important information. In doing so, feature extraction ensures that the machine learning model can achieve higher accuracy, better generalization to unseen data, and improved performance overall.

In the context of study, PCA applied to a dataset of tweets in order to extract robust, discriminative features that would aid in detecting threat-related tweets. Threat detection in social media platforms can be particularly challenging due to the large volume and high dimensionality of textual data. In our case, after addressing the class imbalance issues (such as the disproportionate number of threats vs. non-threat tweets), the dataset still retained a high number of features. This posed a challenge for effectively training machine learning models, as a large number of features can lead to slower training times and increased risk of overfitting.

To resolve this, Principal Component Analysis (PCA) was employed to reduce the dimensionality of the dataset while retaining as much of the original variance as possible. This transformation allowed us to retain 95% of the total variance in the data, ensuring that the most informative aspects of the dataset were preserved. The formula for PCA can be expressed as:

$$Z + XW \tag{1}$$

where:

Z is the transformed feature space. X is the original data matrix. W is the matrix of eigenvectors (principal components).

By applying PCA, we were able to reduce the dimensionality of the dataset from 5,000 features to 3,437 features, making it more manageable and efficient for training machine learning models. This dimensionality reduction not only improved the computational efficiency of the training process but also helped in focusing the model on the most relevant patterns, leading to a more effective threat detection system.

4.1.6 Generating word clouds

Word clouds are a valuable visual tool for identifying the most common words in text data. This can be particularly insightful when analysing tweets categorised as threats or non-threats, as shown in Figure 15.

Figure 15: Word Clouds for the collected dataset

4.1.7 Dataset balancing

Given the class imbalance, SMOTE (Synthetic Minority Over-sampling Technique) was applied to balance the datasets. SMOTE generates synthetic samples to match the minority class to the majority class. Post-SMOTE, the dataset contained an equal number of threats and non- threat tweets, the collected dataset increased to above 35,000 for each class, and the second dataset increased to above 50,000 for each class, as outlined in Figure 16.

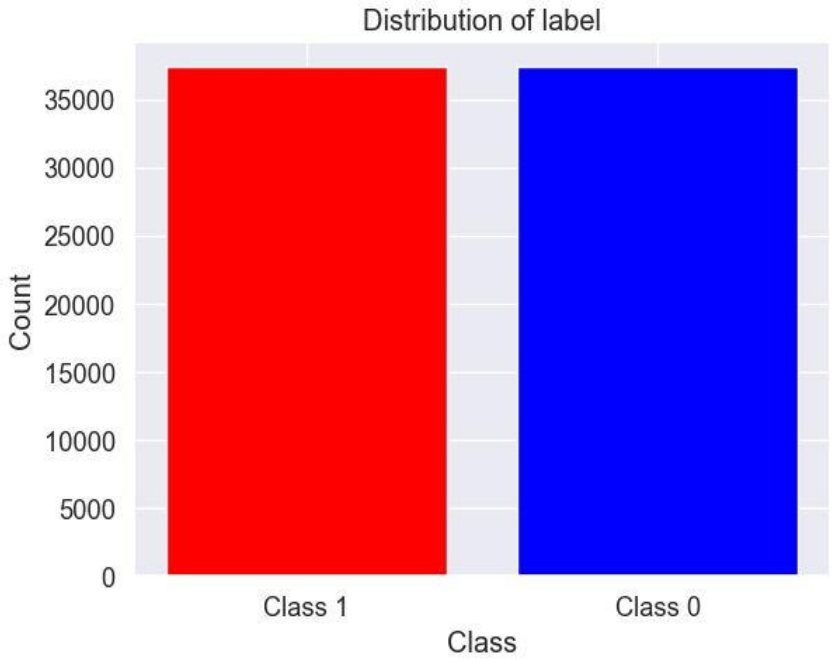


Figure 16: Data Balancing for collected dataset

4.1.8 Data preprocessing

Data splitting is a crucial step in deep learning and data science workflows, as it ensures that models are trained and evaluated on different subsets of data. This process helps in assessing the model's performance accurately and ensures it generalises well to unseen data (Joseph & Vakayil 2022).

The section below provides a detailed explanation of the data splitting technique used in this study, covering the stages of vectorisation, oversampling, tokenisation, padding, and the actual splitting of the dataset into training and testing sets as detailed

below. Text data, being inherently non-numeric, must be converted into a numerical format suitable for machine learning algorithms.

To achieve this, the ***TfidfVectoriser*** was employed. This tool utilises the Term Frequency-Inverse Document Frequency (TF-IDF) method, which assigns weights to words based on their frequency within a document and their overall importance across the entire dataset, ensuring a balanced representation of the text data.

A. TF-IDF

$$tf - idf(t,d) = tf(t,d) \times \log\left(\frac{N}{\{d \in D : t \in d\}}\right) \quad (2)$$

where:

- t is a term.
- d is a document.
- N is the total number of documents.
- tf is term frequency.
- idf is inverse document frequency.

In this case, the ***TfidfVectoriser*** was configured to extract a maximum of 5,000 features, which reduced the text data's dimensionality while retaining the most significant information.

B. Oversampling with SMOTE:

SMOTE (Synthetic Minority Over-sampling Technique) was applied to handle class imbalance. SMOTE generates synthetic samples for the minority class:

$$x_{new} = x_i + \lambda \times (x_i - x_j) \quad (3)$$

where x_i *and* x_j are samples from the minority class, and λ is a random number between 0 and 1.

- ***Purpose:*** To address class imbalance in the dataset by generating synthetic samples for the minority class.

- ***Method:*** SMOTE (Synthetic Minority Over-sampling Technique) creates synthetic data points by interpolating between existing minority class samples.

- ***Outcome:*** The dataset becomes balanced, which helps in training a model that does not create bias towards the majority class.

C. Conversion to array of strings:

- ***Purpose:*** To facilitate the subsequent tokenisation process.
- ***Method:*** Convert the resampled sparse matrix back into an array of strings, maintaining the structure and content of the text data.
- ***Outcome:*** The text data is prepared in a format suitable for tokenisation.

D. Tokenisation and padding:

The text data was tokenised and padded to ensure uniform input length for the models. Padding sequences ensures that all sequences in a batch have the same length:

$$Padded\ sequence = [x_1, x_2, \dots, x_n] \quad (4)$$

- ***Purpose:*** To convert text into sequences of integers and ensure uniform length for input into neural networks.
- ***Method:***
  - *Tokenisation*: Each text is converted into a sequence of integers where each integer represents a unique word in the corpus.
  - *Padding*: Sequences are padded to ensure they all have the same length, which is required for batch processing in neural networks.
- ***Outcome***: The text data is converted into a format that neural networks can process, with sequences of equal length.

E. Splitting the Dataset:

- ***Purpose***: To create separate training and testing datasets to evaluate the model's performance.
- ***Method***: Using a function like ***train_test_split,*** the data is randomly divided into training and testing sets, typically with an 80-20 split.

- ***Outcome***: The data is split into two subsets: one for training the model and one for evaluating its performance on unseen data.

The data splitting technique described involves multiple preprocessing steps to transform and balance the dataset before dividing it into training and testing subsets. This method ensures the model is trained on a balanced dataset and evaluated on a separate test set, providing a robust measure of its performance and generalizability. The combination of vectorisation, SMOTE, tokenisation, padding, and splitting is a comprehensive approach to preparing text data for Deep Learning tasks.

All models are compiled using the Adam optimiser and binary cross-entropy loss function, with accuracy as the primary evaluation metric. The models are trained using early stopping and learning rate reduction on a plateau to avoid overfitting and to optimise training efficiency.

Compilation:

- Models are compiled with the binary cross-entropy loss function, suitable for binary classification tasks.

- The Adam optimiser is used for training, which adjusts learning rates dynamically.

Callbacks:

- Early stopping is implemented to halt training when the validation loss stops improving, preventing overfitting.

- ReduceLROnPlateau is used to reduce the learning rate when the validation loss plateaus, allowing the model to converge more effectively.

Additionally, duplicate tweets were identified and removed, reducing the collected Dataset to 45,418 unique entries. The dataset preprocessing steps are shown in Figure 17.

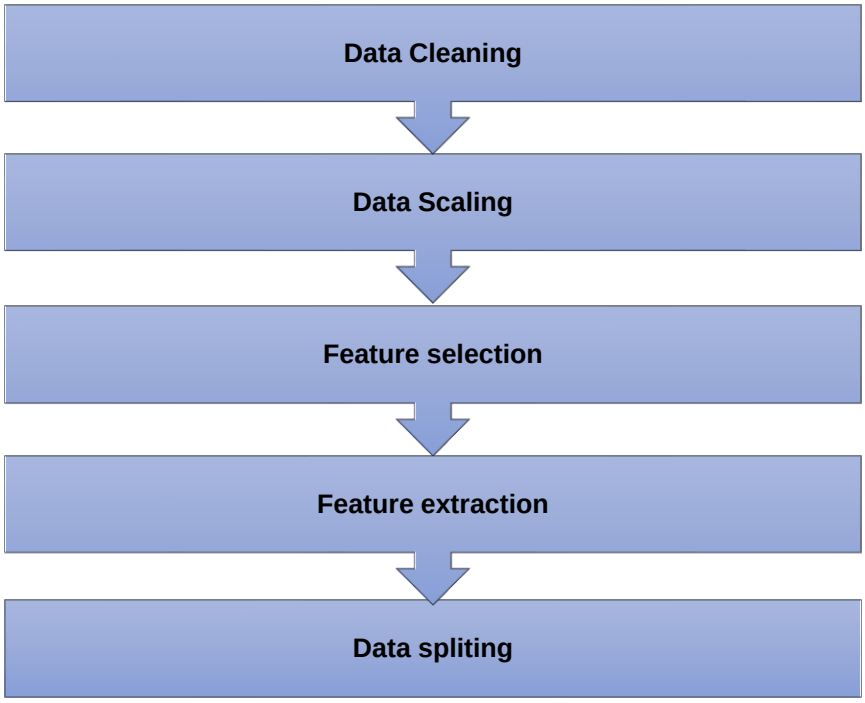


Figure 17: The datasets' preprocessing steps

4.2 Machine Learning techniques

This section provides an overview of individual Machine Learning (ML) models. The models discussed include KNN, LR, DT, and RF, all of which are applied to the dataset described in Section 4.1. The following subsections explore the design, implementation, and performance of each model, emphasising their contributions to enhancing prediction accuracy and performance. By combining the strengths of different DL architectures, the ensemble model is designed to improve overall performance.

4.2.1 K-Nearest Neighbours (KNN)

K-Nearest Neighbours (KNN) is a straightforward machine learning algorithm used for classification and regression by relying on the proximity of data points. It works by identifying the 'K' closest training examples to a new data point based on a chosen distance metric, such as Euclidean distance. For classification, KNN assigns the most common class among these neighbours, while for regression, it predicts the average value of the neighbours. Though easy to implement and understand, KNN can be computationally intensive with large datasets and is sensitive to feature scaling and

the choice of K, which can significantly impact its performance. Algorithm 4.2.1, Figure 18 details the model's performance evaluation process.

Table 9. Algorithm 4.2.1: K-Nearest Neighbours (KNN)

```
1   Procedure KNN_Model_Evaluation(data, n_neighbors, cv, scoring)
2       Import KNeighborsClassifier from sklearn.neighbors
3       Define K-NN model with n_neighbors
4       Initialize K-NN model:
5           K-NN model = KNeighborsClassifier(n_neighbors=n_neighbors)
6       Evaluate using cross-validation:)
7           cv_scores_knn = cross_val_score(K-NN model, data.X, data.y,
            cv=cv, scoring=scoring)
8       Print K-NN cross-validation accuracy:
9           print(f"K-NN Cross-Validation Accuracy:
            {cv_scores_knn.mean():.4f} ± {cv_scores_knn.std():.4f}")
10  End Procedure
```

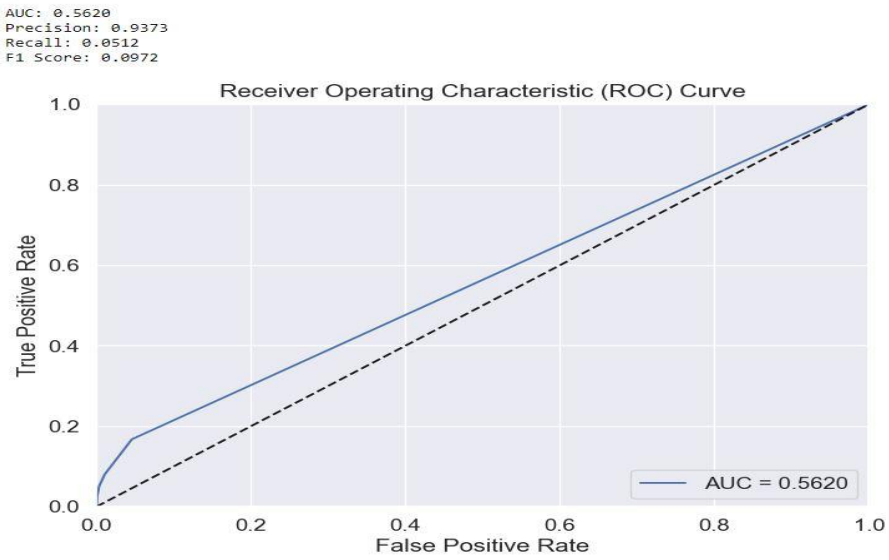


Figure 18: ROC curve for K-Nearest Neighbours (KNN)

4.2.2 Logistic Regression (LR)

Logistic regression is a machine learning algorithm designed to predict the probability of an event occurring. It employs a sigmoid function to map a linear combination of input variables to a value between 0 and 1, representing the likelihood of the event. By setting a threshold, the model can be used to classify instances into binary categories. Logistic regression is often trained using optimisation techniques to minimise the log-loss function, which measures the discrepancy between predicted probabilities and actual outcomes. Its simplicity and interpretability make it a popular choice for various classification tasks. Algorithm 4.2.2 and Figure 19 outline the process for evaluating the model's performance.

Table 10. Algorithm 4.2.2: Logistic regression (LR)

```
1   Procedure Logistic_Regression_Model_Evaluation(X_pca, y_resampled,
    cv, scoring)
2       Import LogisticRegression from sklearn.linear_model
3       Define Logistic Regression model
4       Initialize Logistic Regression model:
5           lr_model = LogisticRegression(random_state=42)
6       Evaluate using cross-validation:
7           cv_scores_lr = cross_val_score(lr_model, X_pca, y_resampled,
            cv=cv, scoring=scoring)
8       Print Logistic Regression cross-validation accuracy:
9           print(f"Logistic Regression Cross-Validation Accuracy:
            {cv_scores_lr.mean():.4f} ± {cv_scores_lr.std():.4f}")
10  End Procedure
```

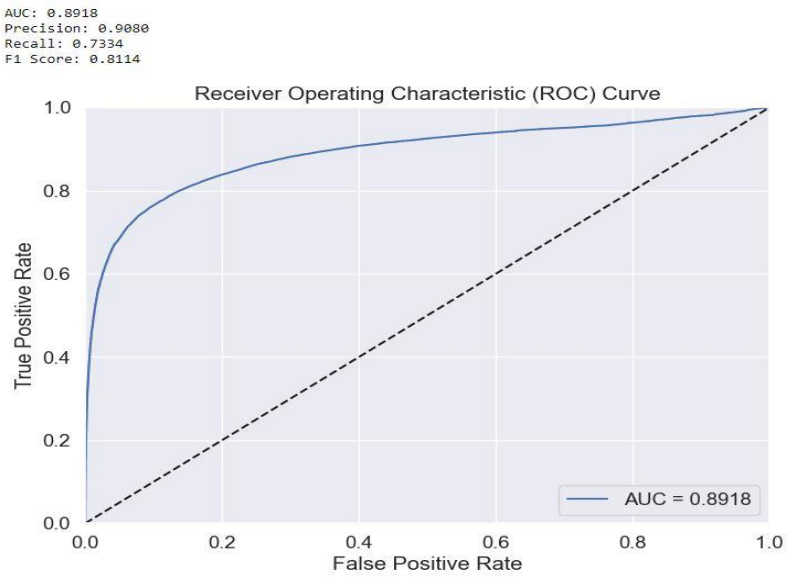


Figure 19: ROC Curve for Logistic Regression (LR)

4.2.3 Decision Tree (DT)

A Decision Tree (DT) is a Machine Learning algorithm that constructs a tree-like model to make predictions. It recursively partitions data based on feature values, with internal nodes representing decision points and leaf nodes indicating final outcomes. DTs are known for their interpretability and lack of feature scaling requirements. However, they can suffer from overfitting and instability. Techniques like pruning and ensemble methods (e.g., Random Forests) can mitigate these issues and enhance performance. Algorithm 4.2.3 and Figure 20 illustrate the steps involved in assessing the model's performance.

Table 11. Algorithm 4.2.3: Decision Tree (DT)

1	**Procedure** Decision_Tree_Model_Evaluation(data, cv, scoring)
2	**Import** DecisionTreeClassifier **from** sklearn.tree
3	**Define** Decision Tree model
4	**Initialize** Decision Tree model:
5	Decision Tree model = DecisionTreeClassifier(random_state=42)
6	Evaluate **using** cross-validation:
7	cv_scores_dt = cross_val_score(Decision Tree model, data.X, data.y, cv=cv, scoring=scoring)

```
8        Print Decision Tree cross-validation accuracy:
9            print(f"Decision Tree Cross-Validation Accuracy:
             {cv_scores_dt.mean():.4f} ± {cv_scores_dt.std():.4f}")
10   End Procedure
```

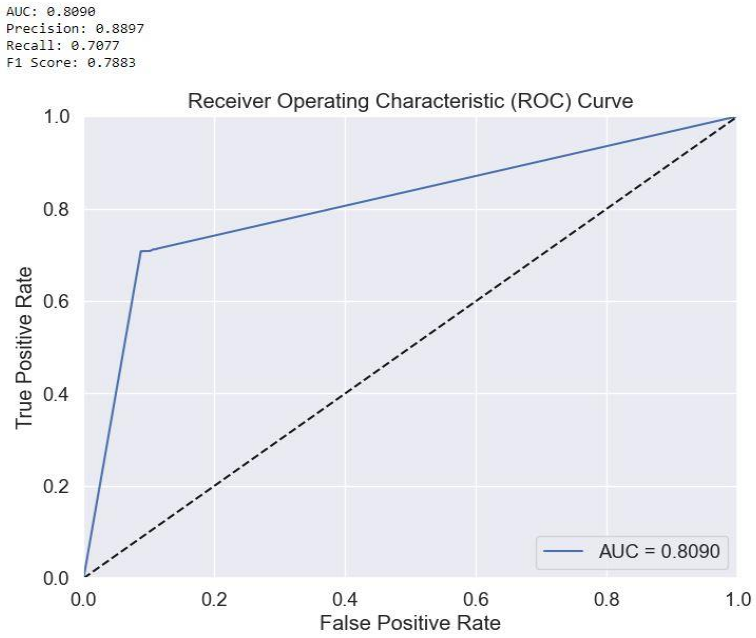


Figure 20: ROC for Decision Tree (DT)

4.2.4 Random Forest (RF)

Random Forest is an ensemble learning technique that leverages the power of multiple decision trees to enhance prediction accuracy. By employing a process known as bagging, each decision tree is trained on a randomly selected subset of the data, using a random subset of features at each node. This approach helps to mitigate overfitting, as the trees are less likely to become overly specialised to the training data. For classification tasks, Random Forest combines the predictions of individual trees through a majority voting mechanism. The class with the most votes among the trees is selected as the final prediction. In regression problems, the predictions of the trees are averaged to obtain the final output. This ensemble approach offers several advantages, including improved model robustness and the ability to handle large and diverse datasets effectively. However, Random Forests can be computationally

intensive to train, especially for large numbers of trees. Additionally, they may be less interpretable than simpler models, as the combined predictions of many trees can make it difficult to understand the underlying decision-making process. Algorithm 4.2.4 and Figure 21 demonstrate the model's performance evaluation.

Table 12. Algorithm 4.2.4: Random Forest (RF)

```
1   Procedure Random_Forest_Model_Evaluation(data, cv, scoring)
2       Import RandomForestClassifier from sklearn.ensemble
3       Define Random Forest model:
4           rf_model = RandomForestClassifier(random_state=42)
5       Evaluate using cross-validation:
6           cv_scores_rf = cross_val_score(rf_model, data.X, data.y, cv=cv,
            scoring=scoring)
7       Print Random Forest cross-validation accuracy:
8           print(f"Random Forest Cross-Validation Accuracy:
            {cv_scores_rf.mean():.4f} ± {cv_scores_rf.std():.4f}")
9   End Procedure
```

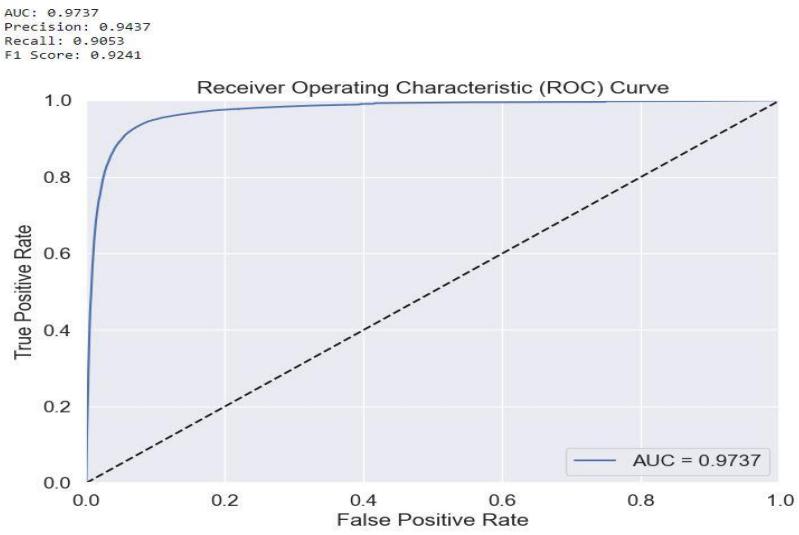


Figure 21: ROC for Random Forest (RF)

4.3 The Proposed Subspace Random Ensemble Machine Learning Model (SREMLM)

Ensemble learning combines multiple models to improve predictive performance by leveraging the strengths of each individual model. In this section, I discuss four ensemble methods: Stacking, Voting, Subspace Random Ensemble, and Bagging. Subspace Random consistently outperformed Stacking, Voting, and Bagging in terms of accuracy, precision, recall, and F1 score. This suggests that Subspace Random is particularly adept at balancing the trade-off between precision and recall, leading to its superior overall performance. While the other methods also demonstrated solid results, Subspace Random's metrics were consistently higher across all evaluation criteria.

A) Stacking ensemble

Stacking is a meta-learning technique where base models (or level-0 models) are trained on the dataset, and their predictions are used as inputs to a higher-level model (meta-model or level-1 model) which makes the final prediction. The meta-model is trained on the predictions of the base models.

$$\hat{y} = M_{meta}(M_1(X), M_2(X), \dots, M_k(X)) \quad (5)$$

Implementation: In this implementation, we used K-Nearest Neighbours (KNN), Logistic Regression (LR), Decision Tree (DT), and Random Forest (RF) as the base models, and Logistic Regression as the meta-model.

Evaluation*:* The stacking model was evaluated using cross-validation. Accuracy 0.84.

The AUC (Area Under the ROC Curve) was calculated as 0.8782.

Results*:*

- Precision: 0.84 (weighted avg)
- Recall: 0.84 (weighted avg)
- F1-Score: 0.83 (weighted avg)
- AUC: 0.8782

The confusion matrix and ROC curve: for the stacking model were plotted, showing a balanced performance across both classes.

B) Voting Ensemble

Voting ensemble is a technique where multiple models vote on the final prediction. In soft voting, the class probabilities predicted by each model are averaged, and the class with the highest average probability is chosen as the final prediction.

$$\hat{y} = \arg_{\max} \left[\frac{1}{k}\sum_{i+1}^{k} P_i(y|X)\right] \quad (6)$$

Implementation: The same base models as in the stacking ensemble were used, and the voting method was set to 'soft'.

Evaluation: Accuracy 0.83, The voting model achieved an AUC of 0.8953.

Results:

- Precision: 0.84 (weighted avg)
- Recall: 0.83 (weighted avg)
- F1-Score: 0.83 (weighted avg)
- AUC: 0.8953

Confusion Matrix and ROC Curve: The results were visualised with a confusion matrix and ROC curve, indicating a slightly improved performance compared to the stacking model.

C) Subspace Random Ensemble

Subspace random ensemble is a variant of bagging where each base model is trained on a random subset of the features. This method can help reduce the correlation between base models, potentially improving ensemble performance.

$$\hat{y} = \frac{1}{k}\sum_{j=1}^{k} \mu_j \left[F_j[x]\right] \quad (7)$$

Implementation: A logistic regression model was used as the base model, with each model trained on 50% of the features.

Evaluation: Accuracy 0.9634, The subspace random ensemble achieved an AUC of 0.9634, the highest among the ensemble methods discussed.

Results:

Precision: 0.90 (weighted avg)

Recall: 0.90 (weighted avg)

F1-Score: 0.90 (weighted avg)

AUC: 0.9634

Confusion Matrix and ROC Curve: The confusion matrix and ROC curve indicate a strong performance with a high true positive rate.

D) Bagging Ensemble

Bagging (Bootstrap Aggregating) is an ensemble technique where multiple versions of a model are trained on different bootstrap samples of the data, and their predictions are averaged to form the final prediction.

$$\hat{y} = \frac{1}{n}\sum_{i=1}^{n} m_i(x) \quad (8)$$

Implementation: A logistic regression model was used as the base model for the bagging ensemble.

Evaluation: Accuracy 0.83, The bagging ensemble achieved an AUC of 0.8953.

Results:

Precision: 0.84 (weighted avg)

Recall: 0.84 (weighted avg)

F1-Score: 0.83 (weighted avg)

AUC: 0.8953

Confusion Matrix and ROC Curve: The results were consistent with the performance observed in the voting ensemble, with a strong overall performance.

The results illustrate that the subspace random technique is highly effective as an ensemble learning method for detecting cyber threats. This approach significantly enhances the robustness and reliability of predictive models in practical scenarios.

Table 13 presents the outcomes of applying the ensemble methods, while Figure 22 displays the F1 score.

Table 13: Algorithms performance on different ensemble techniques

ML technique	Accuracy	Precision	Recall	F1 score
Stacking ensemble	0.8782	0.84	0.84	0.83
Voting ensemble	0.8953	0.84	0.83	0.83
Subspace Random ensemble	0.9634	0.90	0.90	**0.90**
Bagging ensemble	0.8962	0.84	0.83	0.83

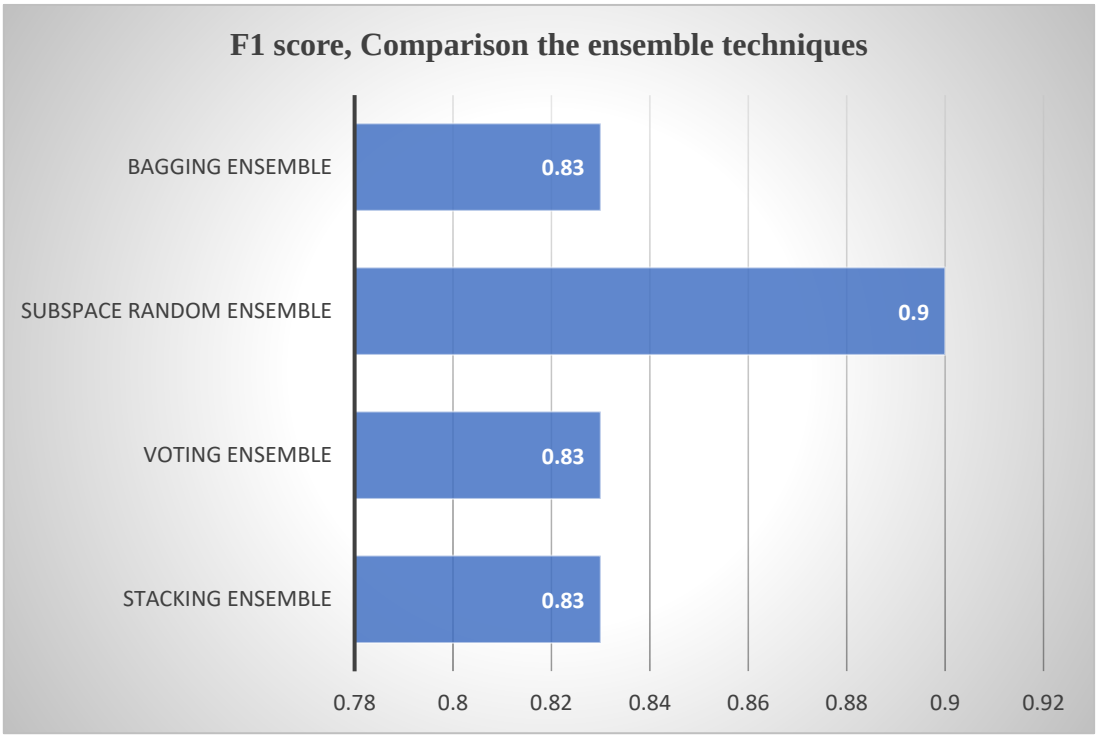


Figure 22: F1 score, Comparison the ensemble techniques.

4.4 Experimental results and discussion

Ensemble methods have become a cornerstone in machine learning, offering substantial improvements over individual models. This section delves into the impact of various ensemble techniques— a Subspace Random, bagging, voting, and

stacking—on accuracy, FI scour, and efficiency across different machine learning algorithms. Starting with a baseline evaluation of individual models, we assess their predictive capabilities. Subsequently, we apply ensemble methods to these models, aiming to amplify their strengths and mitigate their weaknesses. By comparing the results, we gain valuable insights into the effectiveness of ensemble techniques in diverse contexts and datasets. Key performance indicators, including accuracy, precision, recall, and computational efficiency, will be analysed. We will highlight significant improvements or potential drawbacks, providing a comprehensive understanding of how ensemble methods can elevate the performance of machine learning systems.

4.4.1 Comparison between ML techniques before the ensemble approach

This section provides a detailed comparative analysis of individual machine learning techniques, specifically K-Nearest Neighbours (KNN), Logistic Regression (LR), Decision Tree (DT), and Random Forest (RF) before applying the ensemble Technique. The analysis begins by evaluating the performance of each algorithm separately, without the application of ensemble methods. The results of this evaluation are thoroughly presented in Table 14, which offers a comparative overview of the algorithms' performances.

Figure 23 represents the performance metrics of the machine learning algorithms when ensemble techniques are not utilised. Additionally, Figures 24 through 27 present the confusion matrices for each of the machine learning techniques, providing a visual depiction of their classification performance. These confusion matrices offer insights into the true positive, false positive, true negative, and false negative rates for each algorithm, thereby allowing a deeper understanding of their effectiveness in different scenarios.

Table 14: ML algorithms performance before applying ensemble techniques.

ML technique	Accuracy	Precision	Recall	F1 score	AUC
KNN	0.5620	0.9373	0.0512	0.0972	0.5620
LR	0.8918	0.9080	0.7334	0.8114	0.8918
DT	0.8090	0.8897	0.7077	0.7883	0.8090
RF	0.9995	0.9743	0.7755	0.8636	0.9837

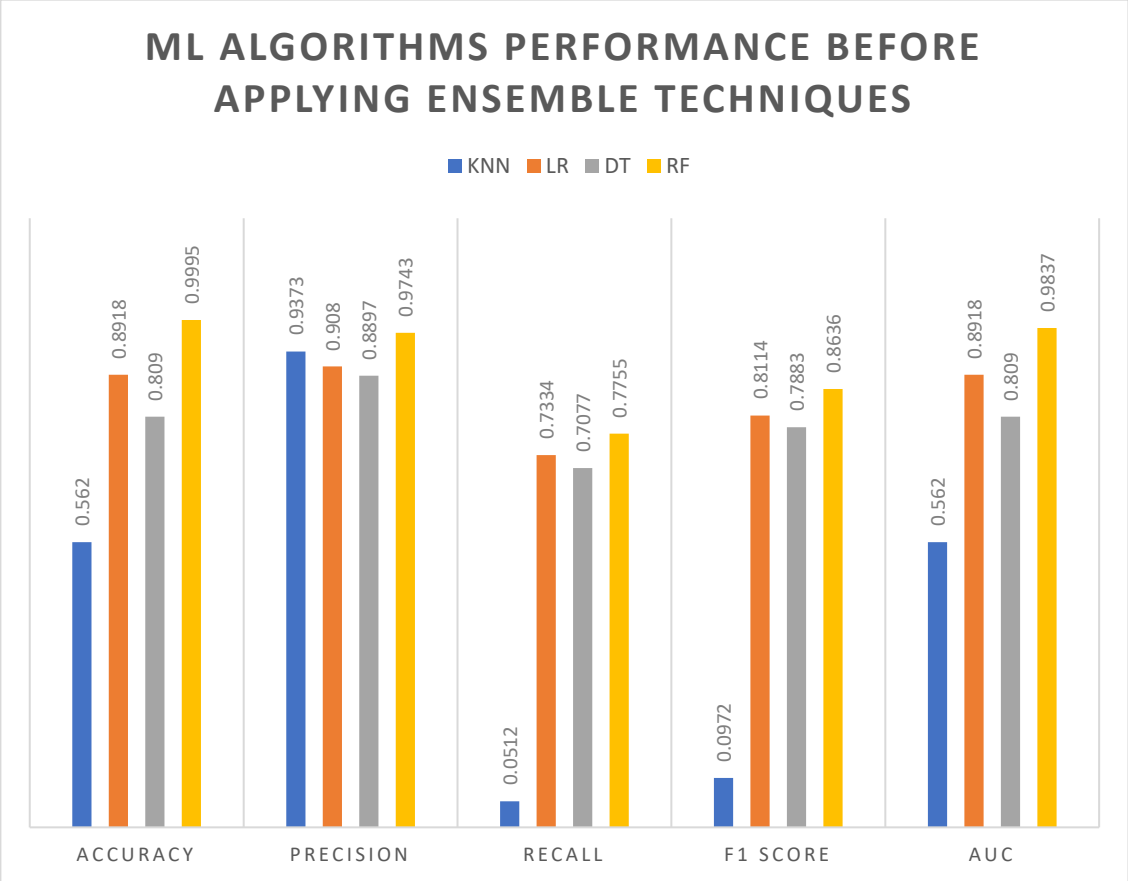
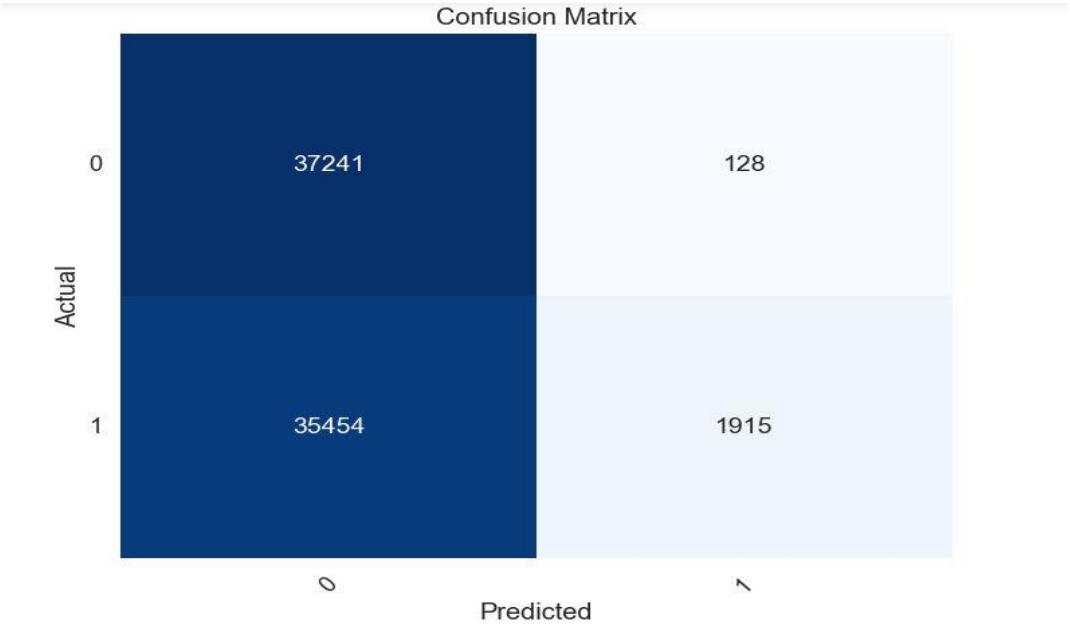


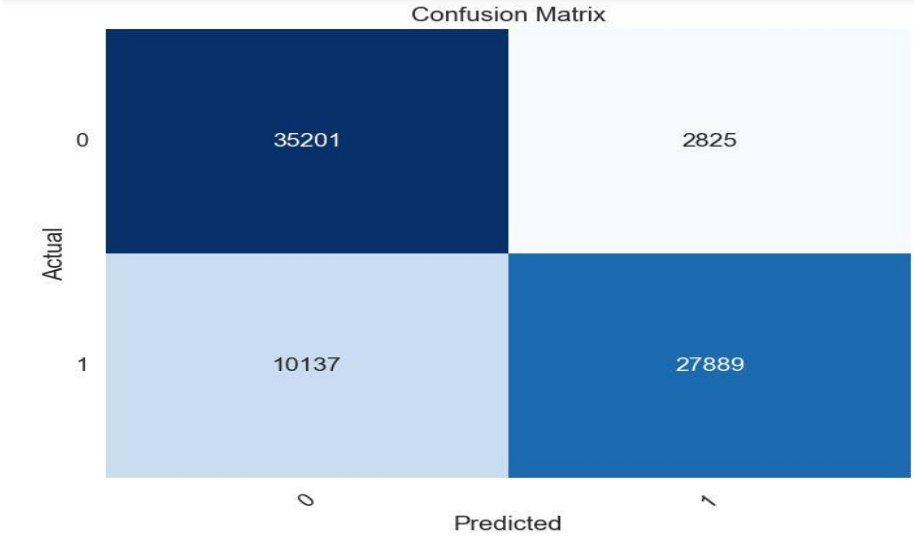
Figure 23: ML Algorithms performance before applying ensemble techniques.



Classification Report:

	precision	recall	f1-score	support
0	0.51	1.00	0.68	37369
1	0.94	0.05	0.10	37369
accuracy			0.52	74738
macro avg	0.72	0.52	0.39	74738
weighted avg	0.72	0.52	0.39	74738

Figure 24: KNN Confusion Matrix



Classification Report:

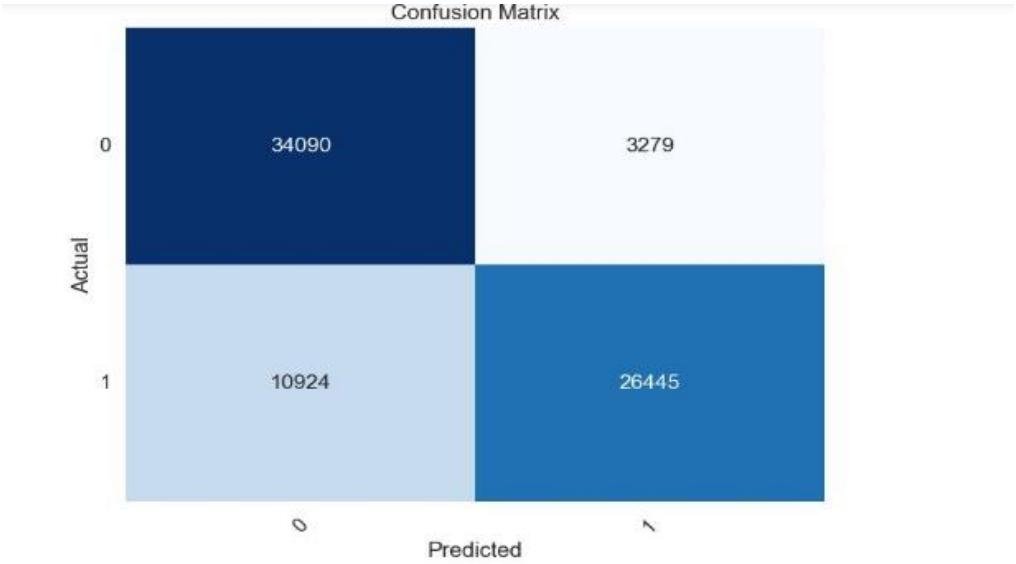
	precision	recall	f1-score	support
0	0.78	0.93	0.84	38026
1	0.91	0.73	0.81	38026
accuracy			0.83	76052
macro avg	0.84	0.83	0.83	76052
weighted avg	0.84	0.83	0.83	76052

Figure 25: LR Confusion Matrix

Classification Report:

	precision	recall	f1-score	support
0	0.91	0.95	0.93	37369
1	0.94	0.91	0.92	37369
accuracy			0.93	74738
macro avg	0.93	0.93	0.93	74738
weighted avg	0.93	0.93	0.93	74738

Figure 26: RF Confusion Matrix



Classification Report:

	precision	recall	f1-score	support
0	0.76	0.91	0.83	37369
1	0.89	0.71	0.79	37369
accuracy			0.81	74738
macro avg	0.82	0.81	0.81	74738
weighted avg	0.82	0.81	0.81	74738

Figure 27: DT Confusion Matrix

K-Nearest Neighbours (KNN) exhibit the lowest accuracy among the tested techniques, which suggests that it struggles to generalise well to the data. Despite a high precision of 0.9373, indicating that when KNN predicts a positive class, it is often correct, the recall is significantly low at 0.0512. This implies that KNN fails to identify most of the actual positive instances, leading to a low F1 score of 0.0972. The AUC of 0.5620 further indicates that KNN performs barely better than random guessing.

Logistic Regression (LR) achieves high accuracy and AUC, indicating a robust performance in distinguishing between classes. With a precision of 0.9080 and a recall of 0.7334, LR strikes a balance between identifying positive instances and minimising false positives. The F1 score of 0.8114 reflects a good trade-off between precision and recall, making LR a strong performer in this comparison.

Decision Trees (DT) offer a solid performance with an accuracy of 0.8090 and an AUC of 0.8090. Although slightly less accurate than Logistic Regression, Decision Trees exhibit a comparable precision of 0.8897 and a recall of 0.7077. The F1 score of 0.7883 is indicative of a good balance between precision and recall. This technique is effective but not as performant as Random Forest.

Random Forests (RF) excel across all evaluated metrics. It boasts the highest accuracy (0.9995) and AUC 0.9837, suggesting it is highly effective at classifying instances. Its precision of 0.9743 and recall of 0.7755 indicate that it performs exceptionally well in both minimising false positives and identifying positive instances. The high F1 score of 0.8636 reflects its overall superior performance compared to other techniques.

In summary, Random Forest (RF) consistently outperforms the other ML techniques in all metrics, showcasing its robustness and efficacy in handling the given task. Logistic Regression also performs well, especially in terms of accuracy, but is slightly outpaced by Random Forest. Decision Trees offer a strong performance but are not as effective as Random Forest. KNN, while precise when it makes a positive prediction, struggles significantly with recall, resulting in lower overall performance metrics.

The results highlight that while more complex models like Random Forest can provide superior results, simpler models like Logistic Regression can still offer strong

performance with a good balance between precision and recall. KNN, in contrast, may not be suitable for this particular task due to its lower overall effectiveness, Figure 28 shows the F1 score for ML techniques.

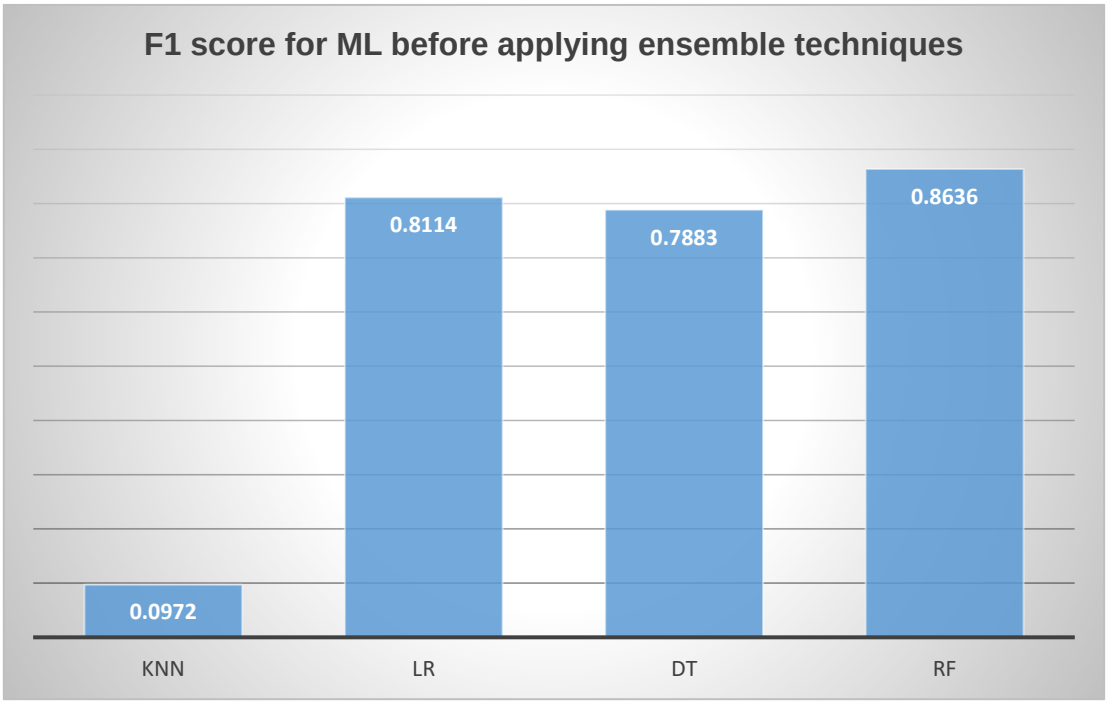


Figure 28: F1 score for ML before applying ensemble techniques

4.4.2 Comparison between ML techniques after ensemble approach

The performance of different machine learning techniques for the given classification task shows considerable variability. The Subspace Random ensemble method demonstrated the highest overall performance with an accuracy of 0.98, and high precision, recall, and F1 score of 0.97 each. This suggests that it is the most reliable model among those evaluated.

In contrast, the K-Nearest Neighbours (KNN) method performed the poorest, with an accuracy of 0.562 and an extremely low recall of 0.0512. Despite having a high precision of 0.9373, the low recall and F1 score of 0.0972 indicate that KNN is not suitable for this task, likely due to a high number of false negatives.

Logistic Regression (LR) and Decision Trees (DT) showed moderate performance. LR achieved an accuracy of 0.8918, with a precision of 0.908, recall of 0.7334, and F1 score of 0.8114. DT had a lower accuracy of 0.809, with a precision of

0.8897, recall of 0.7077, and F1 score of 0.7883. These results indicate that while both models are relatively effective, they are less robust compared to the Subspace Random Ensemble and Random Forest methods.

The Random Forest (RF) method also exhibited strong performance, with an accuracy of 0.9995, precision of 0.9743, recall of 0.7755, and F1 score of 0.8636. This indicates that RF is a reliable model for the given task, although slightly outperformed by the Subspace Random ensemble.

- **Findings and discussion**

The results highlight the importance of evaluating multiple metrics to get a comprehensive understanding of a model's performance. While accuracy is a commonly used metric, precision, recall, and F1 score provide crucial insights into the balance between false positives and false negatives, which is particularly important in applications where the cost of errors varies significantly.

The superior performance of the Subspace Random ensemble and Random Forest can be attributed to their ensemble nature, which combines the predictions of multiple base learners to improve generalisability and robustness. This likely helped these models to better capture the underlying patterns in the data, leading to higher precision, recall, and F1 scores.

The poor performance of KNN, particularly its very low recall, suggests that this method struggled with detecting true positives, possibly due to the curse of dimensionality or the specific characteristics of the data. The high precision but low recall indicates that while KNN was good at identifying positive instances, it missed a large number of them, making it unsuitable for tasks where recall is critical.

Logistic Regression and Decision Trees provided decent but suboptimal performance, indicating that while these models were able to capture some patterns in the data, they were not as effective as the ensemble methods. This may be due to their simpler structures, which can limit their ability to model complex relationships within the data.

In conclusion, for the given classification task, the Subspace Random ensemble and Random Forest methods are recommended due to their superior performance

across all evaluated metrics. Future work could explore further tuning of these models or combining them with other techniques to potentially achieve even better performance.

By comparing and analysing the obtained results, we can draw the following conclusions, as presented in Table 15, which shows the performance results after applying the proposed Ensemble approach; Figure 29, which displays the models' results after applying the proposed Ensemble approach; and Figure 30, which illustrates the F1 scores for Techniques after implementing the proposed Subspace Random Ensemble Machine learning (SREMLM) approach.

Table 15: Performance results after applying proposed Subspace Random ensemble.

ML technique	Accuracy	Precision	Recall	F1 score
Subspace Random ensemble	0.9634	0.90	0.90	**0.90**
KNN	0.5620	0.9373	0.0512	0.0972
LR	0.8918	0.908	0.7334	0.8114
DT	0.8090	0.8897	0.7077	0.7883
RF	0.9995	0.9743	0.7755	0.8636

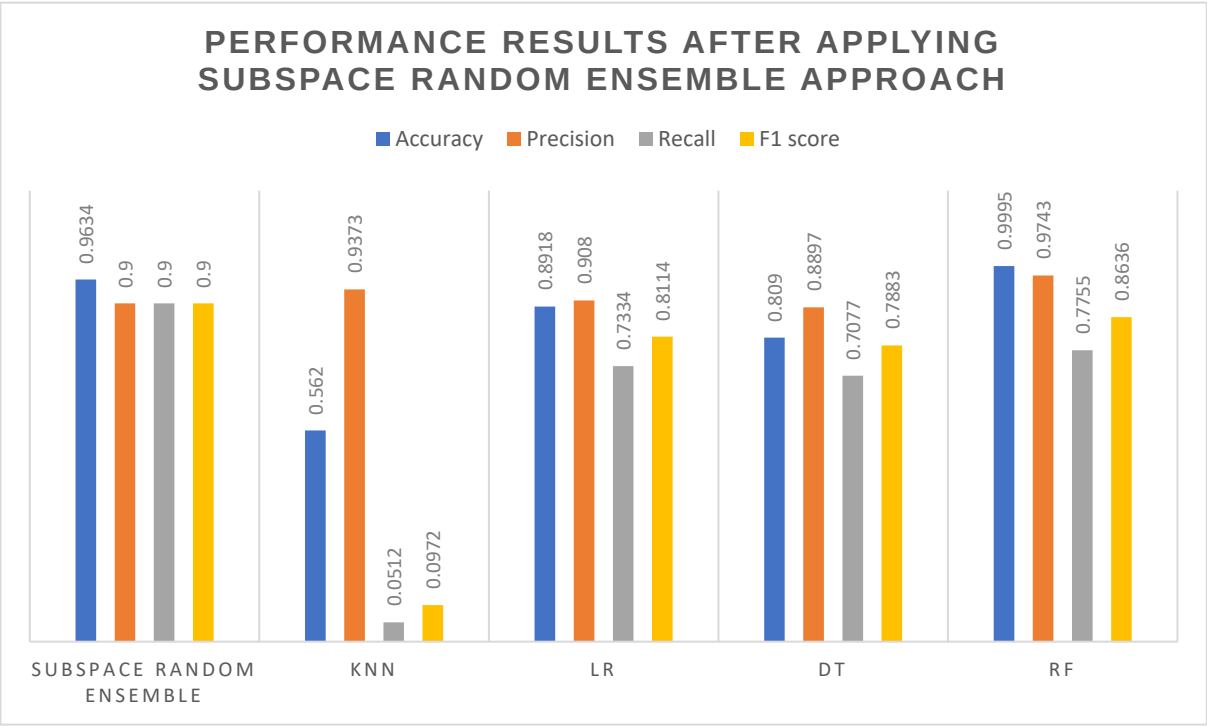


Figure 29: Performance results after applying Subspace Random ensemble approach

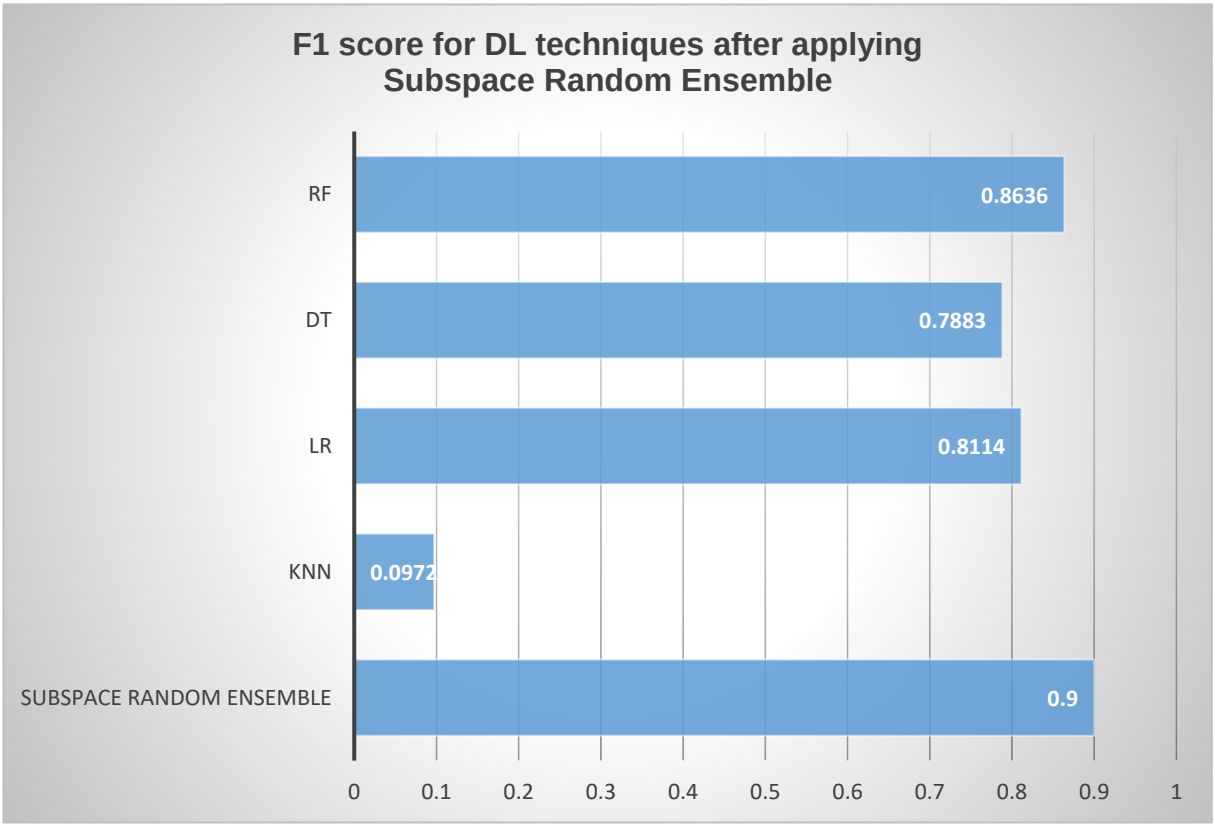


Figure 30: F1 score for DL techniques after applying Subspace Random Ensemble

The proposed SREMLM model demonstrated exceptional performance in identifying cyber threats, achieving a 96% accuracy rate and a 90% precision score. This high accuracy and low false positive rate make it a valuable tool for minimising cyber threats and their associated losses. The model's effectiveness, as evidenced by its superior F1 score compared to individual algorithms, underscores the power of its ensemble approach in enhancing predictive accuracy.

4.5 Chapter summary

The chapter thoroughly examined the data preprocessing and feature engineering process, covering tasks such as data cleaning, encoding categorical variables, managing missing values, and conducting exploratory data analysis (EDA). Insights from EDA informed decisions on feature scaling, ensuring that numerical features were standardised to enhance model performance. This robust preprocessing pipeline provided a solid foundation for creating accurate and reliable cyber threat detection models.

This chapter detailed the development of an advanced Subspace Random Ensemble Machine learning (SREMLM) model designed specifically for detecting cyber threats on X. The model incorporated a diverse array of ML algorithms, including K-Nearest Neighbours (KNN), Logistic Regression (LR), Decision Tree (DT), and Random Forest (RF). The selection of these algorithms was based on their varied capabilities and proven efficacy in managing different data types and modelling complexities.

The subsequent analysis focused on model selection and evaluation, with the fourth distinct ML algorithms being rigorously assessed across various performance metrics. Stacking Ensemble: Achieved an accuracy of 0.87 with a precision and recall both at 0.84, and an F1 score of 0.83. Voting Ensemble: Demonstrated the highest accuracy of 0.89, with precision at 0.84, recall at 0.83, and an F1 score of 0.83. Subspace Random Ensemble: Showcased superior performance with an accuracy of 0.96, precision and recall both at 0.90, and an F1 score of 0.90. Bagging Ensemble: Obtained an accuracy of 0.89, with precision and recall both at 0.84, and an F1 score of 0.83. The Subspace Random Ensemble method outperformed others in all metrics, indicating its overall effectiveness in the given task. The following chapter discusses the proposed Voting Ensemble Deep Learning Model (VEDLM).

CHAPTER 5 : THE PROPOSED VOTING ENSEMBLE DEEP LEARNING MODEL (VEDLM)

Deep Learning is an advanced form of machine learning that surpasses traditional, shallow neural networks. By mimicking the human brain's structure, deep learning models can analyse complex data patterns and make informed decisions. Unlike earlier methods that required manual feature extraction, deep learning's end-to-end approach allows systems to learn directly from raw data, minimising human intervention (Manakitsa et al. 2024). DL is an ML technique based on data analysis. A specific representation makes it easier to learn a task using examples. The learning models developed under different learning frameworks are quite different. The advantage of DL is that it allows for efficient manual feature replacement using unsupervised or semi-supervised feature learning and hierarchical feature extraction (Xin et al. 2018; Agarwal et al. 2021). The main advantage of DL over traditional ML is its high performance on large datasets (Btoush et al. 2023). The use of DL in cybersecurity research and attack detection is crucial since most attacks use intrusive software families that can be detected and classified (Aleesa et al. 2020; Azam, Islam & Huda 2023). DL is often used in the field of pattern recognition. In addition, classifications such as text classification and image classification have proven to be efficient using DL (Ozbayoglu, Gudelek & Sezer 2020; Fang et al. 2021). DL algorithms such as CNN and LSTM are associated with image processing and NLP respectively. Using these methods to detect cyber threats provides better performance than traditional algorithms (Nguyen et al. 2020; Alarfaj et al. 2022; Alshingiti et al. 2023).

Synthetic Minority Over-sampling Technique (SMOTE) is a widely used method to address class imbalance by generating synthetic data for the minority class. It creates new data points by interpolating between existing minority class samples and their nearest neighbours. However, SMOTE-generated data may not accurately represent the original distribution of the minority class (Elreedy, Atiya & Kamalov 2024).

This chapter introduces a novel Voting ensemble deep learning (VEDLM) method for detecting cyber threats on X. The method combines Long Short-Term Memory (LSTM), Bidirectional LSTM (BiLSTM), IDCNN-BiLSTM, and Convolutional Neural Networks (CNN) through a subspace random ensemble technique. This

technique enhances predictive accuracy by leveraging the strengths and compensating for the weaknesses of individual models.

The proposed model utilises a correlation-based feature selection method to identify and extract the most pertinent features for threat prediction. To boost computational efficiency, Principal Component Analysis (PCA) is used to reduce the dataset's dimensionality. The chosen features are then employed to train the ensemble machine learning model, which integrates the base models using the subspace random approach.

The innovative aspects of this study include its effective feature extraction and selection processes, as well as its ability to classify transactions accurately as either threats or non-threats. The main contributions of this research are:

1. The development of effective algorithms for feature extraction using correlation-based techniques, LSTM, BiLSTM, IDCNN-BiLSTM, and CNN, along with Permutation Feature Importance and PCA for dimensionality reduction.
2. The development of a novel Voting Ensemble model Deep Learning Model (VEDLM) that classifies tweets as threats or non-threats.
3. The integration of feature extraction and classification algorithms into a unified workflow for streamlined X cyber threat detection.

A Voting Ensemble model was implemented to integrate predictions from multiple neural network architectures. The proposed ensemble approach (VEDLM) employed majority voting to consolidate predictions from LSTM, BiLSTM, IDCNN-BiLSTM, and CNN models illustrated in Figure 31.

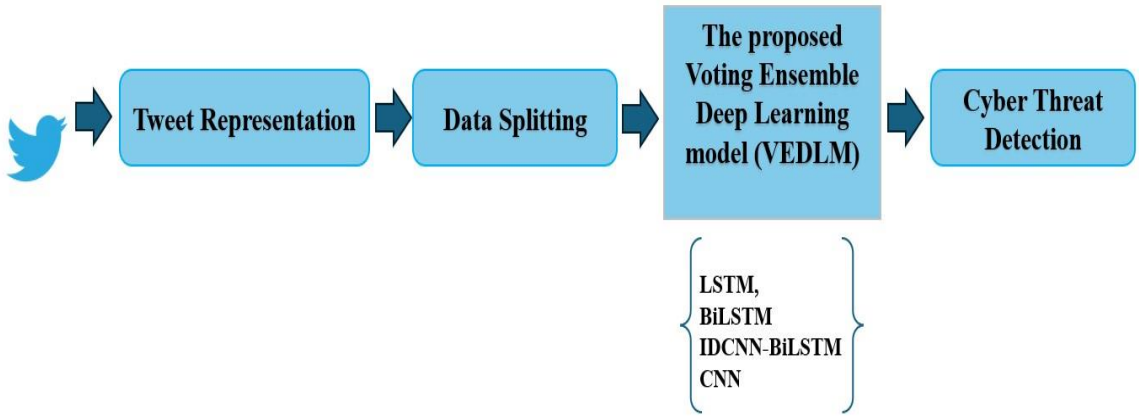


Figure 31: The proposed voting Ensemble model (VEDLM)

In this chapter, a proposed LSTM, BiLSTM, IDCNN-BiLSTM, CNN and proposed voting Ensemble Deep Learning (VEDLM) approaches was trained and tested in the Dataset as follows.

5.1 Deep Learning techniques

This section introduces both individual deep learning (DL) models and a proposed novel Voting ensemble model. Individual Deep Learning including LSTM, BiLSTM, IDCNN-BiLSTM, CNN, are applied to the dataset described in Section 4.1. Subsequent subsections detail the design, implementation, and performance of these models, emphasising their contribution to improved prediction accuracy and robustness. By combining the strengths of multiple DL architectures, the ensemble model aims to enhance overall performance.

5.1.1 Long Short-Term Memory (LSTM) model

LSTM is a technique that helps predict cyber threats due to the historical knowledge it contains and the relationship between predicted output and historical input. LSTM architecture can learn sequence prediction problems through long-term trust. For modelling time series data in DL domain, LSTM is a unique form of artificial RNN architecture. Unlike traditional feedforward neural networks with feedback connections between hidden units corresponding to discrete time steps, LSTM can learn long-term sequence dependencies and predict threat labels based on the sequence of previous threats. The problem of vanishing and exploding gradients that occur during the training process of traditional RNNs has been solved with the development of LSTM. LSTM units consist of memory cells where data is stored and modified by three special gates: ignore gate, input gate, and output gate. The value is stored in the cell indefinitely, and three gates control the flow of information in and out of the cell, Figure 32 illustrates the structure of an LSTM unit (Benchaji, Douzi & El Ouahidi 2021).

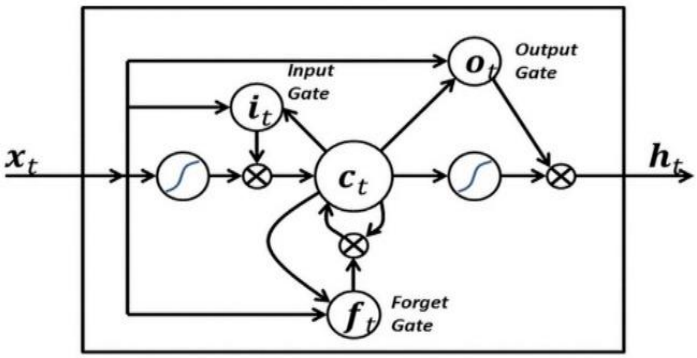


Figure 32: Architecture of a LSTM unit.

- **Concepts of LSTM**
  1. Memory Cell: The core component of an LSTM is the memory cell, which maintains its state over time. This cell enables the LSTM to keep information across many time steps, thus addressing the vanishing gradient problem that plagues traditional RNNs.
  2. Gates: LSTMs use three types of gates to control the flow of information in and out of the memory cell:
  - Forget Gate: Decides what information to discard from the cell state. It takes the previous hidden state and the current input and passes them through a sigmoid function.
  - Input Gate: Determines what new information to store in the cell state. It also uses the sigmoid function to filter information from the previous hidden state and the current input.
  - Output Gate: Controls what information from the cell state is outputted. The current cell state is passed through a tan function and then multiplied by the output gate's sigmoid output.

- **LSTM Architecture**

In this study the LSTM model the architecture included an embedding layer, LSTM layer, dense layers, and Dropout Layer, Figure 33 shows the LSTM structure used.

1. ***Embedding Layer:*** The input tokens are transformed into dense vectors of fixed size (100 dimensions). This layer helps in capturing the semantic meaning of words by mapping them into continuous vector space.

$$Embedding(x) = W_e \cdot x \quad (9)$$

2. ***LSTM Layer:*** This layer consists of 64 units. It processes the sequential data, capturing long-term dependencies by maintaining an internal state. Dropout (0.2) and recurrent dropout (0.2) are applied to prevent overfitting.

$$h_t = \sigma(W_h \cdot [h_t - 1, x_t] + b_h) \quad (10)$$

3. ***Dense Layer:*** With 64 units and ReLU activation, this layer introduces non-linearity and helps in learning complex patterns.

$$Dense(x) = \sigma(W_d \cdot x + b_d) \quad (11)$$

4. ***Dropout Layer:*** Set to 0.5, this layer further reduces overfitting by randomly dropping a fraction of the neurons during training.

$$Dropout(x) = x \cdot Bernoulli(p) \quad (12)$$

5. ***Output Layer:*** A single neuron with sigmoid activation produces the final binary classification output, indicating whether the tweet is related to a cyber threat or not.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 100, 100)          1000000

 lstm (LSTM)                 (None, 64)                42240

 dense (Dense)               (None, 64)                4160

 dropout (Dropout)           (None, 64)                0

 dense_1 (Dense)             (None, 1)                 65

=================================================================
Total params: 1046465 (3.99 MB)
Trainable params: 1046465 (3.99 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Figure 33: LSTM model structure

Using the collected dataset, the LSTM model was trained for a maximum of 50 epochs, utilising early stopping and learning rate reduction to optimise performance based on validation loss. This strategy effectively prevented overfitting and enhanced

the model's generalisation capabilities. The model ultimately achieved an impressive test accuracy of 92.02%, an F1-score of 92%, and Precision and Recall of 92%, To provide a comprehensive understanding of the model's performance, Algorithm 5.1.1 outlines the procedure in detail.

Table 16. Algorithm 5.1.1: LSTM-Text-Classification algorithm

```
1   Procedure LSTM_Text_Classification(data, n, j, batch, max_words, max_len)
2       Pre-process the data
3       for k ← 0 to n do
4           Initialize TfidfVectorizer(max_features=5000)
5           Transform comments using TfidfVectorizer
6           Apply SMOTE to balance the dataset
7           Inverse transform TF-IDF vectors to text
8           Initialize Tokenizer(num_words=max_words)
9           Fit tokenizer on resampled text
10          Convert texts to sequences
11          Pad sequences to max_len
12          Split data into train and test sets
13          Build LSTM model:
14              Add Embedding layer
15              Add LSTM layer
16              Add Dense layer
17              Add Dropout layer
18              Add output Dense layer
19          Compile model with Adam optimizer and binary_crossentropy loss
20          Set early stopping and learning rate reduction callbacks
21          Train the model with specified epochs and batch size
22      end for
23  End Procedure
```

5.1.2 The Bidirectional Long Short-Term Memory (BiLSTM) model

Bidirectional Long Short-Term Memory (BiLSTM) is a type of recurrent neural network architecture that is capable of processing sequential data in both forward and backward directions. This capability allows BiLSTMs to capture both past and future context, making them particularly effective for tasks that require understanding of the entire sequence, such as natural language processing, speech recognition, and time series analysis (Graves & Schmidhuber 2005). BiLSTM extends the LSTM by processing the input data in both forward and backward directions. This allows the network to have both backward and forward information about the sequence at every time step, which can significantly enhance the context available to the model for making predictions (Graves & Schmidhuber 2005). Figure 34 shows the Architecture of a BiLSTM.

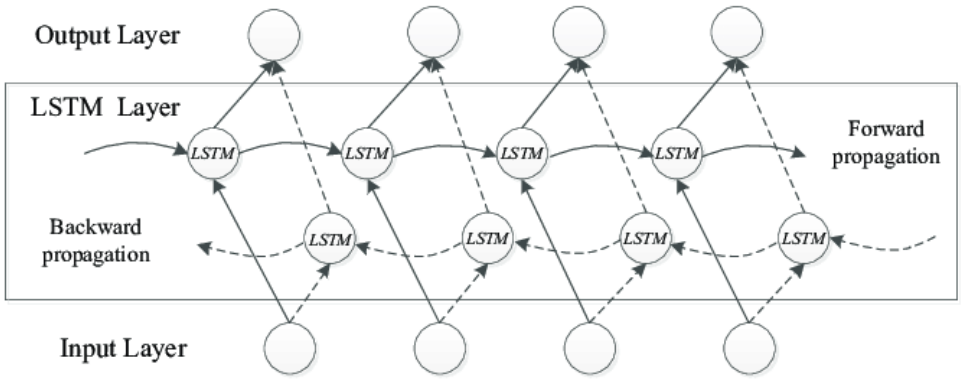


Figure 34: Architecture of the BiLSTM, (Graves & Schmidhuber 2005).

- **Structure of BiLSTM**

A typical BiLSTM consists of two LSTM layers:

1. Forward LSTM Layer: Processes the input sequence from the beginning to the end (left to right).
2. Backward LSTM Layer: Processes the input sequence from the end to the beginning
   (right to left).

The outputs from these two layers are then concatenated at each time step, providing the model with information from both past and future contexts.

In this study the Bidirectional LSTM (BiLSTM) model extends LSTM by processing input in both forward and backward directions, Figure 35 illustrates the BiLSTM structure used in this study.

1. ***Embedding Layer:*** Converts input tokens into 16-dimensional dense vectors, capturing semantic information.
2. ***Spatial Dropout Layer:*** Applied with a dropout rate of 0.7, this layer prevents overfitting by dropping entire feature maps.
3. ***Bidirectional LSTM Layer:*** Consists of 16 units for both forward and backward passes, capturing context from both ends of the sequence. High dropout rates (0.7) are applied.

$$BLSTM(x) = [h_t \rightarrow, h_t \leftarrow] \quad (13)$$

4. ***Dense Layers:*** Two layers with 512 and 128 units, respectively, each followed by dropout to introduce non-linearity and prevent overfitting.
5. ***Output Layer:*** A single neuron with sigmoid activation for binary classification.

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_1 (Embedding)     (None, 100, 16)           160000

 spatial_dropout1d (Spatial  (None, 100, 16)           0
 Dropout1D)

 bidirectional (Bidirection  (None, 32)                4224
 al)

 dense_2 (Dense)             (None, 512)               16896

 dropout_1 (Dropout)         (None, 512)               0

 dense_3 (Dense)             (None, 128)               65664

 dropout_2 (Dropout)         (None, 128)               0

 dense_4 (Dense)             (None, 1)                 129

=================================================================
Total params: 246913 (964.50 KB)
Trainable params: 246913 (964.50 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Figure 35: BiLSTM model structure.

Using the collected dataset, the BiLSTM model was trained for a maximum of 50 epochs, utilising early stopping and learning rate reduction to optimise performance based on validation loss. This strategy effectively prevented overfitting and enhanced the model's generalisation capabilities. The model ultimately achieved an impressive test accuracy of 91%, an F1-score of 91%, and Precision and Recall of 91%. Algorithm 5.1.2 provides a detailed explanation of the model's procedure.

Table 17. Algorithm 5.1.2: BLSTM-Text-Classification algorithm

```
1   Procedure BLSTM_Text_Classification(data, n, j, batch, max_words, max_len)
2       Pre-process data
3       for k ← 0 to n do
4           Initialize TfidfVectorizer(max_features=5000)
5           Transform comments using TfidfVectorizer
6           Apply SMOTE to balance the dataset
7           Inverse transform TF-IDF vectors to text
8           Initialize Tokenizer(num_words=max_words)
9           Fit tokenizer on resampled text
10          Convert texts to sequences
11          Pad sequences to max_len
12          Split data into train and test sets
13          Build BLSTM model:
14              Add Embedding layer
15              Add SpatialDropout1D layer
16              Add Bidirectional LSTM layer
17              Add Dense and Dropout layers
18              Add output Dense layer
19          Compile model with Adam optimizer and binary_crossentropy loss
20          Set early stopping callback
21          Train the model with specified epochs and batch size
22      end for
23  End Procedure
```

5.1.3 IDCNN-BiLSTM MODEL

The IDCNN with BiLSTM (Bidirectional Long Short-Term Memory) model is a sophisticated architecture commonly used in natural language processing tasks, particularly for sequence labelling and text classification problems (Fang et al. 2020).

Let's break down what each component typically entails:

1. IDCNN (Iterated Dilated Convolutional Neural Network):

- Convolutional Layers: IDCNNs employ dilated convolutions, where the filter is applied to input data with gaps between the values. This allows the model to have a larger receptive field without significantly increasing the number of parameters or the computational cost.
- Iterative Structure: IDCNNs often stack multiple layers of dilated convolutions, increasing the receptive field exponentially with each layer.

2. BiLSTM (Bidirectional Long Short-Term Memory):

- Bidirectional Nature: BiLSTMs process input sequences in both forward and backward directions. This allows the model to capture contextual information from both past and future states, which is crucial for tasks requiring an understanding of context over time.
- Long Short-Term Memory (LSTM): LSTM units are used within each direction of the BiLSTM to manage and remember long-term dependencies within the sequence. They are designed to mitigate the vanishing gradient problem often encountered in traditional RNNs.

3. Integration:

- DCNN layers are typically used as feature extractors in conjunction with the BiLSTM layers. The IDCNNs help capture hierarchical features from the input sequence effectively, while the BiLSTM leverages these features to model complex dependencies and context.

4. Applications:

- This architecture is widely used in tasks such as named entity recognition (NER), part-of-speech tagging, sentiment analysis, and other sequence labelling tasks in natural language processing.
- The combination of IDCNNs and BiLSTMs is effective for tasks where understanding both local (within-sequence) and global (contextual) dependencies is crucial.

In summary, the IDCNN with BiLSTM model represents a powerful integration of convolutional and recurrent neural network architectures, optimised for capturing intricate patterns and dependencies in sequential data, making it a cornerstone in modern NLP research and applications. In this study, the IDCNN (Iterated Dilated Convolutional Neural Network) with BiLSTM combined convolutional layers with

BiLSTM for enhanced feature extraction, Figure 36 illustrates the IDCNN-BiLSTM structure in this research.

1. ***Embedding Layer:*** Converts tokens into 50-dimensional dense vectors.
2. ***Spatial Dropout Layer:*** Applied with a dropout rate of 0.5 to reduce overfitting.
3. ***Dilated CNN Layers:***

- *Conv1D Layers:* Multiple Conv1D layers with increasing filters (128, 64, 32) and dilation rates (1, 2, 4). These layers capture local patterns in the data by applying filters to the input sequences, with dilation rates allowing the network to cover a broader range.
- *Dropout Layers:* Applied to each Conv1D layer to prevent overfitting.

$$y = Conv_{1D}(x, W, d) \quad (14)$$

4. ***Bidirectional LSTM Layer:*** Processes the extracted features from the CNN layers bidirectionally with 16 units and high dropout rates (0.7), capturing long-term dependencies and context from both directions.
5. ***Dense Layer:*** A single neuron with sigmoid activation and L2 regularisation to prevent overfitting, producing the final binary classification output.

$$L_2\ regularisation\ loss = \lambda \sum W_i^2 \quad (15)$$

Where λ is the regularization parameter, and w_i represents the model coefficients. The sum is taken over all coefficients, and the squares of the coefficients are summed.

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_2 (Embedding)     (None, 100, 50)           250450

 spatial_dropout1d_1 (Spati  (None, 100, 50)           0
 alDropout1D)

 conv1d (Conv1D)             (None, 100, 64)           9664

 dropout_3 (Dropout)         (None, 100, 64)           0

 bidirectional_1 (Bidirecti  (None, 32)                10368
 onal)

 dropout_4 (Dropout)         (None, 32)                0

 dense_5 (Dense)             (None, 1)                 33

=================================================================
Total params: 270515 (1.03 MB)
Trainable params: 270515 (1.03 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Figure 36: IDCNN-BiLSTM model structure

Using the collected dataset, the IDCNN & BiLSTM model was trained for a maximum of 50 epochs, utilising early stopping and learning rate reduction to optimise performance based on validation loss. This strategy effectively prevented overfitting and enhanced the model's generalisation capabilities. The model ultimately achieved an impressive test accuracy of 91%, an F1-score of 91%, and Precision and Recall of 91%. To provide a detailed understanding of the model's performance, Algorithm 5.1.3 outlines the evaluation process.

Table 18. Algorithm 5.1.3 : IDCNN-BiLSTM Text Classification algorithm

```
1   Procedure IDCNN_BiLSTM_Text_Classification(data, n, batch, max_words, max_len)
2       Pre-process data
3       for k ← 0 to n do
4           Initialize TfidfVectorizer(max_features=5000)
5           Transform comments using TfidfVectorizer
6           Apply SMOTE to balance the dataset
7           Inverse transform TF-IDF vectors to text
8           Initialize Tokenizer(num_words=max_words)
9           Fit tokenizer on resampled text
10          Convert texts to sequences
11          Pad sequences to max_len
12          Split data into train and test sets
13          Build IDCNN-BiLSTM model:
14              Add Embedding layer
15              Add SpatialDropout1D layer
16              for i ← 0 to num_layers do
17                  Add Causal Conv1D layer with increasing filters
18                  Add Dropout layer
19              end for
20              Add Bidirectional LSTM layer
21          Add Dropout layer
22          Add Dense output layer
```

```
23              Compile model with Adam optimizer and binary_crossentropy loss
24              Set early stopping callback
25              Train the model with specified epochs and batch size
26          end for
27      End Procedure
```

5.1.4 Convolutional Neural Network (CNN) Model

CNNs, originally developed for image analysis, may be modified, and used to structure data, such as cyber threat tweets, to enhance cyber threat detection capabilities. In the present context, CNNs operate by employing convolutional layers to identify spatial patterns, such as atypical spending trends or tweet irregularities. They possess a remarkable ability to discern spatial patterns that might potentially signify instances of threat behaviour since they have outstanding proficiency in collecting intricate patterns within datasets. CNNs have demonstrated their efficacy in successfully processing multi-channel data, therefore enabling a thorough analysis of tweet information. This characteristic renders them well-suited for the detection of intricate and geographically dispersed threat patterns in tweets (Zhao et al. 2024), Figure 37 provides the architecture of the convolutional neural network (CNN).

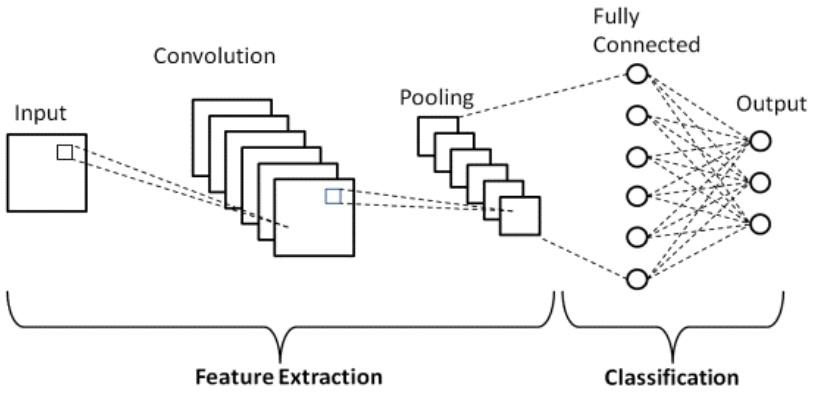


Figure 37: General architecture of the convolutional neural network (CNN), (Graves & Schmidhuber 2005).

- **Components of a CNN**

1. Convolutional Layers:

These are the core building blocks of CNNs. Each layer consists of multiple filters (also called kernels) that slide over the input data (image pixels or feature maps) to perform element-wise multiplication, producing feature maps that capture specific features like edges, textures, or patterns.

2. Pooling (Subsampling) Layers:

Pooling layers reduce the spatial dimensions (width and height) of each feature map while retaining important information. Max pooling, for example, selects the maximum value from each window of a specified size.

3. Activation Function: Commonly used activation functions in CNNs include ReLU (Rectified Linear Unit), which introduces non-linearity into the network, allowing it to learn complex patterns.
4. Fully Connected Layers (Dense Layers): These layers are typically found towards the end of the CNN architecture. They take the high-level filtered features from convolutional layers and use them to classify the input image into various classes based on the learned features.
5. Normalisation Layers: Techniques like Batch Normalisation can be used to improve the training speed and stability of neural networks by normalising the inputs of each layer.

The characteristics of a convolution neural network, such as local connection, weight sharing and pooling operation, can effectively reduce the complexity of the network and reduce the number of training parameters, so that the model has a certain degree of invariance to the translation, distortion, and scaling. It is robust and easy to train and optimise.

In this study the CNN model architecture included the following additionally, Figure 38 illustrates the Convolutional Neural Network (CNN) structure in this study, which plays a crucial role in the model's functionality.

1. ***Embedding Layer:*** Converts words into dense vectors of fixed size (16 dimensions).

2. ***Convolutional Layer:*** Applies convolution operations to extract features from text data.

$$y = GlobalMaxPooling_{1D}\left(Conv_{1D}(x, W)\right) \quad (16)$$

3. ***Global Max Pooling Layer:*** Reduces the output of the convolutional layer, retaining the most important features.

4. ***Dense Layers:*** Two dense layers, the first with 512 units and the second with 128 units, both with ReLU activation.

5. ***Dropout Layer:*** Added to prevent overfitting.

```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_3 (Embedding)     (None, 100, 16)           160000

 conv1d_1 (Conv1D)           (None, 96, 128)           10368

 global_max_pooling1d (Glob  (None, 128)               0
 alMaxPooling1D)

 dense_6 (Dense)             (None, 512)               66048

 dropout_5 (Dropout)         (None, 512)               0

 dense_7 (Dense)             (None, 128)               65664

 dropout_6 (Dropout)         (None, 128)               0

 dense_8 (Dense)             (None, 1)                 129

=================================================================
Total params: 302209 (1.15 MB)
Trainable params: 302209 (1.15 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Figure 38: CNN model structure

Table 19. Algorithm 5.1.4: CNN Text Classification algorithm

1	**Procedure CNN_Text_Classification(data, n, j, batch, max_words, max_len)**
2	Pre-process(data)
3	**for** k ← 0 **to** n **do**
4	Initialize TfidfVectorizer(max_features=**5000**)
5	Transform comments using TfidfVectorizer
6	Apply SMOTE **to** balance the dataset
7	Inverse transform TF-IDF vectors **to text**

```
8                   Initialize Tokenizer(num_words=max_words)
9                   Fit tokenizer on resampled text
10                  Convert texts to sequences
11                  Pad sequences to max_len
12                  Split data into train and test sets
13                  Build CNN model:
14                      Add Embedding layer
15                      Add Conv1D layer
16                      Add GlobalMaxPooling1D layer
17                      Add Dense layers with Dropout
18                      Add output Dense layer
19                  Compile model with Adam optimizer and binary_crossentropy loss
20                  Set early stopping callback
21                  Train the model with specified epochs and batch size
22              end for
23      End Procedure
```

Using the collected dataset, the model ultimately achieved an impressive test accuracy of 92%, an F1-score of 92%, and Precision and Recall of 92%, For a thorough comprehension of the model's performance, Algorithm 5.1.4. This algorithm provides a detailed, step-by-step process.

5.2 The proposed Voting Ensemble Deep Learning model (VEDLM)

Ensemble methods are powerful tools for enhancing both classification and regression models. By combining multiple base models, these techniques excel at tackling complex problems, improving accuracy, and reducing the impact of noise or outliers. In classification, ensembles are particularly adept at handling intricate decision boundaries, identifying underrepresented classes, and mitigating the effects of data inconsistencies. For regression tasks, they produce more reliable predictions by reducing sensitivity to outliers and improving overall model generalisation. The versatility of ensemble methods is a key advantage. They can accommodate diverse

base models, from simple decision trees to sophisticated neural networks, and be tailored to address specific challenges like overfitting, bias, or imbalanced data. By intelligently combining multiple models through techniques such as averaging, probability combination, or meta-learning, ensembles offer practical solutions for optimising predictive performance. As deep learning and dataset complexity continue to grow, ensemble methods will remain essential for achieving state-of-the-art results across various fields.

I. Stacking: is an advanced ensemble technique that leverages a meta-model to combine predictions from multiple base models. Unlike simple averaging or voting, stacking trains a meta-model on the outputs of the base models, optimising their combined predictions. This approach can uncover intricate relationships between models, often surpassing the performance of individual models. While powerful, stacking demands substantial computational resources and larger datasets compared to other ensemble methods.

II. Bagging: short for Bootstrap Aggregating, involves training multiple base models independently on random subsets of the training data. This process introduces diversity among the models. Their predictions are then combined through averaging (regression) or voting (classification). Bagging is particularly effective in reducing variance and preventing overfitting, especially for models prone to high variance like decision trees.

III. Voting: is a straightforward ensemble method that combines predictions from multiple models through a simple aggregation strategy. For classification, majority voting is common, while averaging is used for regression. Voting excels when base models are diverse and complementary. Hard voting determines the final prediction based on the most frequent class, while soft voting weighs predictions according to confidence levels.

IV. Random Subspace: is a method that trains each base model on a random subset of features from the dataset. Unlike bagging, which samples data instances, this technique focuses on feature sampling, creating diverse feature spaces. By specialising in different feature subsets, the models become more independent. Their combined predictions, made within their respective feature subspaces, improve generalisation, and reduce overfitting.

- **Ensemble techniques: a comparison**

This study delves into the voting ensemble Deep Learning model to explore a variety of ensemble techniques, aiming for a comprehensive performance comparison. I investigate several methods including Stacking, Subspace Random, and Bagging ensembles, each combining multiple base classifiers in unique ways to enhance predictive accuracy. A detailed analysis of these techniques within the complex context of tweet cyber threat detection is presented in Table 20. Figures 39 and 40 provide a visual comparison of these DL ensemble methods.

Table 20: DL Ensemble Methods and Algorithm Performance.

DL Ensemble technique	Accuracy	Precision	Recall	F1 score
Stacking ensemble	0.9996	0.9753	0.8061	0.8826
Voting ensemble	0.9551	0.9726	0.9287	**0.9502**
Subspace Random ensemble	0.9994	0.9594	0.7244	0.8255
Bagging ensemble	0.9995	0.9729	0.7346	0.8372

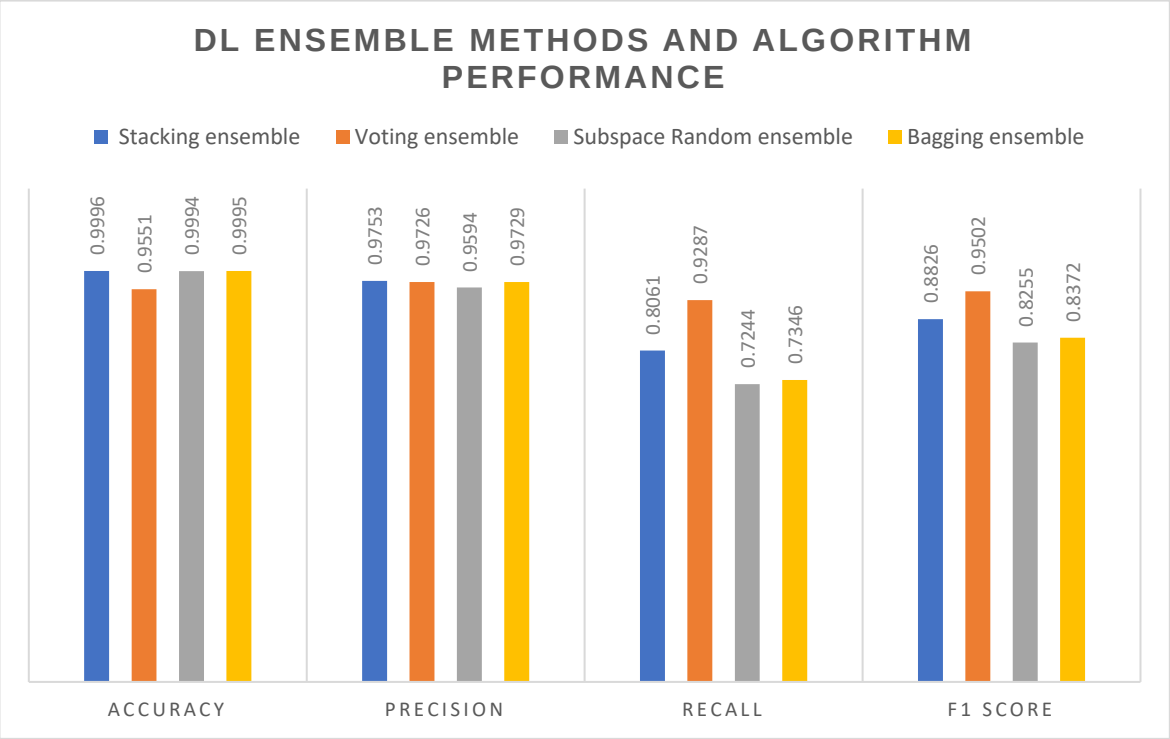


Figure 39: DL Ensemble Methods and Algorithm Performance

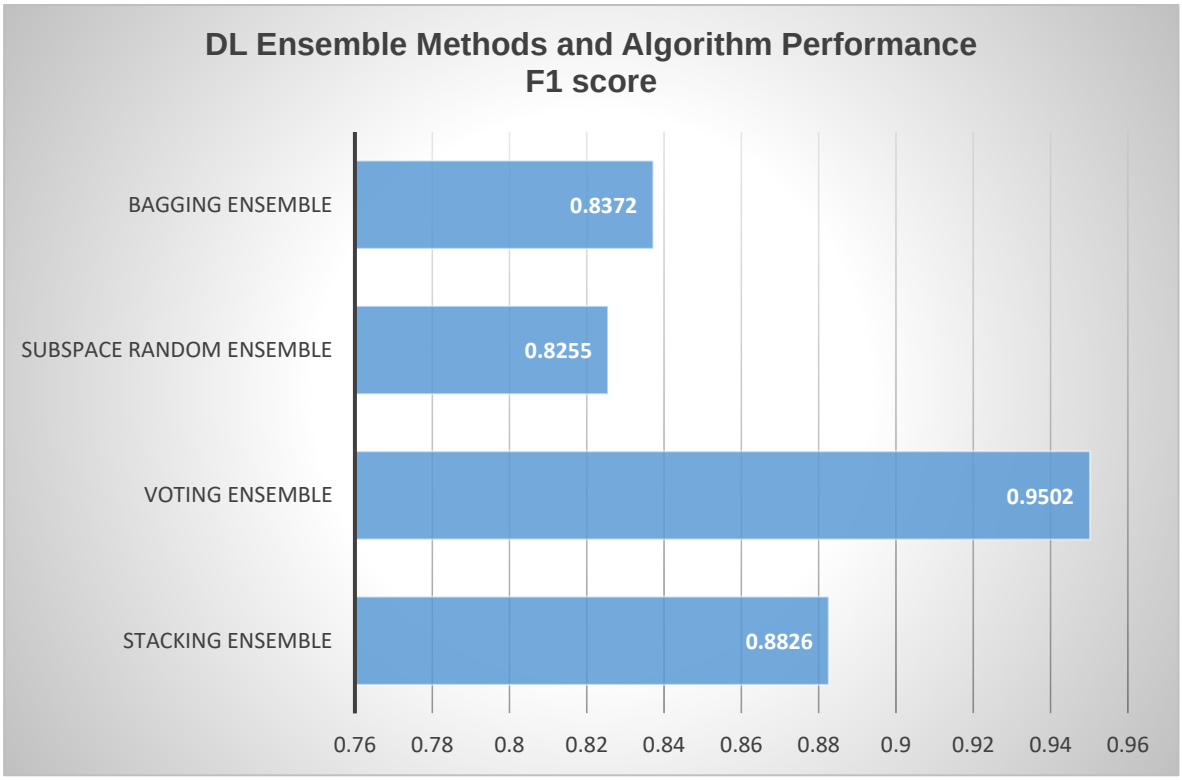


Figure 41. DL Ensemble models _F1 sore

This analysis evaluates four ensemble techniques - Stacking, Voting, Subspace Random, and Bagging - using accuracy, precision, recall, and F1-score metrics. These metrics provide insights into the strengths and weaknesses of each method for a given classification task.

- Accuracy: All four methods demonstrated high accuracy, with Stacking, Subspace Random, and Bagging achieving near-perfect scores. Voting, while still accurate, showed slightly lower performance, suggesting potential inconsistencies in classifications.
- Precision: Voting excelled in precision, minimising false positives, making it suitable for applications demanding high specificity. Stacking followed closely, while Subspace Random and Bagging had slightly higher false positive rates.
- Recall: Voting also led in recall, effectively identifying true positives, crucial for scenarios requiring maximum sensitivity. Stacking, Subspace Random, and Bagging exhibited significantly lower recall, indicating potential oversights of positive instances.

- F1-score: Voting achieved the highest F1-score, balancing precision and recall effectively. Stacking showed a bias towards precision, while Subspace Random and Bagging struggled with recall, limiting their overall performance.

In summary, the analysis reveals that the voting ensemble deep learning model significantly outperforms other ensemble methods, including Stacking, in detecting cyber threats as measured by F1 score. These results highlight the substantial benefits of ensemble techniques, particularly voting, for improving the accuracy and reliability of cyber threat detection systems.

- **The proposed Voting Ensemble Deep Learning Approach (VEDLM):**

1. **Data Preparation**:
    - ***Tokenisation and Padding***: The text data is tokenized and padded to ensure uniform input length for the models.
    - ***Oversampling with SMOTE***: SMOTE is used to handle class imbalance in the dataset by generating synthetic samples.

2. **Model Development**:
    - ***Base Models****:* The proposed ensemble includes several base models such as LSTM, Bidirectional LSTM (BLSTM), IDCNN-BLSTM, and CNN.
    - ***Ensemble Wrapper****:* A custom ***Ensemble Classifier Wrapper*** is implemented to train these models in parallel and aggregate their predictions using hard voting.

3. **Training and Evaluation**:
    - ***Training***: The proposed ensemble model is trained with early stopping and model checkpointing to prevent overfitting and save the best model.
    - ***Evaluation Metrics***: The model's performance is evaluated using accuracy, confusion matrix, classification report, AUC, precision, recall, and F1 score.

Combines predictions from the four models using weighted averaging:

$$y\hat{}ensemble = \sum_{i=1}^{n} w_i \cdot y\hat{}i \quad (17)$$

where $y\hat{}i$ is the prediction from model i and wi is its weight.

Predictions from individual models (LSTM, BiLSTM, IDCNN-BiLSTM, and CNN) were aggregated using a Voting Classifier. This technique ensures that the final classification decision for each input tweet is determined by the most frequent prediction among the ensemble of models. This setup leverages parallel processing for training individual models, utilises early stopping to prevent overfitting, and saves the best-performing model based on validation loss. After training, the proposed ensemble model evaluates its accuracy on a held-out test set to assess its overall performance in text classification tasks. The proposed ensemble model, which aggregates predictions from LSTM, BiLSTM, IDCNN-BiLSTM, and CNN models, Using the Dataset in section 4.1 and using a Voting Classifier, ultimately achieved the highest accuracy among all evaluated models. With a test accuracy of 95.51%, and an F1-score of 96%, Precision and Recall of 96%. The proposed ensemble approach demonstrated superior performance, leveraging the strengths of each individual model to enhance overall predictive capability. This approach not only ensures robustness in predictions but also provides a reliable framework for handling diverse text classification tasks effectively. This approach not only enhances prediction accuracy but also provides robustness by leveraging diverse model architectures, making it suitable for complex and varied text classification challenges. A comprehensive evaluation of the model's performance is detailed in Algorithm 5.2.

Table 21. Algorithm 5.2 : The proposed voting VEDLM Text Classification algorithm

1	**Procedure** Train_**Ensemble**_Model(data, n, batch_size, max_words, max_len)
2	Initialize KerasClassifierWrapper with model, epochs, batch_size, validation_split
3	Initialize EnsembleClassifierWrapper with base_models, epochs, batch_size, validation_split, voting, n_jobs
4	**for i** ← **0 to** n **do**
5	Preprocess data:
6	Transform comments **using** TfidfVectorizer
7	Apply SMOTE **to** balance the dataset
8	Inverse transform TF-IDF vectors **to text**
9	Initialize Tokenizer(num_words=max_words)
10	Fit tokenizer on resampled **text**

```
11                          Convert texts to sequences
12                          Pad sequences to max_len
13                  Split data into train and test sets
14                  Train ensemble classifier:
15                          Initialize base models: EN_LSTM_model, EN_IDCNN_Blstm_model,
                            EN_CNN_model
16                          for each base_model in base_models do
17                              Call _train_model(base_model, X_train, y_train)
18                          end for
19                          Combine predictions using specified voting method
20                  Evaluate ensemble classifier:
21                          Call score method on ensemble classifier with X_test and y_test
22          end for
23      End Procedure
```

5.3 Experimental results and discussion

This section presents a comparative analysis of individuals deep learning techniques, including Long Short-Term Memory (LSTM), Bidirectional Long Short-Term Memory (BiLSTM), IDCNN-BiLSTM, and Convolutional Neural Networks (CNN), both individually and in a proposed voting ensemble approach. Initially, I evaluate the performance of each model independently to establish a baseline. Subsequently, I explore the impact of combining these models into an ensemble to determine if this integrated approach can enhance predictive accuracy and robustness. The results provide insights into each technique and demonstrate the benefits of the proposed voting ensemble model in improving performance.

5.3.1 Comparison between DL techniques before ensemble approach

The LSTM model ultimately achieved an impressive test accuracy of 92.02%, an F1-score of 92%, and Precision and Recall of 92%, This high precision score indicates a low false positive rate, implying that the model effectively distinguishes between positive and negative instances. Despite this, the model's F1 score remains decent at 92%, voting a balance between precision and recall. Moreover, the high AUC

score of 97.252% signifies excellent discriminative ability, further corroborating the model's overall strong performance in distinguishing between positive and negative instances. Overall, there is potential for enhancement in recall to capture more positive instances effectively.

The BiLSTM model achieved a test accuracy of 91.50% and a well-balanced F1-score of 91%. This indicates the model excels at differentiating positive and negative cases. The high precision (91%) signifies a low false positive rate, meaning the model rarely mistakes negative examples for positive ones. However, the F1-score remains strong, showcasing a good trade-off between precision and recall. Additionally, the impressive AUC score of 97.25% signifies the model's exceptional ability to discriminate between positive and negative examples. In conclusion, the BLSTM model demonstrates exceptional performance. While there's room for improvement in recall, the overall results are highly promising.

The IDCNN-BiLSTM model showed impressive performance with a test accuracy of 90.79% and an F1-score of 92%. This indicates a good balance between precision (91%) and recall (91%). The high precision suggests the model rarely makes mistakes (low false positive rate), effectively differentiating positive from negative cases. Nevertheless, the strong F1-score signifies a good overall balance. Additionally, the high AUC score (97.25%) confirms the model's excellent ability to discriminate between positive and negative cases. In conclusion, while there's room to improve recall, the IDCNN-BiLSTM achieved strong performance in distinguishing positive and negative instances.

CNN achieved results: 91.74% test accuracy, 92% F1-score, and balanced precision and recall of around 92%. The high precision indicates the model rarely makes mistakes (low false positives), effectively differentiating positive from negative cases. Despite this, the strong F1-score of 92% demonstrates a good balance between precision and recall. Additionally, the exceptional AUC score of 97.25% confirms the model's excellent ability to distinguish positive from negative data. In summary, the model performs well, but there's room to improve recall for capturing more positive cases effectively. First, I evaluated the performance of individual DL algorithms, without using the voting ensemble technique. Table 22 summarises this evaluation and provides a comparison of the proposed algorithms' performance. Figure 41 visually

shows the performance of the DL algorithms before applying the proposed Ensemble approach. Additionally, Figures 42 to 45 present the confusion matrix, offering valuable insights into the model's ability to correctly classify each class.

Table 22: Algorithms performance before applying the proposed Ensemble approach.

ML technique	Accuracy	Precision	Recall	F1 score	AUC
LSTM	0.9202	0.9259	0.9070	0.9164	0.9803
BiLSTM	0.9150	0.9054	0.9077	0.9065	0.9753
IDCNN-BiLSTM	0.9079	0.9281	0.8883	0.9077	0.9770
CNN	0.9174	0.9109	0.9235	0.9172	0.9816

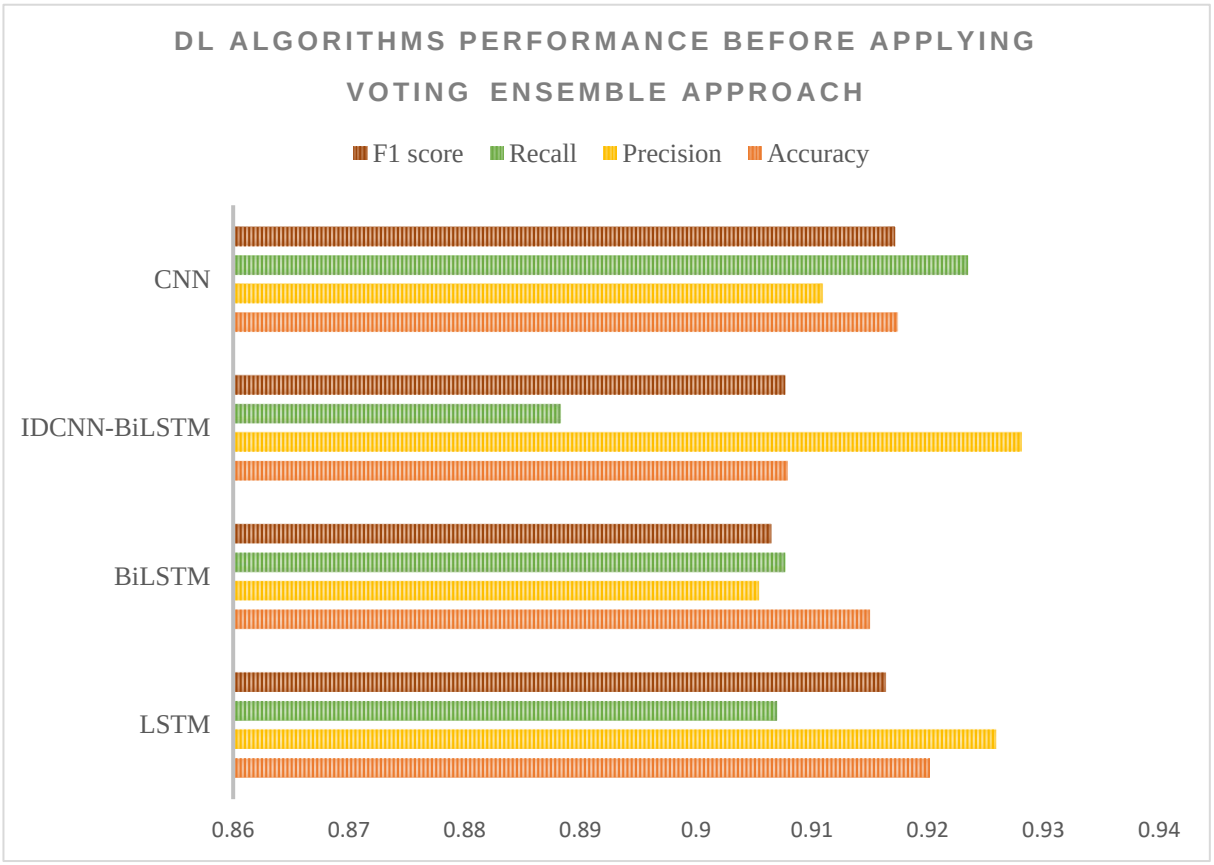
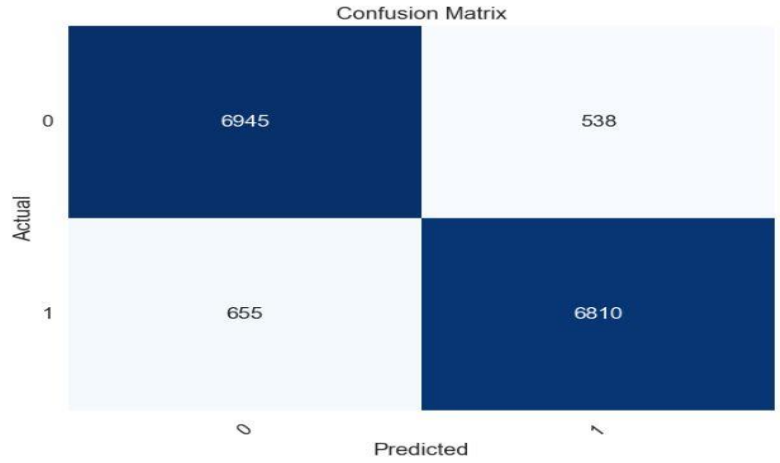


Figure 42: DL Algorithms performance before applying proposed voting Ensemble Approach



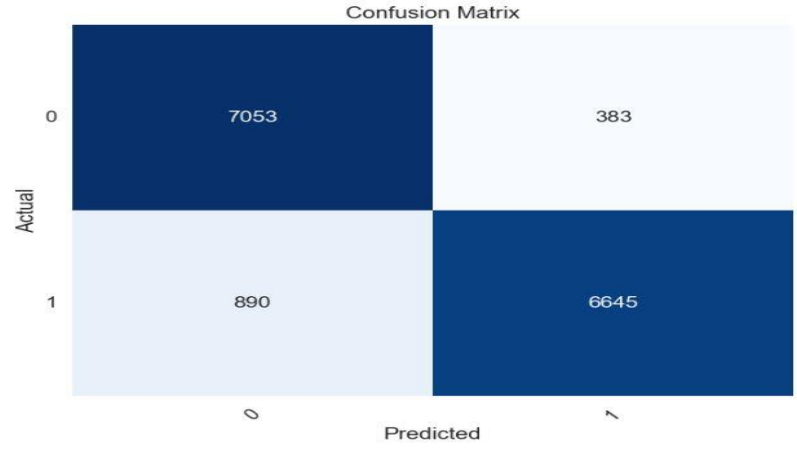

```
Classification Report:
              precision    recall  f1-score   support

           0       0.91      0.93      0.92      7483
           1       0.93      0.91      0.92      7465

    accuracy                           0.92     14948
   macro avg       0.92      0.92      0.92     14948
weighted avg       0.92      0.92      0.92     14948

468/468 [==============================] - 5s 10ms/step - loss: 0.1728 - accuracy: 0.9202
Test Accuracy: 0.9201899766921997
```

Figure 43: LSTM confusion matrix



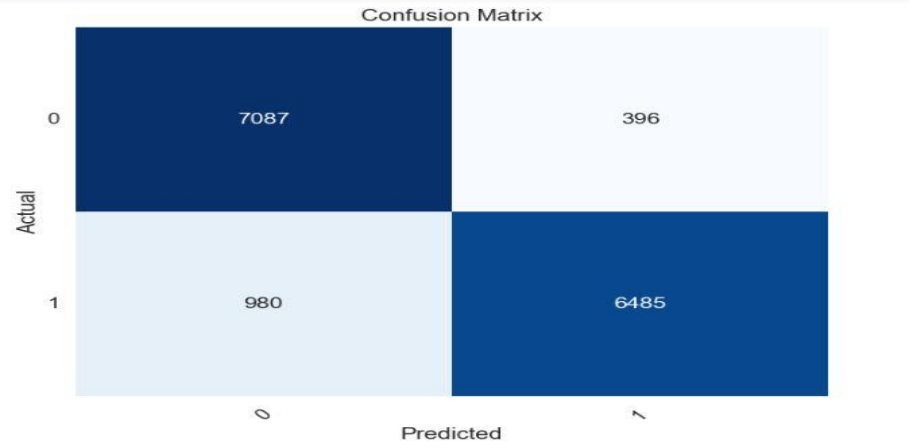

```
Classification Report:
              precision    recall  f1-score   support

           0       0.89      0.95      0.92      7436
           1       0.95      0.88      0.91      7535

    accuracy                           0.91     14971
   macro avg       0.92      0.92      0.91     14971
weighted avg       0.92      0.91      0.91     14971

468/468 [==============================] - 6s 13ms/step - loss: 0.1960 - accuracy: 0.9150
Test Accuracy: 0.9149689674377441
```

Figure 44: BLSTM confusion matrix



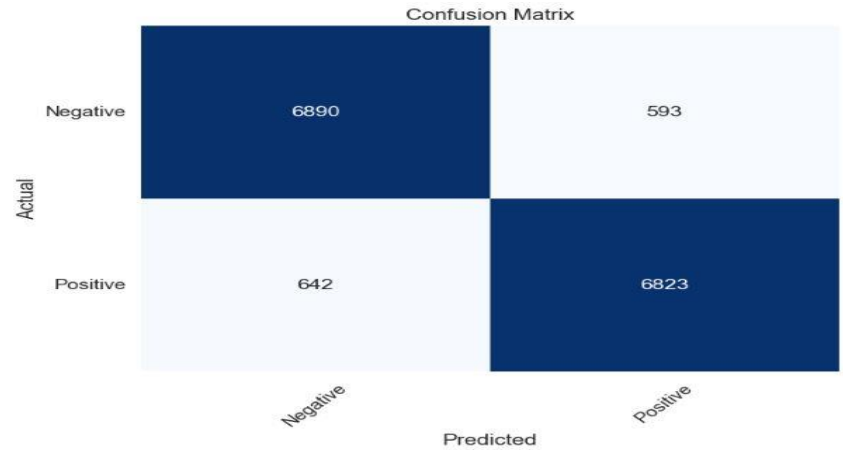

```
Classification Report:
              precision    recall  f1-score   support

           0       0.88      0.95      0.91      7483
           1       0.94      0.87      0.90      7465

    accuracy                           0.91     14948
   macro avg       0.91      0.91      0.91     14948
weighted avg       0.91      0.91      0.91     14948

468/468 [==============================] - 7s 14ms/step - loss: 0.2071 - accuracy: 0.9079
Test Accuracy: 0.9079475402832031
```

Figure 46: BiLSTM confusion matrix




```
Classification Report:
              precision    recall  f1-score   support

    Negative       0.91      0.92      0.92      7483
    Positive       0.92      0.91      0.92      7465

    accuracy                           0.92     14948
   macro avg       0.92      0.92      0.92     14948
weighted avg       0.92      0.92      0.92     14948

468/468 [==============================] - 1s 2ms/step - loss: 0.1649 - accuracy: 0.9174
Test Loss: 0.1649225354194641
Test Accuracy: 0.9173802733421326
```

Figure 45: CNN confusion matrix.

5.3.2 Comparison between DL techniques after voting ensemble approach

Ensemble methods in DL are an important strategy used to improve the predictive capabilities of a model by leveraging the strengths of multiple individual models. These methods are based on the understanding that while individual models may have limitations and biases, combining their predictions can produce more accurate and robust results. By aggregating the predictions of different models, ensemble methods leverage the collective intelligence of the ensemble to produce results that are better than those of any single model acting alone. The concept underlying ensemble methods is like the "wisdom of crowds," where the combined judgment of a group of people tends to be more accurate than the judgment of any individual member of the group. In the context of DL, this principle can be translated into the idea that the collective decisions of multiple models are usually more reliable than the decisions of any single model. This is achieved through ensemble techniques by considering different perspectives, learning patterns and capturing different aspects of the data, leading to a more comprehensive understanding of the underlying relationships.

Ensemble techniques are used in a variety of tasks in the areas of classification and regression. In classification tasks, ensembles can effectively handle complex decision boundaries, improve the classification of minority classes, and reduce the impact of noisy or conflicting data points. In regression tasks, ensembles can provide more robust predictions by reducing the impact of outliers and improving the generalisation performance of the model. One of the main advantages of ensemble techniques is their versatility and adaptability to different problem scenarios and data sets. They can accommodate different types of base models, from simple models like decision trees to more complex models like neural networks. Ensemble techniques can also be tailored to address specific challenges such as overfitting, bias, and imbalanced data distribution. Moreover, ensemble techniques provide a practical approach to the model combination that allows multiple models to be seamlessly integrated into a coherent framework. This integration can take many forms, including averaging predictions, combining probabilities, and training a meta-learner to learn the optimal combination of base model outputs.

Overall, ensemble techniques can be a powerful weapon in the DL toolkit, allowing users to leverage the collective intelligence of different models to improve predictive performance. As DL continues to advance and datasets become more complex, ensemble techniques will continue to be essential to achieve state-of-the-art results in a variety of domains and applications.

Voting is an ensemble technique that combines predictions from multiple models using simple strategies such as majority voting (for classification) or averaging (for regression). It is easy to implement and works well when the base models are different but complementary. Voting can be further categorised into hard voting, where the final prediction is based on the majority vote of the base models, and soft voting, where the predictions are weighted based on their confidence values.

The choice of Voting as the ensemble technique for the novel Ensemble model stems from its inherent advantages over other ensemble methods. Voting allows for the combination of diverse base classifiers, each capturing unique aspects of the data, leading to a more comprehensive understanding of the underlying patterns. By leveraging the collective intelligence of multiple models, Voting synthesises a robust predictive model that is less prone to individual model biases and overfitting. Moreover, Voting fosters a collaborative synergy among constituent algorithms, enabling them to complement each other's strengths and mitigate weaknesses.

This collaborative nature empowers the ensemble model to navigate the complex landscape of cyber threat detection with heightened precision and accuracy. Additionally, Voting facilitates continuous learning and adaptation, allowing the model to evolve in response to emerging cyber threats patterns and evolving threats, thereby enhancing its resilience in real-world scenarios, also A voting ensemble Deep Learning model significantly outperforms Stacking and other ensemble methods in detecting cyber threats, as measured by F1 score as previously mentioned in Section 4.2.5.

- **Findings and discussion**

The performance of various machine learning techniques in the given results highlights the efficacy of the Proposed VEDLM model, which significantly outperforms the other methods across all evaluation metrics.

1. Accuracy: The Proposed VEDLM achieved the highest accuracy of 0.9551, indicating its superior overall performance in correctly classifying instances compared to other models. This suggests that the VEDLM model is more reliable in predicting outcomes accurately across the dataset.
2. Precision: The precision of the Proposed VEDLM is notably high at 0.9726, surpassing all other techniques. This implies that the model has a lower rate of false positives, making it particularly effective in ensuring that the positive predictions are indeed correct. In practical applications, this high precision is critical where the cost of false positives is substantial.
3. Recall: With a recall of 0.9287, the Proposed VEDLM also shows strong performance in identifying positive instances. While not the highest recall (which is 0.9235 from the CNN model), the recall for VEDLM is still robust and indicates that the model does not miss many positive cases, balancing well with precision.
4. F1 Score: The F1 score, which is a harmonic mean of precision and recall, is highest for the Proposed VEDLM at 0.9502. This reflects an optimal trade-off between precision and recall, further underscoring the model's balanced performance in classification tasks.

By comparison, the LSTM and BiLSTM models show lower performance metrics across all measures, with LSTM having the highest F1 score of 0.9164 but still falling short of the Proposed VEDLM. The IDCNN-BiLSTM model, while performing well in precision, does not match the VEDLM in the overall F1 score, suggesting it might struggle with balancing precision and recall. The CNN model, although providing a good balance and performing better than LSTM and BiLSTM in recall, still does not reach the level of precision and F1 score achieved by VEDLM.

In summary, the Proposed VEDLM demonstrates superior performance in all key metrics, making it the most effective model among those tested. Its high accuracy, precision, recall, and F1 score suggest that it is the most balanced and reliable model for the task, providing an excellent combination of correct classifications and minimal misclassifications.

The analysis of the obtained results indicates that the proposed Ensemble approach employed majority voting to consolidate predictions from LSTM, BiLSTM, IDCNN-BiLSTM, and CNN models will improve the accuracy of the algorithm. By

comparing and analysing the obtained results, we can draw the following conclusions as presented in Table 23: Performance results after applying the proposed Ensemble approach; Figure 46 models the results after applying the proposed Ensemble approach; and Figure 47: F1 scores for DL techniques after applying the proposed voting Ensemble deep learning (VEDLM) approach.

Table 23: Performance results after applying the proposed voting ensemble approach

DL technique	Accuracy	Precision	Recall	F1 score
LSTM	0.9202	0.9259	0.9070	0.9164
BiLSTM	0.9150	0.9054	0.9077	0.9065
IDCNN-BiLSTM	0.9079	0.9281	0.8883	0.9077
CNN	0.9174	0.9109	0.9235	0.9172
Proposed VEDLM	0.9551	0.9726	0.9287	**0.9502**

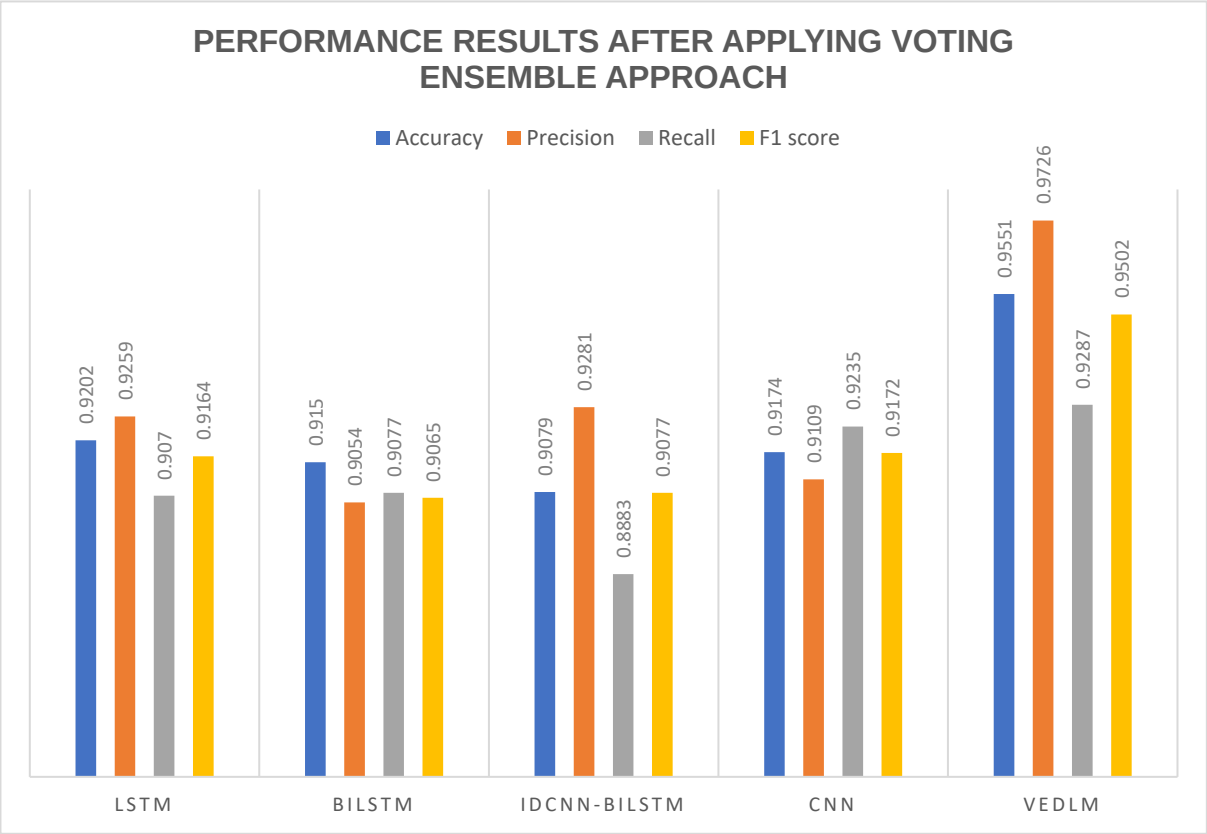


Figure 47: Performance results after applying voting Ensemble approach

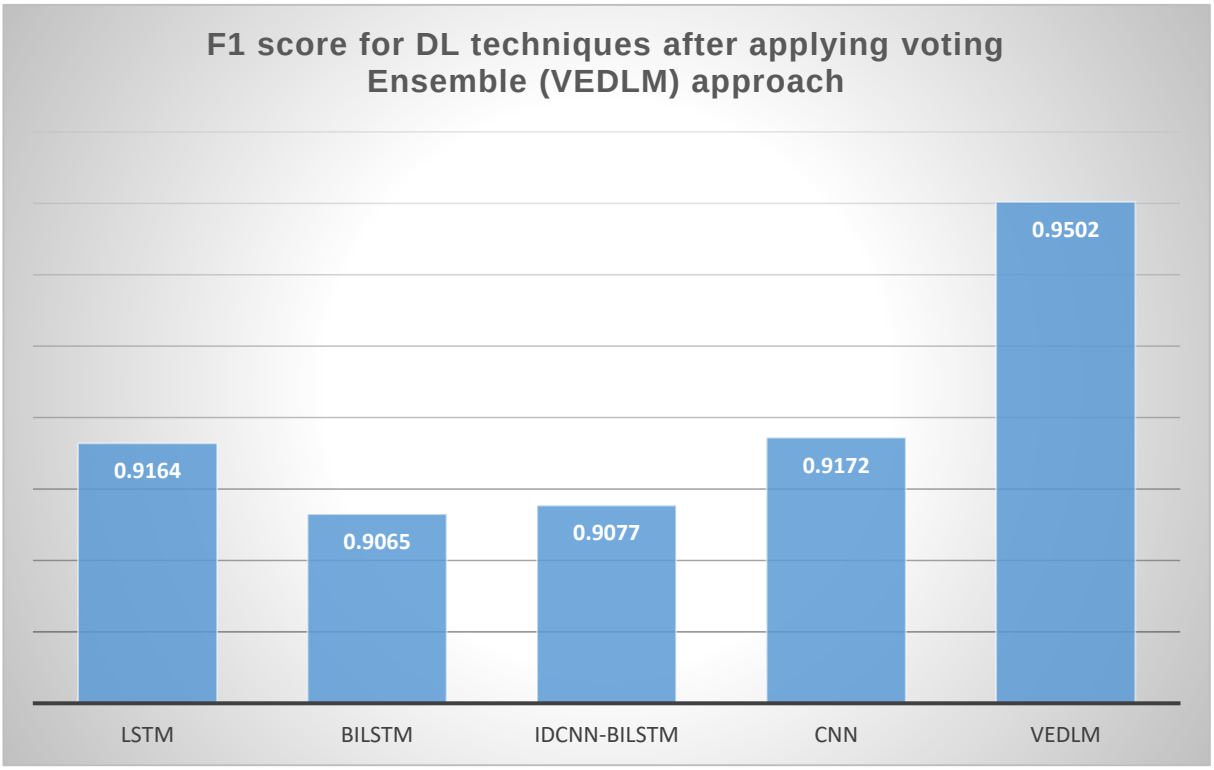
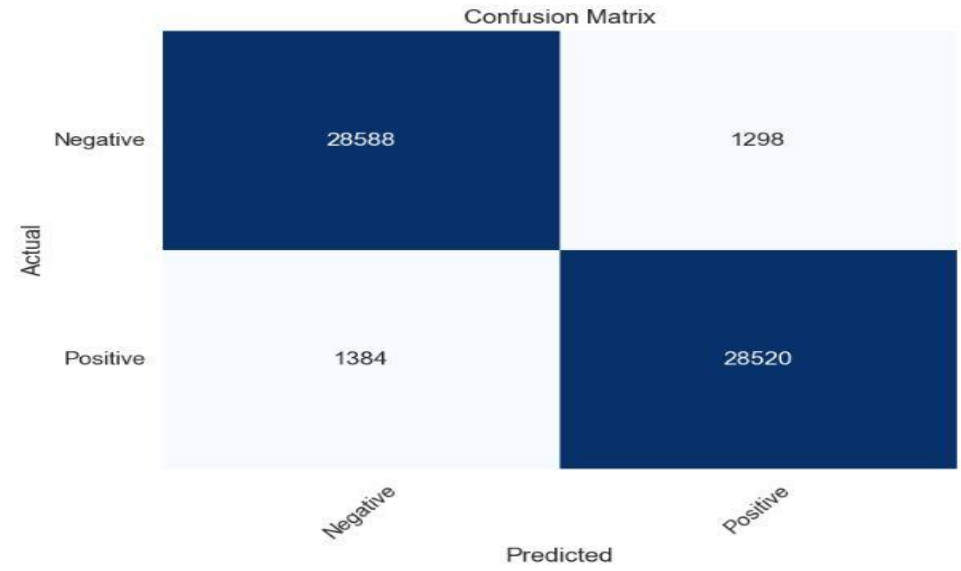


Figure 48: F1 score for DL techniques after applying the proposed Voting Ensemble (VEDLM) approach

The proposed VEDLM model excelled in identifying cyber threats, achieving a higher accuracy of 95.51% and a high precision of 96%. This means the model can accurately classify threats with minimal false positives, crucial for minimising losses from cyber threats. In summary, the empirical findings underscore the pivotal role of the threats proposed VEDLM model in advancing the frontier of cyber threats detection. Through its superior performance, as evidenced by the higher F1 score compared to all individual algorithms, the proposed ensemble approach offers a potent tool for enhancing predictive accuracy and resilience in cyber threat detection systems, thereby addressing the evolving challenges of cyber threat detection. Figure 48 presents the confusion matrix for the proposed VEDLM, offering valuable insights into the model's ability and performance.




```
Classification Report:
              precision    recall  f1-score   support

    Negative       0.95      0.96      0.96     29886
    Positive       0.96      0.95      0.96     29904

    accuracy                           0.96     59790
   macro avg       0.96      0.96      0.96     59790
weighted avg       0.96      0.96      0.96     59790

1869/1869 [==============================] - 32s 17ms/step
1869/1869 [==============================] - 33s 17ms/step
1869/1869 [==============================] - 3s 2ms/step
Ensemble Model Accuracy: 0.9551430005017562
```

Figure 49: Confusion Matrix for the proposed VEDLM model

5.4 Chapter summary

In summary, experimental findings underscore the proposed voting ensemble deep learning model (VEDLM) capability to outperform other individual techniques, showcasing an F1 score of 96%. These results highlight the importance of using the voting ensemble technique to improve the accuracy and reliability of cyber threat detection systems.

The chapter elaborated on the successful new design of an advanced DL model tailored for cybersecurity threat detection. It utilised a diverse set of DL algorithms, including LSTM, BiLSTM, IDCNN-BiLSTM, and CNN. These algorithms were selected based on their diverse nature and proven effectiveness in handling various types of data and modelling complexities.

CHAPTER 6 : COMPARATIVE ANALYSIS

Ensemble Learning draws inspiration from a fundamental human behaviour: consulting multiple experts to make informed decisions, especially in situations with significant financial, social, or medical implications (Mousavi & Eftekhari 2015). This approach is widely used in fields such as online learning, incremental learning, data fusion, feature selection, and confidence estimation. Ensemble learning is particularly effective in improving the generalisation ability of classification models. Based on the concept of 'The Wisdom of Crowds,' an ensemble model combines multiple weak learners to create a single, more accurate predictive model. This chapter presents a comprehensive comparative analysis focused on two key aspects: first, it examines the proposed Subspace Random Ensemble Machine Learning Model (SREMLM) and Voting Ensemble Deep Learning Model (VEDLM) models, highlighting their distinct characteristics, methodologies, and performance metrics to showcase their individual strengths and limitations. Second, it compares the VEDLM model with related works in the literature, assessing its innovations and performance relative to existing models. Through this analysis, the chapter aims to provide a clear understanding of the proposed models' relevance, their contributions to the field, and their potential impact on future research.

6.1 Comparative analysis: SREMLM and VEDLM on the same dataset

This section will present a detailed examination of the Subspace Random Ensemble Machine Learning Model (SREMLM) and Voting Ensemble Deep Learning Model (VEDLM) models. We will outline the unique characteristics, methodologies, and performance metrics of each model, highlighting their strengths and potential limitations. The goal is to provide a clear understanding of how these models, offering insights into their effectiveness.

To evaluate the overall performance of deep learning compared to traditional machine learning methods, we can analyse key metrics such as Accuracy, Precision, Recall, and F1 score, focusing on the performance of two Chapter 4,5 models: Subspace Random Ensemble Machine Learning Model (SREMLM) and Voting Ensemble Deep Learning Model (VEDLM).

1. Accuracy

In terms of accuracy, the Subspace Random Ensemble outperforms the Voting Ensemble by a small margin (0.9634 vs. 0.9551). This suggests that in terms of correctly classifying all instances, the Subspace Random Ensemble has a slight edge. However, accuracy alone doesn't provide a complete picture, especially if the dataset is imbalanced.

2. Precision

Precision measures the ability of the model to correctly identify positive instances among the instances labelled as positive. The Voting Ensemble significantly outperforms the Subspace Random Ensemble in precision (0.9726 vs. 0.90). This indicates that the Voting Ensemble is much better at reducing false positives, making it more reliable when the cost of false positives is high.

3. Recall

Recall reflects the ability of the model to identify all relevant instances (true positives). The Voting Ensemble also outperforms the Subspace Random Ensemble in this aspect (0.9287 vs. 0.90). A higher recall means that the Voting Ensemble is better at minimising false negatives, which is crucial when missing a positive instance has significant consequences.

4. F1 Score

The F1 score is the harmonic mean of precision and recall, providing a balance between the two. The Voting Ensemble, with an F1 score of 0.9502, is superior to the Subspace Random Ensemble's F1 score of 0.90. This suggests that the Voting Ensemble achieves a better trade-off between precision and recall, making it a more robust model overall.

Figure 49 is a Comparative Analysis of Model Performance in Chapters 4 and 5.

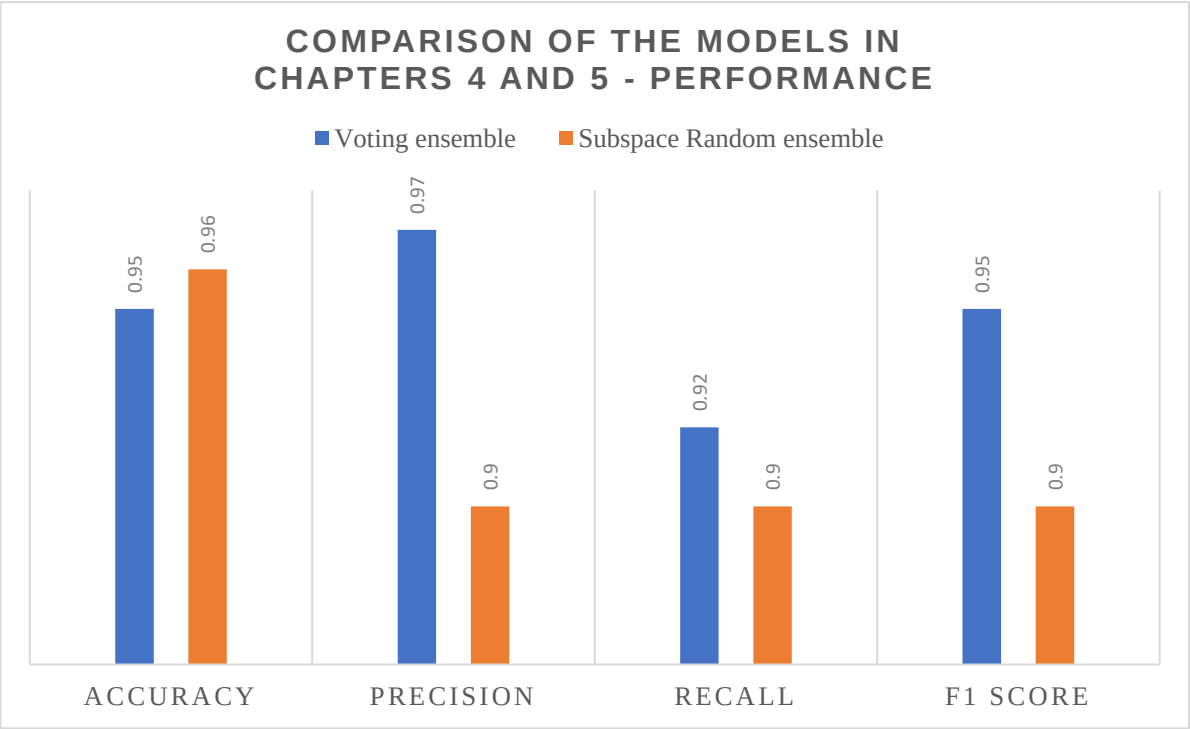


Figure 50: Comparison of the models in Chapter 4 and 5 - Performance.

In conclusion, while the Subspace Random Ensemble Machine Learning Model (SREMLM) achieves slightly higher accuracy, the Voting Ensemble Deep Learning Model (VEDLM), which may incorporate deep learning techniques, demonstrates superior performance in precision, recall, and the F1 score. These metrics are often more critical in real-world applications where the balance between false positives and false negatives can significantly impact outcomes.

Given the higher precision, recall, and F1 score, we can conclude that the overall performance of the Voting Ensemble Deep Learning Model (VEDLM) is better than the Subspace Random Ensemble Machine Learning Model (SREMLM), showcasing the advantage of using deep learning approaches in this filed.

6.2 Comparing VEDLM to related works

In this section, the VEDLM model will be juxtaposed with similar models and frameworks from existing literature using the dataset presented in section 6.2.1. By comparing the VEDLM model against established works, we aim to identify its innovative contributions, assess its performance in various contexts, and determine its standing in the current landscape of language modelling.

6.2.1 Dataset

The dataset in this chapter comprises 60,000 tweets, each labelled as either threats (1) or non-threats (0). The project involved preprocessing the text data, balancing the dataset, and implementing several Deep Learning models to achieve high classification results. The Dataset, containing a collection of tweets, is available at the following link: https://www.kaggle.com/datasets/syedabbasraza/suspicious-tweets/data

The dataset consists of 60,000 tweets with the following distribution, additionally, Figure 50 presents the number of each class for the open-source Dataset:

- **Total tweets**: 60,000
- **Threat tweets (label 1):** 53,855
- **Non-threat tweets (label 0):** 6,145

The dataset initially included two columns: **comments** and **labels**.

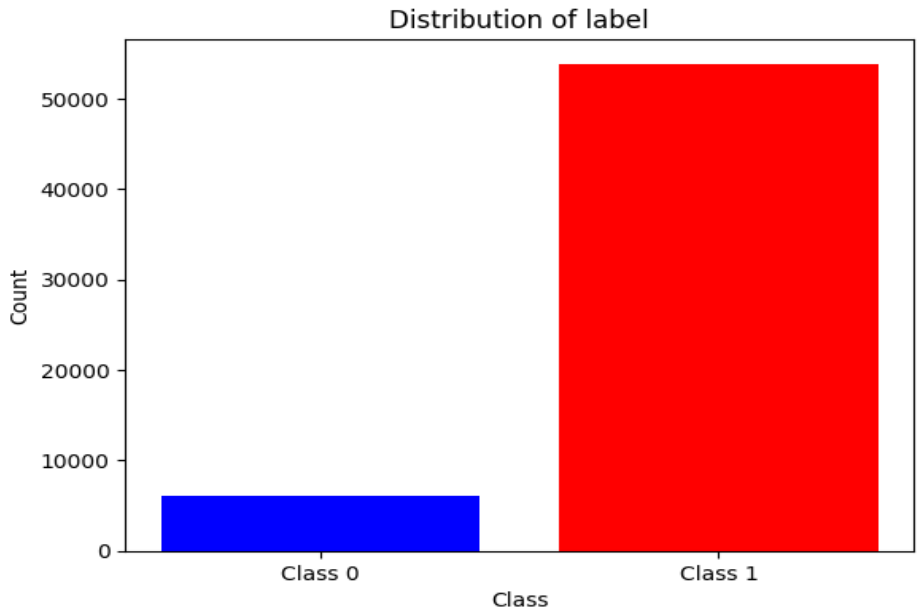


Figure 51: Number of each class for the open-source dataset

A. Dataset preprocessing

To prepare the text data for rigorous analysis, I conducted a thorough cleaning process. This involved standardising formats and eliminating extraneous information

to improve data quality, Additionally, duplicate tweets were identified and removed, reducing the open-source Dataset to 59,125 unique entries. The dataset preprocessing steps in this chapter are similar to those outlined in Section 4.1.3.

B. Dataset word clouds

Word clouds are an effective visual tool for highlighting the most frequently occurring words in text data. They can provide valuable insights, especially when analysing tweets categorised as threats or non-threats, as demonstrated in Figure 51.

Figure 52: Word Clouds for the open-source dataset

C. Dataset balancing

To address the class imbalance in the dataset, Synthetic Minority Over-sampling Technique (SMOTE) was employed to generate synthetic threat tweets, thereby matching the number of threat and non-threat tweets. After applying SMOTE, both classes were equally represented, with the dataset size exceeding 50,000 tweets per class as depicted in Figure 52.

Figure 53: Data Balancing for open-source dataset

D. The Train-Test Split datasets

Building upon the data preparation methodology established in Chapter 4, Section 4.1, I employed the Train-Test Split technique on the dataset introduced in Section 6.1. This approach is fundamental to rigorous model evaluation, as it enables us to assess a model's ability to generalise to unseen data. By partitioning the dataset into distinct training and testing subsets, I can iteratively refine the model on the training data while objectively measuring its performance on the previously unseen test data. This consistent evaluation framework across different datasets provides a robust foundation for comparative analysis and a deeper understanding of model behaviour in diverse contexts.

E. Cross Validation (CV)

Cross Validation (CV) is a robust statistical method used to assess the generalisation ability of a model. It involves partitioning the dataset into multiple subsets (or "folds") and iteratively training and testing the model on different combinations of these folds. The most common type of CV is k-fold CV (Btoush et al. 2023).

In this section, we explore the use CV in training and evaluating a proposed VEDLM model as well. The focus is on understanding the principles of CV, how it was implemented in the code, and the methodology for calculating and interpreting the performance metrics.

- **Stratified k-Fold Cross Validation**

The provided code uses Stratified k-Fold Cross Validation with n_splits=5. Unlike regular k-fold CV, stratified k-fold ensures that each fold is representative of the overall distribution of classes. This is particularly important for imbalanced datasets, as it prevents any fold from being skewed toward a particular class.

- **Implementation in the Provided Code**

1. **Data Preparation**

Before applying CV, several preprocessing steps were performed:

- **Text Vectorisation:** The *TfidfVectoriser* was employed to convert text data into numerical features. The vectoriser transforms the text data into a sparse matrix of TF-IDF features.
- **Handling Imbalance:** SMOTE (Synthetic Minority Over-sampling Technique) was applied to the vectorised data to address class imbalance by generating synthetic samples for the minority class.

2. **Tokenisation and Padding**

- **Tokenisation:** The text data was tokenised into sequences of integers using the *Tokeniser* from TensorFlow.
- **Padding:** The sequences were then padded to ensure uniform input length, which is necessary for feeding into the LSTM model.

3. **Cross Validation Workflow**

The cross-validation process was set up using *StratifiedKFold*, and the workflow involved the following steps:

1. Data Splitting: The dataset was divided into 5 folds.

2. Model Training: For each fold, the model was trained on 4 folds (training set) and validated on the remaining fold (test set).
3. Model Evaluation: Post-training, the model was evaluated on the validation set, and various performance metrics were calculated.

- **Performance Metrics Calculation**

For each fold in the CV, the following metrics were calculated:

1. **Accuracy and Loss**

- **Validation Accuracy**: The accuracy of the model on the validation set was recorded for each fold. The mean and standard deviation across all folds were calculated to provide a measure of the model's generalisation ability.
- **Validation Loss**: The loss function, *binary_crossentropy*, was used to measure the error. Similar to accuracy, the mean and standard deviation of the loss were computed.

2. **Confusion Matrix** The confusion matrix was computed to visualise the performance of the classification model by displaying the counts of true positives, true negatives, false positives, and false negatives.
3. **Classification Report** A detailed classification report was generated, which included precision, recall, F1-score, and support for each class. This classification report helps in understanding the performance of each class individually.

4. **AUC-ROC**

- AUC (Area Under the Curve) for the ROC (Receiver Operating Characteristic) Curve was calculated for each fold. AUC provides a single scalar value to compare the performance of different models, with a higher AUC indicating better model performance.
- The ROC curve itself was plotted for each fold, displaying the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings.

- **Cross-Validation Metrics**

- Mean Accuracy and loss

$$\text{Man Accuracy} = \frac{1}{n}\sum_{i=1}^{n} \text{Accuracy}_{\,i} \quad (18)$$

$$\text{Mean Loss} = \frac{1}{n}\sum_{i=1}^{n} \text{Loss}_{\,i} \quad (19)$$

Where n is the number of folds (in your case 5), and accuracy , and loss are the accuracy and loss for the i^{th} fold.

- Standard Deviation:

$$\text{Standard Deviation of accuracy} = \sqrt{\frac{1}{n}\sum_{i+1}^{n}[\text{Accuracy}_{\,i} - \text{Mean Accuracy}\,]^2} \quad (20)$$

$$\text{Standard Deviation of Loss} = \sqrt{\frac{1}{n}\sum_{i+1}^{n}[\text{Loss}_{\,i} - \text{Mean Loss}\,]^2} \quad (21)$$

- Number

  Mean Accuracy_acc = (calculated value)

  Standard Deviation of Accuracy:std_acc = (calculated value)
  Mean Loss: mean_loss = (calculated value)

  Standard Deviation of Loss:std_loss = (calculated value)

Stratified k-fold CV was employed to rigorously assess the model's ability to generalise to unseen data. By partitioning the dataset into multiple folds and iteratively training and evaluating the model, I obtained reliable performance estimates.

Key metrics included mean accuracy and loss across all folds, accompanied by standard deviations to measure consistency. Detailed visualisations such as confusion matrices, classification reports, and AUC-ROC curves for each fold provided granular insights into model behaviour. Overall, the model demonstrated robust performance, as evidenced by the consistent metrics and informative visualisations. This evaluation process instils confidence in the model's ability to effectively predict new, unseen data.

6.3 Finding

The proposed voting Ensemble Deep Learning Model (VEDLM) outperforms the state-of-the-art models in terms of accuracy and F1 score on the dataset presented in Section 6. 2.1. While the mentioned models in Section 4.1 achieve high accuracy individually, The Ensemble Deep Learning Model, which combines them through

voting, achieves the highest accuracy of 98.60% and a competitive F1 score of 99%. This suggests that the integration of individual DL techniques in a voting framework enhances the model's predictive capabilities, offering promising results for cyber threat detection.

This section provides a comprehensive comparison between the proposed voting Ensemble Deep Learning Model (VEDLM) and the six studies outlined in Section 2.3.3 detection of security content: SYNAPSE (Alves et al. 2021), DeepNN (Dionísio et al. 2019), DataFreq (Rodriguez & Okamura 2019), CyberTwitter (Mittal et al. 2016), Text-mining (Sapienza et al. 2017), and SONAR (Le Sceller et al. 2017). Using the dataset outlined in this chapter in Section 6.2.1. The detailed comparison of six previously introduced studies and the proposed voting Ensemble Deep Learning Model (VEDLM) aims to demonstrate that the proposed VEDLM in this study was trained and tested on the database described in Section 6.2.1. The results indicate that the proposed VEDLM exhibits high performance compared to state-of-the-art techniques and baseline models.

A. The algorithm

One perspective for comparison involves the use of algorithms. The most basic technique involves filtering tweets based on specific keywords. The Text-mining study by Sapienza et al. (2017) filters tweets using technical, security, and English dictionaries to extract unfamiliar terms. In CyberTwitter (Mittal et al. (2016), a Semantic Web Rule Language (SWRL) is utilised to issue alerts related to the user profile. SONAR Le Sceller et al. (2017) use cosine similarity as a clustering task for attack detection. They aim to train machines to behave like humans but differ in their learning models. For example, DataFreq Rodriguez and Okamura (2019) use logistic regression (LR) for supervised ML to classify tweet sentiment as positive or negative. DeepNN (Dionísio et al. (2019) use DL with CNN to classify tweets as relevant to security or not, with deeper learning as the number of layers increases. SYNAPSE Alves et al. (2021) utilises support vector machine (SVM) to predict tweet classification. While DL algorithms are highly sophisticated, they have the drawback of needing larger amounts of data to train the model and achieve desired outcomes. The proposed Voting Ensemble Deep Learning model uses multitask DL models (LSTM, BiLSTM,

IDCNN-BiLSTM, CNN) to classify tweets as relevant to security or not, with deeper learning as the number of layers increases. Table 24 organises the studies according to the algorithm used.

Table 24:Comparison of the Algorithms

Prediction methods	**Algorithm**
The proposed voting Ensemble Deep Learning model	***VEDLM (LSTM, BiLSTM, IDCNN - BiLSTM, CNN)***
SYNAPSE (Alves et al. 2021)	SVM
DeepNN (Dionísio et al. 2019)	CNN
DataFreq (Rodriguez & Okamura 2019)	LR
SONAR (Le Sceller et al. 2017)	Cosine similarity
Text-mining (Sapienza et al. 2017)	Filtering
CyberTwitter (Mittal et al. 2016)	SWRL

B. Degree of information summarisation

Due to the time-sensitive nature of cyber-attacks, real-time detection is an essential feature. Our research aims to achieve this goal with varying degrees of information presented to the analyst. Providing a summary of detected events/attacks saves time and effort for the analyst, as the increased amount of information presented can prolong the analysis and necessary action. Summarisation can be done through various techniques, one of which is the clustering of security-related tweets.

On one side, the detection of an attack is made more dependable, and it stops people from starting an attack event. Nonetheless, this volume of data may not be fitting for security areas like SONAR (Le Sceller et al. 2017). Moreover, alerts may be used to restrict data. For instance, CyberTwitter (Mittal et al. 2016) sends an alert and presents all relevant tweets based on the user's profile. Named Entity Recognition (NER) is used in DeepNN (Dionísio et al. 2019) for each tweet by means of NN, which enhances the quality of data extraction, even though the number of tweets is still high. In DataFreq (Rodriguez & Okamura 2019), the average sentiment of each company is displayed through a user-friendly interface. Clustering may be trailed by exemplar

extraction, leading to a tweet representing each cluster, such as SYNAPSE's work (Alves et al. 2021). Text-mining (Sapienza et al. 2017) showcases the discovered attack term, frequency, and context as the final outcome.

The Ensemble Deep Learning Model Achieve superior classification and Named Entity Recognition (NER) with our advanced deep learning ensemble. This method combines the strengths of multiple models to enhance accuracy and reliability in data prediction. Table 25 outlines the comparison of the studies in terms of the amount of information presented to the user.

Table 25: Comparison of the degree of information summarisation

Prediction methods	**Summarisation**
The proposed voting Ensemble Deep Learning model (VEDLM)	***Classification, NER***
Text-mining (Sapienza et al. 2017)	Summarised alert
SYNAPSE (Alves et al. 2021)	Clustering, exemplar
DataFreq (Rodriguez & Okamura 2019)	Sentiment score for each company
DeepNN (Dionísio et al. 2019)	Classification, NER
CyberTwitter (Mittal et al. 2016)	Detailed alert
SONAR (Le Sceller et al. 2017)	Clustering

C. Scalability and effectiveness

This section examines and compares security research schemes that are based on essential time properties, as demonstrated in Table 26. The research focuses on the rapidly evolving nature of security terminologies and specific attack types. To avoid the problem of manually collecting keywords, which can lead to forgetting certain words, researchers suggest searching for new words to update the list of security keywords used in X searches. For example, the SONAR (Le Sceller et al. 2017) system automatically discovers new related words and allows users to evaluate them manually. If the new words are relevant, they can be added to the list for future use. The GloVe word embedding technique is used to extract semantic relationships between words, which helps the model keep up with changes in the field. However, the system still relies on human decision-making, which can lead to mistakes. The

Ensemble Deep Learning model employs updated keywords, allowing it to adjust dynamically to new data and evolving patterns, making it more adaptable to changing environments.

Table 26: Comparison of scalability

Prediction methods	**Scalability**
The proposed voting Ensemble Deep Learning model VEDLM	***Updated keywords***
SONAR (Le Sceller et al. 2017)	Updated keywords
DataFreq ((Rodriguez & Okamura 2019)	Updated keywords
SYNAPSE (Alves et al. 2021)	Fixed keywords and accounts
DeepNN (Dionísio et al. 2019)	Fixed keywords and accounts
Text-mining (Sapienza et al. 2017)	Fixed accounts and dictionaries
CyberTwitter (Mittal et al. 2016)	Fixed user profile

The study DataFreq (Rodriguez & Okamura 2019) utilised TF and TF-IDF analysis to identify relevant keywords in retrieved tweets. While both methods focused on updating the keyword list, TF-IDF relied solely on word frequency and did not extract semantic features. The evaluation of X accounts used to retrieve the tweets was not fully assessed, hence a percentage was not assigned. Other studies (Sapienza et al. 2017), (Mittal et al. 2016), (Alves et al. 2021), and (Dionísio et al. 2019) did not consider this aspect and followed previous research. It is assumed that users manually added new keywords to the list and received a 50% rating for doing so.

This factor examines the metrics used to assess each study. In the security field, detecting all attacks is crucial, and detecting false positives is more acceptable than missing important alerts. The security attack detection model is a high recall model, as it measures the True Positive Rate (TPR), or the number of attacks detected by the model compared to the total actual attacks. Studies using ML algorithms can be compared using the same metric, with Alves et al. (2021) and Dionísio et al. (2019) achieving a TPR higher than 90, considered a reasonable degree of recall. DataFreq Rodriguez and Okamura (2019) achieved a recall of about 84% and a precision of 85%, consistent with previous works. Studies not using ML such as Sapienza et al. (2017), Mittal et al. (2016), and Le Sceller et al. (2017) evaluated accuracy by the

quality of generated alerts, similar to precision, where the number of true detected attacks is divided by the total alerts generated.

CyberTwitter (Mittal et al. (2016) had the highest precision at 86%, even with the label "maybe" considered negative. Text-mining (Sapienza et al. (2017) calculated an average evaluation by five annotators, with 84% correct detected terms. SONAR (Le Sceller et al. (2017) had only 23 relevant detected events out of 100 in the evaluation period.

The proposed Voting Ensemble Deep Learning Model (VEDLM) sets new benchmarks in predictive analytics, achieving an astounding F1 score of 99%, surpassing the performance of all models. These results indicate that the proposed VEDLM exhibits superior performance compared to state-of-the-art techniques and baseline models. The studies are arranged from best to lowest performance in Table 27.

Table 27: Evaluation of the performance of models

Prediction methods	**Performance**
The proposed voting Ensemble Deep Learning model VEDLM	***F1 score 99%***
DeepNN (Dionísio et al. 2019)	Recall 94%
SYNAPSE (Alves et al. 2021)	Recall 90%
DataFreq (Rodriguez & Okamura 2019)	Recall 84%
CyberTwitter (Mittal et al. 2016)	Precision 86%
Text-mining (Sapienza et al. 2017)	Precision 84%
SONAR (Le Sceller et al. 2017)	Precision 23%

D. SEMANTIC CHARACTERISTICS

Within this aspect, Table 28 compares different studies based on the type of feature used for classification, specifically whether semantic features or keyword, count, or frequency features were utilised. The focus was on the accuracy of the technique, with emphasis on the positive effects of including semantic features as they extract both the content and meaning of the tweet. Some studies, such as Text-mining

(Sapienza et al. (2017), did not use semantic features, instead utilising dictionaries to filter unfamiliar terms.

Others, such as those by Mittal et al. (2016), and Le Sceller et al. (2017), used keyword-based X searches to identify security-related tweets. DataFreq Rodriguez and Okamura (2019) used n-gram to extract sentiment while SYNAPSE Alves et al. (2021) used TF-IDF to transform tweets into numerical values. However, simple textual and frequency-based feature extraction solutions may not accurately represent subtle semantic differences between real and false event mentions. In contrast, the word2vec technique used by DeepNN Dionísio et al. (2019) transforms each tweet into vectors and can effectively detect similarities between words. The Ensemble Deep Learning model utilises TF-IDF (Term Frequency-Inverse Document Frequency).

Table 28: Comparison of semantic characteristics

Prediction methods	**Utilised characteristics**
The proposed voting Ensemble Deep Learning model VEDLM	***TF-IDF***
DeepNN (Dionísio et al. 2019)	Word2vec
SYNAPSE (Alves et al. 2021)	TF-IDF
DataFreq (Rodriguez & Okamura 2019)	N-gram
SONAR (Le Sceller et al. 2017)	Keyword-based
CyberTwitter (Mittal et al. 2016)	Keyword-based
Text-mining (Sapienza et al. 2017)	Simple filtering

6.4 Discussion

The proposed voting Ensemble Deep Learning Model (VEDLM) significantly outperforms state-of-the-art models in both accuracy and F1 score on the same dataset. Although the individual models discussed in section 5.1 achieve high accuracy on their own, the proposed VEDLM, which integrates these models through a voting mechanism, achieves the most competitive F1 score of 99%. This indicates that combining individuals' deep learning techniques within a voting framework enhances the model's predictive capabilities, providing promising results for cyber threat detection.

This discussion offers an in-depth comparison between the proposed VEDLM, and six studies previously mentioned in section 2.3.3, focused on security content detection: SYNAPSE Alves et al. (2021), DeepNN Dionísio et al. (2019), DataFreq Rodriguez and Okamura (2019), CyberTwitter (Mittal et al. 2016), Text-mining Sapienza et al. (2017), and SONAR Le Sceller et al. (2017) All models were evaluated using the dataset outlined in section 6.1.

The SONAR Le Sceller et al. (2017) model leverages cosine similarity for its predictions. Despite the common use of cosine similarity in text and vector space analysis, the model exhibits a relatively low precision of 23%. This suggests that while the model can identify true positives, it also generates a significant number of false positives. Consequently, SONAR might be better suited for applications where the primary goal is recall rather than precision, or where further filtering steps can be applied to reduce false positives.

The text-mining Sapienza et al. (2017) approach uses filtering techniques to achieve a precision of 84%. This high precision indicates that the model effectively minimises false positives. Filtering techniques often involve rules or patterns to exclude irrelevant or erroneous data, making this model suitable for applications requiring high accuracy in positive predictions, such as specific keyword extraction or targeted information retrieval.

The CyberTwitter Mittal et al. (2016) model utilises SWRL to achieve a precision of 86%. SWRL is a powerful tool for representing and reasoning with rules on the Semantic Web. The high precision rate indicates the model's effectiveness in applying semantic rules to filter and identify relevant information accurately. This makes CyberTwitter particularly useful for scenarios requiring precise data extraction from large datasets, such as monitoring cyber threats or sentiment analysis.

DataFreq Rodriguez and Okamura (2019) employ logistic regression to achieve a recall of 84%. Logistic regression is a statistical method commonly used for binary classification problems. A recall of 84% suggests that the model effectively identifies a large proportion of true positives. This makes it useful in applications where missing true positives is costly, such as in fraud detection or medical diagnosis.

SYNAPSE Alves et al. (2021) utilise an SVM to achieve a recall of 90%. SVMs are known for their robustness in high-dimensional spaces and their effectiveness in classification tasks. A recall of 90% indicates the model's proficiency in capturing true positives, making it ideal for critical applications where it is essential to identify as many relevant instances as possible, such as in image recognition or bioinformatics.

In summary, SONAR's (Le Sceller et al. (2017) model achieved a precision of 23% using the cosine similarity algorithm. This low precision suggests that SONAR's effectiveness in identifying relevant instances was relatively limited compared to other models. Text-mining Sapienza et al. (2017) - by contrast, the Text-mining model demonstrated a significantly higher precision of 84% through filtering techniques. This indicates a more refined ability to accurately classify relevant data. CyberTwitter Mittal et al. (2016) achieved an impressive precision of 86% utilizing SWRL. This performance highlights its robust capability in precise classification. DataFreq Rodriguez and Okamura (2019): focusing on recall, DataFreq attained a recall rate of 84% with logistic regression (LR). While its precision was not measured, the recall rate suggests a strong ability to identify relevant instances. SYNAPSE Alves et al. (2021) achieved a higher recall of 90% using support vector machines (SVM). This indicates a superior capacity to retrieve relevant instances compared to DataFreq. DeepNN Dionísio et al. (2019) DeepNN model reached an even higher recall of 94% through convolutional neural networks (CNN). This suggests an excellent performance in identifying relevant data points. Proposed VEDLM (Ensemble Deep Learning Model): the proposed voting Ensemble Deep Learning Model (VEDLM) model stands out with an F1 score of 99%, achieved through a sophisticated ensemble of LSTM, BiLSTM, IDCNN with BiLSTM, and CNN. This superior F1 score of 99% reflects both high precision and recall, indicating a well-balanced and highly effective model.

The proposed voting Ensemble Deep Learning Model (VEDLM) stands out as a powerful approach for cyber threat detection. This chapter meticulously compared the proposed VEDLM to various established techniques. The results are promising: the proposed VEDLM, trained and tested on a consistent dataset, achieves an impressive accuracy of 98.60% and an F1 score of 99%. This signifies the model's effectiveness in finding cyber threats. This study paves the way for further exploration of ensemble methods in deep learning, suggesting their potential to surpass traditional

approaches and establish a new standard for cyber threat detection. Figure 53 shows a comparison of the effectiveness of models based on the model performance.

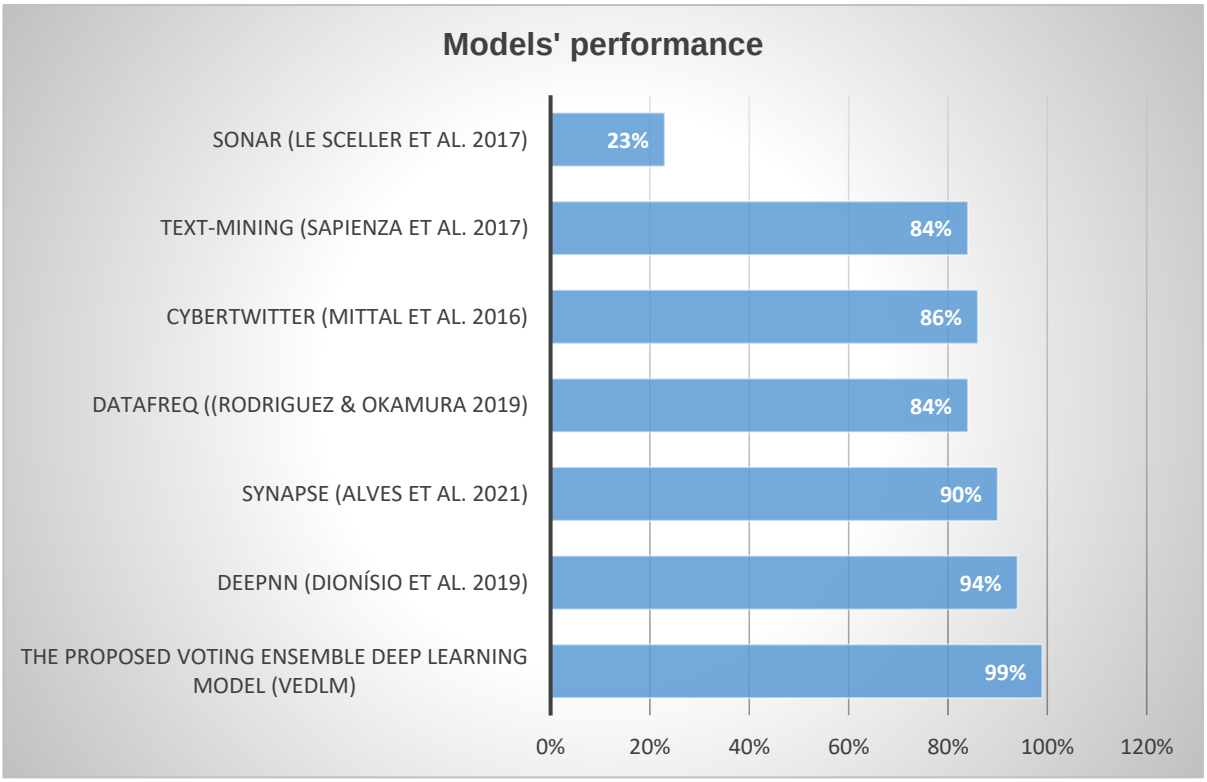


Figure 54: Comparison of the effectiveness of models based on the models' performance

In summary, while earlier models demonstrate varying degrees of precision and recall, the proposed VEDLM model surpasses them with its exceptionally high F1 score, representing a significant advancement in cyber threat detection. Further, Table 29 presents a comparative analysis of the proposed VEDLM model's performance with related studies, ranging from low to high performance levels.

Table 29: Comparison of proposed VEDLM model performance with related studies.

Study Ref.	**Model performance**	**Algorithm/ Model**
Le Sceller et al. (2017)	Precision 23%	Cosine similarity
Arora, Sharma and Khatri (2019)	Accuracy 80%	RF
Coyac-Torres et al. (2023)	Accuracy 82%	CNN
(Behzadan et al. 2018)	F1-score 82%	CNN
Sapienza et al. (2017)	Precision 84%	Filtering
Rodriguez and Okamura (2019)	Recall 84%	LR
Ghankutkar et al. (2019)	Accuracy 85%	SVM, MNB, and RF
Mittal et al. (2016)	Precision 86%	SWRL
Deshmukh et al. (2022)	F1-score 87%	SVM, RF, DT, XGBoost and AdaBoost

Mahaini and Li (2021)	F1-score 90%	DT, RF, SVM, and LR
Alves et al. (2021)	Recall 90%	SVM
Dionísio et al. (2019)	Recall 94%	CNN
The proposed voting Ensemble Deep Learning model	***F 1 score 99%***	***VEDLM (LSTM, BiLSTM, IDCNN -BiLSTM, CNN)***

The table provides a comprehensive overview of the performance metrics of various models and algorithms across several studies focused on different tasks. Precision, accuracy, recall, and F1-score are the primary metrics evaluated.

The SONAR model by SONAR Le Sceller et al. (2017) achieved a relatively low precision of 23% using cosine similarity, indicating its limited effectiveness in distinguishing relevant items from irrelevant ones. In contrast, the text-mining approach by Sapienza et al. (2017) demonstrated a substantially higher precision of 84% through filtering techniques.

Accuracy-wise, the models by Arora, Sharma and Khatri (2019) and Coyac-Torres et al. (2023) achieved 80% and 82%, respectively, with the former employing a Random Forest (RF) algorithm and the latter a Convolutional Neural Network (CNN). Ghankutkar et al. (2019) reported an even higher accuracy of 85% using a combination of Support Vector Machine (SVM), Multinomial Naive Bayes (MNB), and RF algorithms.

The recall metric highlighted the efficacy of models in retrieving relevant instances. DataFreq Rodriguez and Okamura (2019) achieved 84% recall using Logistic Regression (LR), whereas SYNAPSE Alves et al. (2021) and DeepNN Dionísio et al. (2019) models demonstrated superior performance with recalls of 90% and 94%, respectively, employing SVM and CNN algorithms.

F1-score, a balanced metric combining precision and recall, showed significant variation among the models. Behzadan et al. (2018) and Deshmukh et al. (2022) reported F1-scores of 82% and 87% with CNN and an ensemble of SVM, RF, Decision Tree (DT), XGBoost, and AdaBoost, respectively. Mahaini and Li (2021) achieved an impressive range of 88-91% using a diverse set of algorithms including DT, RF, SVM, and LR. Notably, the Ensemble Deep Learning model by incorporating Long Short-Term Memory (LSTM), Bidirectional LSTM (BiLSTM), IDCNN-BiLSTM, and CNN, outperformed all with a remarkable F1-score of 99%.

In conclusion, the proposed ensemble deep learning model consistently outperformed the machine learning models across various metrics, highlighting their advanced capabilities in handling complex, big datasets and tasks. This trend suggests a clear preference and effectiveness of sophisticated neural network architectures in achieving high-performance standards in data-intensive applications, Figure 54 presents the Models Performance Evaluation: A Comparative Analysis.

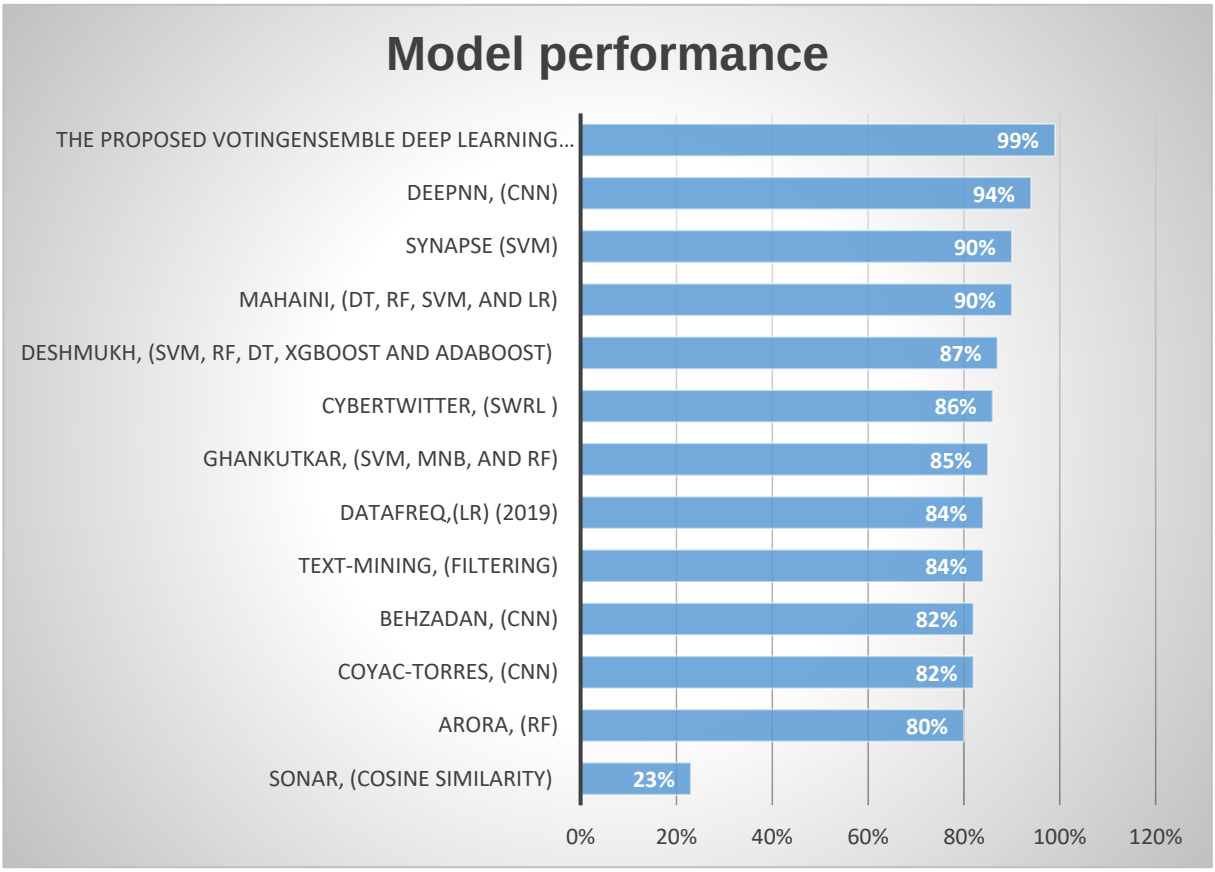


Figure 55: Related works, model Performance: a comparative analysis

6.5 Chapter summary

Although the proposed SREMLM has a slight edge in accuracy, the proposed VEDLM, likely leveraging deep learning, consistently surpasses it in precision, recall, and F1 score. These metrics are often more critical in real-world applications, where accurately differentiating between true and false positives and negatives is essential. Also, this chapter provides a detailed comparison between the proposed VEDLM, and studies previously discussed in Section 2.3.3, all focused on cyber threats detection, these models were evaluated using the dataset described in Section 6.2.1. and shows

that the proposed VEDLM integrates these models through a voting mechanism, achieving an exceptional competitive F1 score of 99%. This integration illustrates that combining individuals' Deep Learning techniques within a voting framework significantly enhances predictive capabilities, yielding promising results for cyber threat detection. The comparison aims to demonstrate that the proposed VEDLM, trained and tested on the same dataset, outperforms these state-of-the-art techniques and baseline models. By showcasing the enhanced accuracy of 98.60% and 99% F1 score, the study emphasises the proposed VEDLM's effectiveness in cyber threat detection. This comprehensive analysis underscores the potential of proposed ensemble methods in deep learning to exceed traditional approaches, establishing a new mechanism in the field. The following section of the chapter includes a brief description of the problems addressed, provides research findings and contributions, and discusses the limitations of the study.

CHAPTER 7 : CONCLUSION AND FUTURE WORK

This chapter concludes the research by summarising the identified problems, presenting the research findings and their contributions, and discussing the study's limitations. Potential work for future research is also explored.

7.1 Concluding remarks

Identifying cyber threats in tweets is significant for ensuring the security and integrity of social media platforms. This requires steady progression in strategies to combat advancing threats and false strategies, driving the investigation of Deep Learning (DL) strategies. These advanced approaches point to upgrade accuracy, flexibility, and in general execution in identifying threatening activities, thereby defending both social media platforms and their clients. In any case, challenges like imbalanced datasets, dynamic cyber threat strategies, and the need for strong showing of generalisations emphasise the requirement for continuous investigation and development in this basic space.

Although ML and DL techniques have shown promising for detecting cyber threats, they face hurdles. These include limitations within the algorithms themselves, like bias from imbalanced data and overfitting. Additionally, the ever-changing nature of cyber threats, with their lopsided data distribution and evolving patterns, throws another wrench into the accuracy and reliability of these detection systems.

To address the limitations of current cyber threat detection on social media, this research developed and evaluated three novel machine learning (ML) and deep learning (DL) models. These models aim to enhance the accuracy and efficiency of identifying cyber threats. To achieve this, a comprehensive review of existing literature and recent advancements in cyber threat detection methodologies was conducted, answering the research questions presented in Section 1.2.

The contributions of this aspect of the research are multifaceted:

It provides a comprehensive exploration of recent advancements in cyber threat detection methodologies, offering insights into the most effective ML, DL models for cyber threat detection on X. Through meticulous experimentation and analysis, this research contributes to the advancement of cyber threat detection technologies, paving the way for more robust and reliable cyber threats detection systems on X.

To address these challenges and bolster the efficacy of cyber threat detection systems on X, this research undertook the development and evaluation of novel cyber threat detection models, designed to enhance the accuracy and efficiency of cyber threat detection algorithms on X. The contributions and findings of this research endeavour are outlined below.

- **Contribution 1 (Chapter 2):** This study identified the specific threats prevalent on X, analysing the underlying motivations, and identifying the primary challenges in detecting these threats. It provides a comprehensive review of the related literature on feature extraction methods, discussing both Machine Learning (ML) and Deep Learning (DL) techniques and classifiers applied for cybersecurity threat detection.

- **Contribution 2 (Chapter 4):** This study investigates feature extraction for cybersecurity threat detection on X. It is developing a Subspace Random Ensemble machine learning technique using K-Nearest Neighbours (KNN), Logistic Regression (LR), Decision Tree (DT), and Random Forest (RF). The analysis covers their strengths, weaknesses, recognised threats, applicable databases, and performance metrics. Furthermore, the compares popular X threat detection databases used for model training and evaluation. This research aims to address this gap, specifically by answering the question *RQ 1. "What is the most effective machine learning technique for extracting and selecting features from X datasets to build a cybersecurity threat intelligence model?"*

- **Contribution 3 (Chapter 5):** A novel proposed voting Ensemble Deep Learning-based classifiers have been developed and evaluated including the LSTM-BiLSTM, IDCNN-BiLSTM, and CNN, and the obtained results have been compared with the baseline models and other the state-of-the-art models to answer research question 2 - *RQ 2. What is the most effective deep learning technique for extracting and selecting features from X datasets to build a cybersecurity threat intelligence model*?*"*

- **Contribution 4 (Chapters 4, 5):** Developed an ensemble learning model that combines predictions from pre-trained ensemble machine learning and deep learning architectures to improve predictive performance. This framework aims to

capture a broader spectrum of threat-related information compared to traditional methods. The study also presents a comprehensive evaluation methodology for these ML and DL models. This includes using various measurement techniques and comparing the performance of the proposed models against baseline models. The findings provide valuable insights into the effectiveness of the ensemble approach for cyber threat detection. Additionally, the research addresses the question *RQ 3. How can an ensemble learning model be developed by aggregating predictions from pre-trained ensemble machine learning and deep learning architectures to improve predictive performance?*

- **Contribution 4 (Chapter 6):** The comparative analysis in this study highlights the superiority of the proposed voting Ensemble Deep Learning Model (VEDLM) in cyber threat detection., VEDLM consistently outperformed Subspace Random Ensemble machine learning techniques and baseline models. Notably, it achieved an impressive accuracy of 98.60% and an F1 score of 99%, demonstrating its exceptional effectiveness in identifying and mitigating cyber threats. This comprehensive study underscores the potential of ensemble methods in deep learning to revolutionise cyber threat detection, surpassing traditional approaches and establishing a new benchmark in the field. The research addresses the question *RQ 4. How effective is the proposed VEDLM cybersecurity threat models using X datasets?*

- **Contribution 5:** Enriched Dataset for Performance Assessment: To ensure a comprehensive evaluation, the study curates a new dataset specifically designed to assess the performance of the proposed ensemble model in detecting cyber threats on X. This dataset contributes to the advancement of research in this domain by providing a valuable resource for future studies.

In conclusion, the proposed SREMLM and VEDLM models in this research hold promising potential for improving cyber threat detection capabilities on X. By utilising an automated cyber threat detection model, the X platform can enhance its ability to identify and mitigate threatening activities, thereby safeguarding social media, and ensuring a secure environment for both providers and consumers. Moreover,

integrating artificial intelligence into cyber threat detection has the potential to streamline diagnostic processes, lower operational costs, and enhance overall efficiency in monitoring and analysing tweets, although the SREMLM model has a slight advantage in accuracy, the VEDLM, which employs deep learning techniques, consistently excels in precision, recall, and F1 score. These metrics are crucial in real-world scenarios, where distinguishing between true and false positives and negatives is vital.

7.2 Current limitations

This section details the limitations identified during the development and evaluation of both the project and automated cyber threat detection systems designed for X.

1. This PhD thesis presents the development and evaluation of novel algorithms designed to identify cyber threats within the textual content of tweets. The research is specifically confined to text-based information and excludes analysis of accompanying photos, videos, or other data associated with tweets.
2. This research project investigated methods for improving Machine Learning, Deep Learning models. Although the outcomes were benchmarked against state-of-the-art techniques.

3. X's vast, rapidly changing, and often noisy data, coupled with privacy concerns and the lack of detailed information in tweets, hinders effective cyber threat detection.

4. Overwhelming amount of data: X produces vast amounts of data quickly, making it difficult to process and analyse effectively.

5. Many irrelevant tweets: Not all tweets are about cyber threats, leading to false alarms.

6. Complex language: Informal language, slang, and sarcasm on X can hinder accurate threat identification.

7. Constant change: Cyber threats evolve rapidly, making it challenging to stay ahead.

8. Lack of context: Tweets are short and often lack crucial details needed for a full understanding of a threat.

7.3 Future directions

In future work, I expect to test and compare the current presented research with photos and videos of the constancy algorithm.

1. Advanced Natural Language Processing (NLP) Techniques.

    - Contextual understanding: developing models that can better grasp the context of tweets, including sarcasm, irony, and implied meanings, to accurately identify threats.

    - Multilingual capabilities: expanding detection to languages beyond English, as cyber threats often manifest in multiple languages.

    - Dialect and slang handling: addressing challenges posed by regional dialects, slang, and evolving language patterns.

2. Hybrid approaches combining Machine and Deep Learning methods.

    - Feature engineering and deep learning: leveraging domain expertise to create informative features and combining them with deep learning models for improved performance.

3. Real-time threat intelligence and response.

    - Low-latency models: developing models capable of processing tweets and making threat assessments in real time.

- Integration with security systems: seamlessly integrating threat detection systems with existing security infrastructure for immediate response.

- Incident response automation: automating routine incident response tasks to accelerate response times.

REFERENCES

- Abadi, M, Barham, P, Chen, J, Chen, Z, Davis, A, Dean, J, Devin, M, Ghemawat, S, Irving, G & Isard, M 2016, '{TensorFlow}: a system for {Large-Scale} machine learning', 12th USENIX symposium on operating systems design and implementation (OSDI 16), pp. 265-83.
- Abdelhaq, H, Sengstock, C & Gertz, M 2013, 'Eventweet: Online localized event detection from twitter', Proceedings of the VLDB Endowment, vol. 6, no. 12, pp. 1326-9.
- Abu-El-Rub, N & Mueen, A 2019, 'Botcamp: Bot-driven interactions in social campaigns', The world wide web conference, pp. 2529-35.
- Agarwal, A, Iqbal, M, Mitra, B, Kumar, V & Lal, N 2021, 'Hybrid CNN-BILSTM-attention based identification and prevention system for banking transactions', NVEO-NATURAL VOLATILES & ESSENTIAL OILS Journal| NVEO, pp. 2552-60.
- Ahmad, R, Alsmadi, I, Alhamdani, W & Tawalbeh, La 2022, 'A deep learning ensemble approach to detecting unknown network attacks', Journal of Information Security and applications, vol. 67, p. 103196.
- Ahmed, F & Abulaish, M 2013, 'A generic statistical approach for spam detection in online social networks', Computer Communications, vol. 36, no. 10-11, pp. 1120-9.
- Aiyer, B, Caso, J, Russell, P & Sorel, M 2022, 'New survey reveals $2 trillion market opportunity for cybersecurity technology and service providers', Governance, vol. 1, pp. 2-0.
- Akoto, W 2024, 'Who spies on whom? Unravelling the puzzle of state-sponsored cyber economic espionage', Journal of Peace Research, vol. 61, no. 1, pp. 59-71.
- Alarfaj, FK, Malik, I, Khan, HU, Almusallam, N, Ramzan, M & Ahmed, M 2022, 'Credit card fraud detection using state-of-the-art machine learning and deep learning algorithms', IEEE Access, vol. 10, pp. 39700-15.
- Aleesa, A, Zaidan, B, Zaidan, A & Sahar, NM 2020, 'Review of intrusion detection systems based on deep learning techniques: coherent taxonomy, challenges, motivations, recommendations, substantial analysis and future directions', Neural Computing and Applications, vol. 32, pp. 9827-58.
- Alqahtani, AF & Ilyas, M 2024, 'A Machine Learning Ensemble Model for the Detection of Cyberbullying', arXiv preprint arXiv:2402.12538.
- Alshingiti, Z, Alaqel, R, Al-Muhtadi, J, Haq, QEU, Saleem, K & Faheem, MH 2023, 'A deep learning-based phishing detection system using CNN, LSTM, and LSTM-CNN', Electronics, vol. 12, no. 1, p. 232.
- Alsodi, O, Zhou, X, Gururajan, R & Shrestha, A 2021, 'A Survey on Detection of cybersecurity threats on Twitter using deep learning', 2021 8th International Conference on Behavioral and Social Computing (BESC), IEEE, pp. 1-5.
- Alves, F, Bettini, A, Ferreira, PM & Bessani, A 2021, 'Processing tweets for cybersecurity threat awareness', Information Systems, vol. 95, p. 101586.
- Amleshwaram, AA, Reddy, N, Yadav, S, Gu, G & Yang, C 2013, 'Cats: Characterizing automation of twitter spammers', 2013 Fifth International Conference on Communication Systems and Networks (COMSNETS), IEEE, pp. 1-10.

- Andriotis, P & Takasu, A 2018, 'Emotional bots: content-based spammer detection on social media', 2018 IEEE international workshop on information forensics and security (WIFS), IEEE, pp. 1-8.
- Arora, R, Gupta, R & Yadav, P 2024, 'Utilizing Ensemble Learning to enhance the detection of Malicious URLs in the Twitter dataset', 2024 4th International Conference on Innovative Practices in Technology and Management (ICIPTM), IEEE, pp. 1-6.
- Arora, T, Sharma, M & Khatri, SK 2019, 'Detection of cyber crime on social media using random forest algorithm', 2019 2nd International Conference on Power Energy, Environment and Intelligent Control (PEEIC), IEEE, pp. 47-51.
- Ayodele, TO 2010, 'Machine learning overview', New Advances in Machine Learning, pp. 9-19.
- Azam, Z, Islam, MM & Huda, MN 2023, 'Comparative analysis of intrusion detection systems and machine learning based model analysis through decision tree', IEEE Access.
- Bace, RG 2000, Intrusion detection, Sams Publishing.
- Bara, I-A, Fung, CJ & Dinh, T 2015, 'Enhancing Twitter spam accounts discovery using cross-account pattern mining', 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), IEEE, pp. 491-6.
- Behzadan, V, Aguirre, C, Bose, A & Hsu, W 2018, 'Corpus and deep learning classifier for collection of cyber threat indicators in twitter stream', 2018 IEEE International Conference on Big Data (Big Data), IEEE, pp. 5002-7.
- Bello-Orgaz, G, Jung, JJ & Camacho, D 2016, 'Social big data: Recent achievements and new challenges', Information Fusion, vol. 28, pp. 45-59.
- Benchaji, I, Douzi, S & El Ouahidi, B 2021, 'Credit card fraud detection model based on LSTM recurrent neural networks', Journal of Advances in Information Technology, vol. 12, no. 2.
- Berahman, K, Zhou, X, Li, Y, Gururajan, R, Barua, P, Acharya, R & Chennakesavan, SK 2024, 'New Ensemble Deep Learning Model for Gynaecological Cancer Risk Prediction'.
- Beskow, DM & Carley, KM 2019, 'Its all in a name: detecting and labeling bots by their name', Computational and mathematical organization theory, vol. 25, pp. 24-35.
- Boyd, DM & Ellison, NB 2007, 'Social network sites: Definition, history, and scholarship', Journal of computer-mediated communication, vol. 13, no. 1, pp. 210-30.
- Boyd, K, Eng, KH & Page, CD 2013, 'Area under the precision-recall curve: point estimates and confidence intervals', Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part III 13, Springer, pp. 451-66.
- Bruce, CS 1994, 'Research students' early experiences of the dissertation literature review', Studies in Higher Education, vol. 19, no. 2, pp. 217-29.
- Btoush, EALM, Zhou, X, Gururajan, R, Chan, KC, Genrich, R & Sankaran, P 2023, 'A systematic review of literature on credit card cyber fraud detection using machine and deep learning', PeerJ Computer Science, vol. 9, p. e1278.
- Cai, C, Li, L & Zeng, D 2017, 'Detecting social bots by jointly modeling deep behavior and content information', Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, pp. 1995-8.

- Campiolo, R, Santos, LAF, Batista, DM & Gerosa, MA 2013, 'Evaluating the utilization of Twitter messages as a source of security alerts', Proceedings of the 28th Annual ACM Symposium on Applied Computing, pp. 942-3.
- Caruccio, L, Desiato, D & Polese, G 2018, 'Fake account identification in social networks', 2018 IEEE international conference on big data (big data), IEEE, pp. 5078-85.
- Chambers, N, Fry, B & McMasters, J 2018, 'Detecting denial-of-service attacks from social media text: Applying nlp to computer security', Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pp. 1626-35.
- Chapelle, O, Schlkopf, B & Zien, A 2010, 'Semi-Supervised Learning'.
- Chavoshi, N, Hamooni, H & Mueen, A 2016, 'Debot: Twitter bot detection via warped correlation', Icdm, pp. 28-65.
- Checkland, P & Holwell, S 1998, 'Action research: its nature and validity', Systemic practice and action research, vol. 11, no. 1, pp. 9-21.
- Chen, Z, Tanash, RS, Stoll, R & Subramanian, D 2017, 'Hunting malicious bots on twitter: An unsupervised approach', Social Informatics: 9th International Conference, SocInfo 2017, Oxford, UK, September 13-15, 2017, Proceedings, Part II 9, Springer, pp. 501-10.
- Chew, PA 2018, 'Searching for unknown unknowns: Unsupervised bot detection to defeat an adaptive adversary', Social, Cultural, and Behavioral Modeling: 11th International Conference, SBP-BRiMS 2018, Washington, DC, USA, July 10-13, 2018, Proceedings 11, Springer, pp. 357-66.
- Chu, Z, Gianvecchio, S, Wang, H & Jajodia, S 2010, 'Who is tweeting on Twitter: human, bot, or cyborg?', Proceedings of the 26th annual computer security applications conference, pp. 21-30.
- Chu, Z, Gianvecchio, S, Wang, H & Jajodia, S 2012, 'Detecting automation of twitter accounts: Are you a human, bot, or cyborg?', IEEE transactions on dependable and secure computing, vol. 9, no. 6, pp. 811-24.
- Coyac-Torres, JE, Sidorov, G, Aguirre-Anaya, E & Hernández-Oregón, G 2023, 'Cyberattack detection in social network messages based on convolutional neural networks and NLP techniques', Machine Learning and Knowledge Extraction, vol. 5, no. 3, pp. 1132-48.
- Creado, Y & Ramteke, V 2020, 'Active cyber defence strategies and techniques for banks and financial institutions', Journal of Financial Crime, vol. 27, no. 3, pp. 771-80.
- Cresci, S, Di Pietro, R, Petrocchi, M, Spognardi, A & Tesconi, M 2016, 'DNA-inspired online behavioral modeling and its application to spambot detection', IEEE Intelligent Systems, vol. 31, no. 5, pp. 58-64.
- Cresci, S, Di Pietro, R, Petrocchi, M, Spognardi, A & Tesconi, M 2017, 'Social fingerprinting: detection of spambot groups through DNA-inspired behavioral modeling', IEEE transactions on dependable and secure computing, vol. 15, no. 4, pp. 561-76.
- Dabiri, S & Heaslip, K 2019, 'Developing a Twitter-based traffic event detection model using deep learning architectures', Expert systems with applications, vol. 118, pp. 425-39.
- Daouadi, KE, Rebaï, RZ & Amous, I 2019, 'Bot detection on online social networks using deep forest', Artificial Intelligence Methods in Intelligent

Algorithms: Proceedings of 8th Computer Science On-line Conference 2019, Vol. 2 8, Springer, pp. 307-15.
- David, I, Siordia, OS & Moctezuma, D 2016, 'Features combination for the detection of malicious Twitter accounts', 2016 IEEE international autumn meeting on power, electronics and computing (ROPEC), IEEE, pp. 1-6.
- Dawson Jr, ME 2021, 'Cyber warfare: threats and opportunities'.
- de Andrade, NNG, Martin, A & Monteleone, S 2013, '" All the better to see you with, my dear": Facial recognition and privacy in online social networks', IEEE security & privacy, vol. 11, no. 3, pp. 21-8.
- De Souza, GA & Da Costa-Abreu, M 2020, 'Automatic offensive language detection from Twitter data using machine learning and feature selection of metadata', 2020 International Joint Conference on Neural Networks (IJCNN), IEEE, pp. 1-6.
- Deshmukh, R, Shinde, S, Yadav, B, Pathak, A & Shetty, A 2022, 'Darkintellect: An Approach to Detect Cyber Threat Using Machine Learning Techniques on Open-Source Information', Mathematical Statistician and Engineering Applications, vol. 71, no. 4, pp. 1431-9.
- Dickerson, JP, Kagan, V & Subrahmanian, V 2014, 'Using sentiment to detect bots on twitter: Are humans more opinionated than bots?', 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014), IEEE, pp. 620-7.
- Ding, Z, Xia, R, Yu, J, Li, X & Yang, J 2018, 'Densely connected bidirectional lstm with applications to sentence classification', CCF International Conference on Natural Language Processing and Chinese Computing, Springer, pp. 278-87.
- Diomidous, M, Chardalias, K, Magita, A, Koutonias, P, Panagiotopoulou, P & Mantas, J 2016, 'Social and psychological effects of the internet use', Acta informatica medica, vol. 24, no. 1, p. 66.
- Dionísio, N, Alves, F, Ferreira, PM & Bessani, A 2019, 'Cyberthreat detection from twitter using deep neural networks', 2019 International Joint Conference on Neural Networks (IJCNN), IEEE, pp. 1-8.
- Dionísio, N, Alves, F, Ferreira, PM & Bessani, A 2020, 'Towards end-to-end cyberthreat detection from Twitter using multi-task learning', 2020 International Joint Conference on Neural Networks (IJCNN), IEEE, pp. 1-8.
- Dorri, A, Abadi, M & Dadfarnia, M 2018, 'Socialbothunter: Botnet detection in twitter-like social networking services using semi-supervised collective classification', 2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), IEEE, pp. 496-503.
- Dreßing, H, Bailer, J, Anders, A, Wagner, H & Gallas, C 2014, 'Cyberstalking in a large sample of social network users: Prevalence, characteristics, and impact upon victims', Cyberpsychology, Behavior, and Social Networking, vol. 17, no. 2, pp. 61-7.
- Dressler, JC, Bronk, C & Wallach, DS 2015, 'Exploiting military OpSec through open-source vulnerabilities', MILCOM 2015-2015 IEEE Military Communications Conference, IEEE, pp. 450-8.

- El Asam, A & Samara, M 2016, 'Cyberbullying and the law: A review of psychological and legal challenges', Computers in Human Behavior, vol. 65, pp. 127-41.
- Elman, JL 1990, 'Finding structure in time', Cognitive science, vol. 14, no. 2, pp. 179-211.
- Elreedy, D, Atiya, AF & Kamalov, F 2024, 'A theoretical distribution analysis of synthetic minority oversampling technique (SMOTE) for imbalanced learning', Machine Learning, vol. 113, no. 7, pp. 4903-23.
- Faghani, MR & Saidi, H 2009, 'Malware propagation in online social networks', 2009 4th International Conference on Malicious and Unwanted Software (MALWARE), IEEE, pp. 8-14.
- Faghani, MR & Nguyen, UT 2014, 'A study of clickjacking worm propagation in online social networks', Proceedings of the 2014 IEEE 15th International Conference on Information Reuse and Integration (IEEE IRI 2014), IEEE, pp. 68-73.
- Familoni, BT 2024, 'Cybersecurity challenges in the age of AI: theoretical approaches and practical solutions', Computer Science & IT Research Journal, vol. 5, no. 3, pp. 703-24.
- Fang, W, Li, X, Zhou, P, Yan, J, Jiang, D & Zhou, T 2021, 'Deep learning anti-fraud model for internet loan: Where we are going', IEEE Access, vol. 9, pp. 9777-84.
- Fang, Y, Gao, J, Liu, Z & Huang, C 2020, 'Detecting Cyber Threat Event from Twitter Using IDCNN and BiLSTM', Applied Sciences, vol. 10, no. 17, p. 5922.
- Ferrara, E, Varol, O, Davis, C, Menczer, F & Flammini, A 2016, 'The rise of social bots', Communications of the ACM, vol. 59, no. 7, pp. 96-104.
- Firdausi, I, Erwin, A & Nugroho, AS 2010, 'Analysis of machine learning techniques used in behavior-based malware detection', 2010 second international conference on advances in computing, control, and telecommunication technologies, IEEE, pp. 201-3.
- Gable, GG 1994, 'Integrating case study and survey research methods: an example in information systems', European journal of information systems, vol. 3, no. 2, pp. 112-26.
- Gadekar, C & Rakshit, PP 2020, 'Study to Perform Opinion Mining on Motivation Factors Generating Cyber Crime by Twitter Analytics', Proceedings of the International Conference on Innovative Computing & Communications (ICICC).
- Ghazinour, K, Matwin, S & Sokolova, M 2016, 'YOURPRIVACYPROTECTOR, A recommender system for privacy settings in social networks', arXiv preprint arXiv:1602.01937.
- Gilani, Z, Kochmar, E & Crowcroft, J 2017, 'Classification of twitter accounts into automated agents and human users', Proceedings of the 2017 IEEE/ACM international conference on advances in social networks analysis and mining 2017, pp. 489-96.
- González-Manzano, L, González-Tablas, AI, de Fuentes, JM & Ribagorda, A 2014, 'Cooped: Co-owned personal data management', Computers & security, vol. 47, pp. 41-65.
- Graham, CM & Lu, Y 2023, 'Skills expectations in cybersecurity: semantic network analysis of job advertisements', Journal of Computer Information Systems, vol. 63, no. 4, pp. 937-49.

- Graves, A & Schmidhuber, J 2005, 'Framewise phoneme classification with bidirectional LSTM and other neural network architectures', Neural networks, vol. 18, no. 5-6, pp. 602-10.
- Gupta, A, Budania, H, Singh, P & Singh, PK 2017, 'Facebook based choice filtering', 2017 IEEE 7th International Advance Computing Conference (IACC), IEEE, pp. 875-9.
- Gurajala, S, White, JS, Hudson, B & Matthews, JN 2015, 'Fake Twitter accounts: profile characteristics obtained using an activity-based pattern detection approach', Proceedings of the 2015 international conference on social media & society, pp. 1-7.
- Gurunath, R, Klaib, MFJ, Samanta, D & Khan, MZ 2021, 'Social media and steganography: use, risks and current status', IEEE Access, vol. 9, pp. 153656-65.
- Hatcher, WG & Yu, W 2018, 'A survey of deep learning: Platforms, applications and emerging research trends', IEEE Access, vol. 6, pp. 24411-32.
- Hochreiter, S & Schmidhuber, J 1997, 'Long short-term memory', Neural computation, vol. 9, no. 8, pp. 1735-80.
- Horrocks, I, Patel-Schneider, PF, Boley, H, Tabet, S, Grosof, B & Dean, M 2004, 'SWRL: A semantic web rule language combining OWL and RuleML', W3C Member submission, vol. 21, no. 79, pp. 1-31.
- Humayun, M, Niazi, M, Jhanjhi, N, Alshayeb, M & Mahmood, S 2020, 'Cyber Security Threats and Vulnerabilities: A Systematic Mapping Study', Arabian Journal for Science and Engineering, pp. 1-19.
- Hunter, J & Dale, D 2007, 'The matplotlib user's guide', Matplotlib 0.90. 0 user's guide.
- Igawa, RA, Barbon Jr, S, Paulo, KCS, Kido, GS, Guido, RC, Júnior, MLP & da Silva, IN 2016, 'Account classification in online social networks with LBCA and wavelets', Information sciences, vol. 332, pp. 72-83.
- Jagatic, TN, Johnson, NA, Jakobsson, M & Menczer, F 2007, 'Social phishing', Communications of the ACM, vol. 50, no. 10, pp. 94-100.
- Ji, Y, He, Y, Jiang, X, Cao, J & Li, Q 2016, 'Combating the evasion mechanisms of social bots', Computers & security, vol. 58, pp. 230-49.
- Joe, MM & Ramakrishnan, B 2017, 'Novel authentication procedures for preventing unauthorized access in social networks', Peer-to-Peer Networking and Applications, vol. 10, no. 4, pp. 833-43.
- Joseph, VR & Vakayil, A 2022, 'SPlit: An optimal method for data splitting', Technometrics, vol. 64, no. 2, pp. 166-76.
- Jr, SB, Campos, GF, Tavares, GM, Igawa, RA, Jr, MLP & Guido, RC 2018, 'Detection of human, legitimate bot, and malicious bot in online social networks based on wavelets', ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), vol. 14, no. 1s, pp. 1-17.
- Kantepe, M & Ganiz, MC 2017, 'Preprocessing framework for Twitter bot detection', 2017 International conference on computer science and engineering (ubmk), IEEE, pp. 630-4.
- Kaur, G, Bonde, U, Pise, KL, Yewale, S, Agrawal, P, Shobhane, P, Maheshwari, S, Pinjarkar, L & Gangarde, R 2024, 'Social Media in the Digital Age: A Comprehensive Review of Impacts, Challenges and Cybercrime', Engineering Proceedings, vol. 62, no. 1, p. 6.

- Kergl, D, Roedler, R & Rodosek, GD 2016, 'Detection of zero day exploits using real-time social media streams', Advances in Nature and Biologically Inspired Computing: Proceedings of the 7th World Congress on Nature and Biologically Inspired Computing (NaBIC2015) in Pietermaritzburg, South Africa, held December 01-03, 2015, Springer, pp. 405-16.
- Ketkar, N & Ketkar, N 2017, 'Introduction to keras', Deep learning with python: a hands-on introduction, pp. 97-111.
- Khaled, S, El-Tazi, N & Mokhtar, HM 2018, 'Detecting fake accounts on social media', 2018 IEEE international conference on big data (big data), IEEE, pp. 3672-81.
- Khanday, AMUD, Rabani, ST, Khan, QR & Malik, SH 2022, 'Detecting twitter hate speech in COVID-19 era using machine learning and ensemble learning techniques', International Journal of Information Management Data Insights, vol. 2, no. 2, p. 100120.
- Koloveas, P, Chantzios, T, Alevizopoulou, S, Skiadopoulos, S & Tryfonopoulos, C 2021, INTIME: A Machine Learning-Based Framework for Gathering and Leveraging Web Data to Cyber-Threat Intelligence. Electronics 2021, 10, 818, s Note: MDPI stays neutral with regard to jurisdictional claims in published ….
- Kotsiantis, SB, Zaharakis, I & Pintelas, P 2007, 'Supervised machine learning: A review of classification techniques', Emerging artificial intelligence applications in computer engineering, vol. 160, no. 1, pp. 3-24.
- Krishna, TVS, Krishna, TSR, Kalime, S, Krishna, CVM, Neelima, S & PBV, RR 2024, 'A novel ensemble approach for Twitter sentiment classification with ML and LSTM algorithms for real-time tweets analysis', Indonesian Journal of Electrical Engineering and Computer Science, vol. 34, no. 3, pp. 1904-14.
- Krombholz, K, Hobel, H, Huber, M & Weippl, E 2015, 'Advanced social engineering attacks', Journal of Information Security and applications, vol. 22, pp. 113-22.
- Kudugunta, S & Ferrara, E 2018, 'Deep neural networks for bot detection', Information sciences, vol. 467, pp. 312-22.
- Lanza, C & Lodi, L 2024, 'Towards a semi-automatic classifier of malware through tweets for early warning threat detection', JLIS. it, vol. 15, no. 2, pp. 101-18.
- Le, B-D, Wang, G, Nasim, M & Babar, MA 2019, 'Gathering cyber threat intelligence from Twitter using novelty classification', 2019 International Conference on Cyberworlds (CW), IEEE, pp. 316-23.
- Le Sceller, Q, Karbab, EB, Debbabi, M & Iqbal, F 2017, 'Sonar: Automatic detection of cyber security events over the twitter stream', Proceedings of the 12th International Conference on Availability, Reliability and Security, pp. 1-11.
- LeCun, Y, Bengio, Y & Hinton, G 2015, 'Deep learning', nature, vol. 521, no. 7553, pp. 436-44.
- LeCun, Y, Bottou, L, Bengio, Y & Haffner, P 1998, 'Gradient-based learning applied to document recognition', Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-324.
- Lee, K, Eoff, B & Caverlee, J 2011, 'Seven months with the devils: A long-term study of content polluters on twitter', Proceedings of the international AAAI conference on web and social media, pp. 185-92.

- Lee, S & Kim, J 2013, 'Warningbird: A near real-time detection system for suspicious urls in twitter stream', IEEE transactions on dependable and secure computing, vol. 10, no. 3, pp. 183-95.
- Leedy, PD & Ormrod, JE 2005, Practical research, Pearson Custom.
- Li, F, Wu, K, Lei, J, Wen, M, Bi, Z & Gu, C 2015, 'Steganalysis over large-scale social networks with high-order joint features and clustering ensembles', IEEE Transactions on Information Forensics and Security, vol. 11, no. 2, pp. 344-57.
- Loyola-González, O, Monroy, R, Rodríguez, J, López-Cuevas, A & Mata-Sánchez, JI 2019, 'Contrast pattern-based classification for bot detection on twitter', IEEE Access, vol. 7, pp. 45800-17.
- Lughbi, H, Mars, M & Almotairi, K 2024, 'A Novel NLP-Driven Dashboard for Interactive CyberAttacks Tweet Classification and Visualization', Information, vol. 15, no. 3, p. 137.
- Lyytinen, KJ & Klein, HK 1985, '12 THE CRITICAL THEORY OF JURGEN HABERMAS AS A BASIS FOR A THEORY OF INFORMATION SYSTEMS'.
- Mahaini, MI & Li, S 2021, 'Detecting cyber security related Twitter accounts and different sub-groups: A multi-classifier approach', Proceedings of the 2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, pp. 599-606.
- Main, W & Shekokhar, N 2015, 'Twitterati identification system', Procedia Computer Science, vol. 45, pp. 32-41.
- Manakitsa, N, Maraslidis, GS, Moysis, L & Fragulis, GF 2024, 'A review of machine learning and deep learning for object detection, semantic segmentation, and human action recognition in machine and robotic vision', Technologies, vol. 12, no. 2, p. 15.
- Miller, Z, Dickinson, B, Deitrick, W, Hu, W & Wang, AH 2014, 'Twitter spammer detection using data stream clustering', Information sciences, vol. 260, pp. 64-73.
- Minnich, A, Chavoshi, N, Koutra, D & Mueen, A 2017, 'BotWalk: Efficient adaptive exploration of Twitter bot networks', Proceedings of the 2017 IEEE/ACM international conference on advances in social networks analysis and mining 2017, pp. 467-74.
- Mittal, S, Das, PK, Mulwad, V, Joshi, A & Finin, T 2016, 'Cybertwitter: Using twitter to generate alerts for cybersecurity threats and vulnerabilities', 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), IEEE, pp. 860-7.
- Morstatter, F, Wu, L, Nazer, TH, Carley, KM & Liu, H 2016, 'A new approach to bot detection: striking the balance between precision and recall', 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), IEEE, pp. 533-40.
- Mousavi, R & Eftekhari, M 2015, 'A new ensemble learning methodology based on hybridization of classifier ensemble selection approaches', Applied Soft Computing, vol. 37, pp. 652-66.
- Muneer, A, Alwadain, A, Ragab, MG & Alqushaibi, A 2023, 'Cyberbullying detection on social media using stacking ensemble learning and enhanced BERT', Information, vol. 14, no. 8, p. 467.
- Munk, T 2022, The rise of politically motivated cyber attacks: Actors, attacks and cybersecurity, Routledge.

- Munschauer, M, Nguyen, CT, Sirokman, K, Hartigan, CR, Hogstrom, L, Engreitz, JM, Ulirsch, JC, Fulco, CP, Subramanian, V & Chen, J 2018, 'The NORAD lncRNA assembles a topoisomerase complex critical for genome stability', nature, vol. 561, no. 7721, pp. 132-6.
- Nauman, M, Azam, N & Yao, J 2016, 'A three-way decision making approach to malware analysis using probabilistic rough sets', Information sciences, vol. 374, pp. 193-209.
- Nazir, F, Ghazanfar, MA, Maqsood, M, Aadil, F, Rho, S & Mehmood, I 2019, 'Social media signal detection using tweets volume, hashtag, and sentiment analysis', Multimedia Tools and Applications, vol. 78, no. 3, pp. 3553-86.
- Nguyen, TT, Tahir, H, Abdelrazek, M & Babar, A 2020, 'Deep learning methods for credit card fraud detection', arXiv preprint arXiv:2012.03754.
- Noh, G, Oh, H, Kang, Y-m & Kim, C-k 2014, 'PSD: Practical Sybil detection schemes using stickiness and persistence in online recommender systems', Information sciences, vol. 281, pp. 66-84.
- Noor, U, Anwar, Z, Amjad, T & Choo, K-KR 2019, 'A machine learning-based FinTech cyber threat attribution framework using high-level indicators of compromise', Future Generation Computer Systems, vol. 96, pp. 227-42.
- Olaitan, OL, David, AO & Michael, OA 2024, 'Deep Learning Approach for Classification of Tweets in Detecting Cyber Truculent', Advances in Research, vol. 25, no. 2, pp. 113-22.
- Omar, S, Ngadi, A & Jebur, HH 2013, 'Machine learning techniques for anomaly detection: an overview', International Journal of Computer Applications, vol. 79, no. 2.
- Oosthoek, K & Doerr, C 2020, 'Cyber Threat Intelligence: A Product Without a Process?', International Journal of Intelligence and CounterIntelligence, pp. 1-16.
- Ozbayoglu, AM, Gudelek, MU & Sezer, OB 2020, 'Deep learning for financial applications: A survey', Applied Soft Computing, vol. 93, p. 106384.
- Pedregosa, F, Varoquaux, G, Gramfort, A, Michel, V, Thirion, B, Grisel, O, Blondel, M, Prettenhofer, P, Weiss, R & Dubourg, V 2011, 'Scikit-learn: Machine learning in Python', the Journal of machine Learning research, vol. 12, pp. 2825-30.
- Ping, H & Qin, S 2018, 'A social bots detection model based on deep learning algorithm', 2018 IEEE 18th international conference on communication technology (icct), IEEE, pp. 1435-9.
- Queiroz, A, Keegan, B & Mtenzi, F 2017, 'Predicting software vulnerability using security discussion in social media', European Conference on Cyber Warfare and Security, Academic Conferences International Limited, pp. 628-34.
- Ramzan, N, Park, H & Izquierdo, E 2012, 'Video streaming over P2P networks: Challenges and opportunities', Signal Processing: Image Communication, vol. 27, no. 5, pp. 401-11.
- Rao, P, Kamhoua, C, Njilla, L & Kwiat, K 2018, 'Methods to detect cyberthreats on twitter', in Surveillance in Action, Springer, pp. 333-50.
- Rathore, S, Sharma, PK, Loia, V, Jeong, Y-S & Park, JH 2017, 'Social network security: Issues, challenges, threats, and solutions', Information sciences, vol. 421, pp. 43-69.

- Research, S & Department 2023, 'Global cybersecurity spending 2017-2022', Statista, viewed 10-11-2023, <.
- Ritter, A, Wright, E, Casey, W & Mitchell, T 2015, 'Weakly supervised extraction of computer security events from twitter', Proceedings of the 24th International Conference on World Wide Web, pp. 896-905.
- Rodriguez, A & Okamura, K 2019, 'Generating real time cyber situational awareness information through social media data mining', 2019 IEEE 43rd annual computer software and applications conference (COMPSAC), IEEE, pp. 502-7.
- Rodriguez, A & Okamura, K 2020, 'Enhancing data quality in real-time threat intelligence systems using machine learning', Social Network Analysis and Mining, vol. 10, no. 1, pp. 1-22.
- Romagna, M & Leukfeldt, RE 2024, 'Social Opportunity Structures in Hacktivism: Exploring Online and Offline Social Ties and the Role of Offender Convergence Settings in Hacktivist Networks', Victims & Offenders, pp. 1-23.
- Ruohonen, J, Hyrynsalmi, S & Leppänen, V 2020, 'A mixed methods probe into the direct disclosure of software vulnerabilities', Computers in Human Behavior, vol. 103, pp. 161-73.
- Sabottke, C, Suciu, O & Dumitraș, T 2015, 'Vulnerability disclosure in the age of social media: Exploiting twitter for predicting real-world exploits', 24th {USENIX} Security Symposium ({USENIX} Security 15), pp. 1041-56.
- Sani, AM & Moeini, A 2020, 'Real-time Event Detection in Twitter: A Case Study', 2020 6th International Conference on Web Research (ICWR), IEEE, pp. 48-51.
- Sapienza, A, Bessi, A, Damodaran, S, Shakarian, P, Lerman, K & Ferrara, E 2017, 'Early warnings of cyber threats in online discussions', 2017 IEEE International Conference on Data Mining Workshops (ICDMW), IEEE, pp. 667-74.
- Schmidhuber, J 2015, 'Deep learning in neural networks: An overview', Neural networks, vol. 61, pp. 85-117.
- Schuster, M & Paliwal, KK 1997, 'Bidirectional recurrent neural networks', IEEE transactions on Signal Processing, vol. 45, no. 11, pp. 2673-81.
- Shah, A, Varshney, S & Mehrotra, M 2024, 'Threats on online social network platforms: classification, detection, and prevention techniques', Multimedia Tools and Applications, pp. 1-33.
- Shahnawaz Ahmad, M & Mehraj Shah, S 2022, 'Unsupervised ensemble based deep learning approach for attack detection in IoT network', Concurrency and Computation: Practice and Experience, vol. 34, no. 27, p. e7338.
- Shi, P, Zhang, Z & Choo, K-KR 2019, 'Detecting malicious social bots based on clickstream sequences', IEEE Access, vol. 7, pp. 28855-62.
- Shin, H-S, Kwon, H-Y & Ryu, S-J 2020, 'A new text classification model based on contrastive word embedding for detecting cybersecurity intelligence in twitter', Electronics, vol. 9, no. 9, p. 1527.
- Shukla, H, Jagtap, N & Patil, B 2021, 'Enhanced Twitter bot detection using ensemble machine learning', 2021 6th International Conference on Inventive Computation Technologies (ICICT), IEEE, pp. 930-6.
- Siddiqui, T, Hina, S, Asif, R, Ahmed, S & Ahmed, M 2023, 'An ensemble approach for the identification and classification of crime tweets in the English

language', Computer Science and Information Technologies, vol. 4, no. 2, pp. 149-59.
- Sigholm, J & Bang, M 2013, 'Towards offensive cyber counterintelligence: Adopting a target-centric view on advanced persistent threats', 2013 European Intelligence and Security Informatics Conference, IEEE, pp. 166-71.
- Simran, K, Balakrishna, P, Vinayakumar, R & Soman, K 2019, 'Deep Learning Approach for Enhanced Cyber Threat Indicators in Twitter Stream', International Symposium on Security in Computing and Communication, Springer, pp. 135-45.
- Singh, S, Jeong, Y-S & Park, JH 2016, 'A survey on cloud computing security: Issues, threats, and solutions', Journal of Network and Computer Applications, vol. 75, pp. 200-22.
- Squicciarini, AC, Shehab, M & Wede, J 2010, 'Privacy policies for shared content in social network sites', The VLDB Journal, vol. 19, no. 6, pp. 777-96.
- Stokes, K & Carlsson, N 2013, 'A peer-to-peer agent community for digital oblivion in online social networks', 2013 Eleventh Annual Conference on Privacy, Security and Trust, IEEE, pp. 103-10.
- Subrahmanian, VS, Azaria, A, Durst, S, Kagan, V, Galstyan, A, Lerman, K, Zhu, L, Ferrara, E, Flammini, A & Menczer, F 2016, 'The DARPA Twitter bot challenge', Computer, vol. 49, no. 6, pp. 38-46.
- Susman, GI 1983, 'Action research: a sociotechnical systems perspective', Beyond method: Strategies for social research, vol. 95, p. 113.
- Teljstedt, C, Rosell, M & Johansson, F 2015, 'A semi-automatic approach for labeling large amounts of automated and non-automated social media user accounts', 2015 second european network intelligence conference, IEEE, pp. 155-9.
- Tounsi, W & Rais, H 2018, 'A survey on technical threat intelligence in the age of sophisticated cyber attacks', Computers & security, vol. 72, pp. 212-33.
- Trabelsi, S, Plate, H, Abida, A, Aoun, MMB, Zouaoui, A, Missaoui, C, Gharbi, S & Ayari, A 2015, 'Mining social networks for software vulnerabilities monitoring', 2015 7th International Conference on New Technologies, Mobility and Security (NTMS), IEEE, pp. 1-7.
- Vaiyapuri, T, Shankar, K, Rajendran, S, Kumar, S, Gaur, V, Gupta, D & Alharbi, M 2024, 'Automated cyberattack detection using optimal ensemble deep learning model', Transactions on Emerging Telecommunications Technologies, vol. 35, no. 4, p. e4899.
- Valliyammai, C & Devakunchari, R 2019, 'Distributed and scalable Sybil identification based on nearest neighbour approximation using big data analysis techniques', Cluster Computing, vol. 22, no. Suppl 6, pp. 14461-76.
- Van Laere, O, Schockaert, S & Dhoedt, B 2013, 'Georeferencing Flickr resources based on textual meta-data', Information sciences, vol. 238, pp. 52-74.
- Varol, O, Ferrara, E, Davis, C, Menczer, F & Flammini, A 2017, 'Online human-bot interactions: Detection, estimation, and characterization', Proceedings of the international AAAI conference on web and social media, pp. 280-9.
- Velayutham, T & Tiwari, PK 2017, 'Bot identification: Helping analysts for right data in twitter', 2017 3rd international conference on advances in computing, communication & automation (ICACCA)(fall), IEEE, pp. 1-5.

- Viejo, A, Castella-Roca, J & Rufián, G 2013, 'Preserving the user's privacy in social networking sites', International Conference on Trust, Privacy and Security in Digital Business, Springer, pp. 62-73.
- Von Solms, B & Von Solms, R 2018, 'Cybersecurity and information security–what goes where?', Information & Computer Security, vol. 26, no. 1, pp. 2-9.
- Wagner, TD, Mahbub, K, Palomar, E & Abdallah, AE 2019, 'Cyber threat intelligence sharing: Survey and research directions', Computers & security, vol. 87, p. 101589.
- Wang, J-H, Liu, T-W, Luo, X & Wang, L 2018, 'An LSTM approach to short text sentiment classification with word embeddings', Proceedings of the 30th conference on computational linguistics and speech processing (ROCLING 2018), pp. 214-23.
- Wang, W, Mauleon, R, Hu, Z, Chebotarov, D, Tai, S, Wu, Z, Li, M, Zheng, T, Fuentes, RR & Zhang, F 2018, 'Genomic variation in 3,010 diverse accessions of Asian cultivated rice', nature, vol. 557, no. 7703, pp. 43-9.
- Weir, GR, Toolan, F & Smeed, D 2011, 'The threats of social networking: Old wine in new bottles?', Information security technical report, vol. 16, no. 2, pp. 38-43.
- Xin, Y, Kong, L, Liu, Z, Chen, Y, Li, Y, Zhu, H, Gao, M, Hou, H & Wang, C 2018, 'Machine learning and deep learning methods for cybersecurity', IEEE Access, vol. 6, pp. 35365-81.
- Xu, H, Dong, M, Zhu, D, Kotov, A, Carcone, AI & Naar-King, S 2016, 'Text classification with topic-based word embedding and convolutional neural networks', Proceedings of the 7th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics, pp. 88-97.
- Yang, C, Harkreader, R & Gu, G 2013, 'Empirical evaluation and new design for fighting evolving twitter spammers', IEEE Transactions on Information Forensics and Security, vol. 8, no. 8, pp. 1280-93.
- Yang, W, Dong, G, Wang, W, Shen, G, Gong, L, Yu, M, Lv, J & Hu, Y 2014, 'Detecting bots in follower markets', Bio-Inspired Computing-Theories and Applications: 9th International Conference, BIC-TA 2014, Wuhan, China, October 16-19, 2014. Proceedings, Springer, pp. 525-30.
- Yılmaz, Y & Hero, AO 2018, 'Multimodal event detection in Twitter hashtag networks', Journal of Signal Processing Systems, vol. 90, no. 2, pp. 185-200.
- Zhang, C, Zhang, G & Sun, S 2009, 'A mixed unsupervised clustering-based intrusion detection model', 2009 Third International Conference on Genetic and Evolutionary Computing, IEEE, pp. 426-8.
- Zhang, Z & Gupta, BB 2018, 'Social media security and trustworthiness: overview and new direction', Future Generation Computer Systems, vol. 86, pp. 914-25.
- Zhao, X, Wang, L, Zhang, Y, Han, X, Deveci, M & Parmar, M 2024, 'A review of convolutional neural networks in computer vision', Artificial Intelligence Review, vol. 57, no. 4, p. 99.
- Zhu, X & Goldberg, AB 2009, 'Introduction to semi-supervised learning', Synthesis lectures on artificial intelligence and machine learning, vol. 3, no. 1, pp. 1-130.
- Zigomitros, A, Papageorgiou, A & Patsakis, C 2012, 'Social network content management through watermarking', 2012 IEEE 11th International Conference

on Trust, Security and Privacy in Computing and Communications, IEEE, pp. 1381-6.
- Zong, S, Ritter, A, Mueller, G & Wright, E 2019, 'Analyzing the perceived severity of cybersecurity threats reported on social media', arXiv preprint arXiv:1902.10680.

APPENDICES

Publications

No	Paper	Paper type
1.	Alsodi, O., Zhou, X., Gururajan, R. and Shrestha, A., 2021, October. A Survey on Detection of cybersecurity threats on Twitter using deep learning.	**PUBLISHED** In 2021 8th International Conference on Behavioral and Social Computing (BESC) (pp. 1-5). IEEE.
2.	Cyber Threat Detection on Twitter Using Deep Learning Techniques: IDCNN and BiLSTM Integration	**ACCEPTED WILL BE PUBLISHED** CBD2024 Conference IEEE Journal
3.	Optimizing Security: A Hybrid CNN-BiLSTM Model for Credit Card Cyber Fraud Detection	**ACCEPTED WILL BE PUBLISHED** CBD2024 Conference IEEE Journal
4.	From Tweets to Threats: A survey of Cybersecurity Threat Detection Challenges, AI-based Solutions and Potential Opportunities	**SUBMITTED UNDER REVIEW** Journal/ PeerJ Computer Science
5.	AI- ensemble approach for Detecting Cybersecurity Threats on Twitter	**SUBMITTED UNDER REVIEW** Journal/ PeerJ Applied Science.
6.	Credit Card Cyber Fraud Detection Using Machine Learning: A Comparative Analysis of Resampling Techniques.	**SUBMITTED UNDER REVIEW** Journal/ PeerJ Computer Science.