ENHANCING CYBERSECURITY: MACHINE LEARNING AND NATURAL LANGUAGE PROCESSING FOR ARABIC PHISHING EMAIL DETECTION

Thesis Submitted in Partial Fulfilment of the Requirements of the Degree of Doctor of Philosophy

SAID A. SALLOUM

School of Science, Engineering and Environment
University of Salford, UK

2023

Dedication

I would like to dedicate this work to my mom, who left me unexpectedly in November 2018; it excruciates my soul that I don't have her by my side to share this moment. I would also like to dedicate this to my wonderful better half and amazing sons – Ayham, Aysar and Azad.

Acknowledgments

First of all, I want to express my gratitude towards my supervisors, Prof. Sunil Vadera and Dr. Tarek Gaber, who have unconditionally supported me during the course of my PhD. They have tolerated, encouraged and educated me, and it was their guidance that helped me with my research as well as with writing my thesis. I am immensely grateful to Prof. Khaled Shaalan, my local supervisor, for his guidance, support, and advice throughout my PhD journey. His expertise in the field has been pivotal in shaping both my research and academic development. Prof. Shaalan's deep knowledge and insightful perspectives have consistently challenged and inspired me, driving me to new heights in my academic pursuits. The passion and dedication he brings to his work have been a constant source of inspiration for me. I would also like to thank all the specialists who were ready to take part in the research and willing to give me honest and valuable feedback. I want to thank my supportive family as well. I am lucky to have such supportive and loving parents who cheered me on during all my pursuits in life, and I owe all my success to my mother. I acknowledge all the sacrifices that she made for me. I also want to thank my siblings for always praying for me and supporting me. Moreover, I am grateful to my friends and co-workers for always being there for me. Last, but not the least, I want to thank my beloved wife, Raghad Alfaisal; I deeply appreciate her unwavering love, support and acceptance throughout my degree. It was her persistence that made me complete my thesis. She never let me give up, encouraging me whenever I became disheartened. I also want to give well-earned credit to my three little boys for the obedience and patience they showed while I was writing my thesis, which made my PhD a little less hard. I would like to express my sincere gratitude to IWSPA-AP 2018 Datasets NDA for providing the experimental data used in this research. I extend my appreciation to Prof. Rakesh M. Verma, the rightful owner of this dataset, for granting permission to utilize it in its original form or translated into Arabic.

Thank you, everyone.

List of Publications

The thesis comprises of the sections of the papers that have been published as a part of my research.

- Salloum, S., Gaber, T., Vadera, S., & Shaalan, K. (2021). Phishing Email Detection Using Natural Language Processing Techniques: A Literature Survey. *Procedia Computer Science, 189*, 19-28. https://doi.org/10.1016/j.procs.2021.05.077

- Salloum S., Gaber T., Vadera S., Shaalan K. (2021) Phishing Website Detection from URLs Using Classical Machine Learning ANN Model. In: Garcia-Alfaro J., Li S., Poovendran R., Debar H., Yung M. (eds) Security and Privacy in Communication Networks. SecureComm 2021. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 399. Springer, Cham. https://doi.org/10.1007/978-3-030-90022-9_28

- S. Salloum, T. Gaber, S. Vadera and K. Shaalan, "A Systematic Literature Review on Phishing Email Detection Using Natural Language Processing Techniques," in IEEE Access, vol. 10, pp. 65703-65727, 2022, https://doi: 10.1109/ACCESS.2022.3183083

- Salloum, S., Gaber, T., Vadera, S., & Shaalan, K. (2023). A New English/Arabic Parallel Corpus for Phishing Emails. ACM Transactions on Asian and Low-Resource Language Information Processing. https://doi.org/10.1145/3606031

List of Abbreviations

The table below illustrates the meanings of different abbreviations utilized in the thesis.

AB	AdaBoost
APWG	Anti-Phishing Working Group
BN	Bayes Networks
BoW	Bag-of-Words
BSFS	Binary Search Feature Selection
CharEmbedding	Character-Level Convolutional Neural Network
Chi-S	Chi-Square
CNN	Convolutional Neural Network
CS	Cuckoo Search
DMC	Dynamic Markov Chain
DT	Decision Tree
DTM	Document-Term Matrix
EAPD	English–Arabic Phishing Detection model
GRU	Gated Recurrent Unit
IBPD	Intent-Based Phishing Detection
IWSPA-AP	First Security and Privacy Analytics Anti-Phishing Shared Task
KNN	K-Nearest Neighbours
LDA	Latent Dirichlet Allocation
LIR	Linear Regression

LR	Logistic Regression
LSA	Latent Semantic Analysis
LSTM	Long short-term memory
LTM	Latent Topic Model
MI	Mutual information
ML	Machine Learning
MLP	Multilayer Perceptron
NB	Naïve Bayes
NER	Named Entity Recognition
NLP	Natural Language Processing
NN	Neural Networks
OVV	Out Of Vocabulary
PCA	Principal Component Analysis
POS	Part-of-speech tagging
PRISMA	Preferred Reporting Items for Systematic Reviews and Meta-Analysis
RF	Random Forest
RNN	Recurrent neural network
ROC	Receiver Operating Characteristic
SLR	Systematic Literature Review
SVM	Support Vector Machine
TF-IDF	Term Frequency-Inverse Document Frequency
WEKA	Waikato Environment for Knowledge Analysis
XGBoost	Extreme Gradient Boosting

Abstract

Phishing is a significant threat to the modern world, causing considerable financial losses. Although electronic mail has shown to be a valuable asset around the world in terms of facilitating communication for all parties involved, whether huge corporations or individuals communicating in their everyday lives, it has also brought with it its own set of issues. Scammers take advantage of such issues by sending out bogus emails to susceptible persons in order to acquire access to their personal information. Phishing email detection is considered an important research field, and the research community has tried hard to address this problem in various common languages like English. There are some other important languages, such as Arabic, which have not been given much attention when it comes to phishing detection. Arabic is the native language of more than 300 million people and is ranked as the fifth most extensively used language throughout the world. In terms of content-based phishing email detection, there has been relatively little research on Arabic language phishing emails. This study presents an English-Arabic Phishing Detection (EAPD) model developed on the word level (Term Frequency-Inverse Document Frequency (TF-IDF), Document-Term Matrix (DTM), and FastText embedding) and the character-level convolutional neural network (CharEmbedding) to decrease this gap. It will be one of the first studies to explore the extent to which machine learning (ML) and natural language processing (NLP) methods can be used to develop models for detecting English/Arabic phishing attacks. An English-Arabic parallel phishing email corpus was developed using the English and Arabic text provided by the leading security and privacy analytics anti-phishing shared task (IWSPA-AP 2018). To evaluate the effectiveness of the EAPD model, a collection of balanced 1258 emails in Arabic and English, featuring equal ratios of legitimate and phishing emails, was used. The experiments indicate that when using the Multilayer Perceptron (MLP) classifier combined with TF-IDF, the EAPD achieved an accuracy of 95.3% on Arabic datasets. The English text, on the other hand, reached a 95.7% accuracy when paired with the Support Vector Machine (SVM) classifier and TF-IDF. Salloum's list, a new set of Arabic stop words, was introduced and found that while traditional ML classifiers remained largely unaffected, deep learning (DL) models with FastText embedding, especially LSTM, showed a significant 14% variance following the integration of this extended list. Overall, this study presents a promising approach for detecting phishing emails in both English and Arabic, with high accuracy and efficiency.

Table of Content

List of Tables

List of Figures

Chapter One

Introduction

1.1 Overview

The immense growth of internet technologies has dramatically altered online user interaction while creating increasingly severe security problems. Newly emerging threats of the present world target computer users and could potentially steal their identity and money. Along with the use of technology, phishing threats use social engineering to pilfer data related to a victim's identity and accounts. Hence, it is imperative to curtail the threat and criminal activity associated with phishing. According to the Anti-Phishing Working Group (APWG), "the total number of phishing attacks detected by the APWG in Q3 of 2022 was 1,270,883, that was up notably from the 1,097,811 seen in Q2 of 2022, and from the 1,025,968 seen in Q1 of 2022" [1]. Phishing attacks with the subject of the coronavirus disease of 2019 (COVID-19) have been deployed since mid-September of the following year. Phishing attacks mostly have a textual composition of subjects, such as internet and security technologies, as well as information regarding the COVID-19 pandemic to attract their targets [2]. The incidence of phishing has seen a substantial increase, accompanied by a corresponding rise in the associated damages, as indicated by available data. Phishing, characterised by the extraction of sensitive information through deceptive means, poses a significant social engineering threat. Among the various communication channels exploited for such attacks, emails and instant messaging stand out prominently. Attackers adeptly portray themselves as legitimate and trustworthy individuals. This study specifically focuses on email, recognizing it as the primary communication tool frequently manipulated by attackers [3], [4], thus rendering it a crucial channel for analysis [5]. Despite its prevalence, the automatic detection of phishing emails and messages remains a challenging task [6]. Described as a sophisticated form of digital deception, phishing emails are meticulously crafted to mimic credible messages, with the primary objective of enticing recipients into divulging confidential information, thereby compromising their cybersecurity. These deceptive emails are skilfully designed to replicate the appearance and tone of authentic communications from trusted sources like banks or reputable corporations. Through a combination of persuasive language, scenarios engineered to induce a

sense of urgency, and alarming claims, these emails are strategically crafted to manipulate recipients into interacting with perilous links, downloading malicious software, or willingly providing sensitive data such as login credentials or financial information. The overarching goal of these deceitful emails is to facilitate data theft or introduce malware into computer systems, posing an ever-present and dynamic threat in the realm of digital communication. This study proposes a novel approach to address the challenge of phishing email detection termed Intent-Based Phishing Detection (IBPD). This innovative cybersecurity strategy aims to identify phishing emails by delving into the core intent of the scammer. It relies on advanced linguistic analysis, machine learning algorithms attuned to context, and insights derived from behavioural patterns. Instead of merely scanning for overt red flags like suspicious links or specific keywords, IBPD adopts a more nuanced approach, targeting the psychological and manipulative strategies inherent in phishing attempts. This refined methodology ensures improved effectiveness against complex and evolving threats, such as spear phishing and socially engineered schemes, by dynamically adapting to the evolving tactics employed by scammers.

To filter phish, numerous studies have been employed where these studies used either a rule-based method that will signal an email as phish or legitimate [7]–[9] or use text mining [10] or ML tactics considering phish filtering to classify a message as phish or legitimate. This later approach has employed ML algorithms like AdaBoost (AB) [11]–[14], Bayes Networks (BN) [12], [15]–[17], Convolutional Neural Network (CNN) [18]–[21], Decision Tree (DT) [11]–[13], [15], [17], [22]–[25]**,** SVM [13], [16], [19], [22], [23], [26], [27], k-nearest neighbors (KNN) [13], [15], [23], [25], [26], [28], Logistic Regression (LR) [14], [19], [22], [26], [29]–[33], Linear Regression (LIR) [26], [32], MLP [14], [34]–[37], Naïve Bayes (NB) [11], [18], [40], [41], [24], [26], [28], [31], [33], [37]–[39], Neural Networks (NN) [31], [40], [42]–[44], Recurrent neural network (RNN) [18], [19], [29], [30], [45]–[49], and Random Forest (RF) [11], [13], [15], [24], [26], [33], [44], [50], [51]**.**

The task of phish filtering is closely related to email language because phishing emails are typically written in a specific style and use specific language to trick their recipients into taking harmful actions. Phishing emails often use techniques such as social engineering, where the attacker tries to manipulate the recipient into revealing sensitive information or performing a certain action. This

can involve using urgent or threatening language, impersonating a trusted authority or institution, or offering a fake reward or incentive in exchange for action. Therefore, to effectively filter phishing emails, ML models need to be able to recognise these patterns and features within the language used in the email. This can involve analysing the email's content, as well as its structure and formatting, to identify suspicious or anomalous features. For example, a phishing email may use a generic greeting rather than addressing the recipient by name, or it may contain spelling or grammatical errors. It may also use hyperlinks that lead to suspicious or fake websites, or it may ask the recipient to provide sensitive information such as passwords or credit card numbers. By analysing these features and patterns within the language used in the email, ML models can be trained to effectively filter out phishing emails and prevent them from reaching their intended targets.

Most of the major studies on anti-phish filters have focused on English or other European languages with some exceptions such as the work of [52] that was for Japanese; [53] for Chinese; [54]–[57] for Turkish and some other works, like French, Russian, Italian, and Lithuanian [24], [58]. Although the number of non-English phishes is rising, there has been insufficient research in some widely used languages. In particular, Arab word users are likely to get the phish in either English or Arabic or also, bilingual. Arabic is the native language of more than 300 million people and is ranked as the fifth most extensively used language throughout the world. Arabic is the 6th language with the highest number of speakers. For Arabic language phishing email detection, no work is still done in terms of content-based phishing email detection. This research aims to decrease this gap through the development of the English-Arabic Phishing Detection (EAPD) model which employs ML & NLP approaches. The EAPD model process in the phish filtering process of an email message body text in English or Arabic languages.

1.2 Motivations

The detection of phishing emails in Arabic holds significant importance in today's digital landscape [59]. As the Internet continues to connect individuals and organisations worldwide, cybercriminals are expanding their operations to target users who communicate primarily in Arabic [60]. According to a news report [61], Saudi Arabia leads the Gulf Cooperation Council (GCC) in terms

of phishing attacks nearly a million phishing attacks by the end of 2020, with Kaspersky's spam and phishing report for Q2 revealing 973, 061 phishing attacks in just three months. United Arab Emirates (UAE) (617,347), Egypt (492,532), Oman (193,379), Qatar (128,356), and Kuwait (106,245) were the countries with the most attacks after Saudi Arabia, with Bahrain having the fewest (67,581). This shows the importance of phishing attack in the Arabic context.

With a growing user base and online presence in Arabic-speaking regions, the need for effective Arabic phishing email detection becomes paramount. Phishing attacks are becoming increasingly sophisticated and tailored to exploit the vulnerabilities of specific target groups [62]. While English-language phishing emails have received considerable attention, the detection mechanisms and defence strategies must be extended to encompass the Arabic language as well [63]. By focusing on Arabic phishing email detection, the study addresses the unique linguistic and cultural aspects that make Arabic-speaking users susceptible to phishing attacks. Arabic, as one of the world's major languages, is used by millions of individuals for personal and professional communication [64]. However, the existing research and tools for phishing email detection predominantly target English content, leaving a significant gap in the protection of Arabic-speaking users. The rise in Arabic phishing attacks highlights the urgent need to develop robust detection methods tailored specifically to the Arabic language.

Phishing is a mind game that supports hackers by tricking the users. It is important to recognise the intention of the sender of the email as part of the detection and management of phishing attacks. In phishing detection, the information security personnel must involve themselves in the analysis of text in the email body. Fraudsters have been known to translate phishing emails from English into Arabic for various reasons [63]. Firstly, they may want to target Arabic-speaking audiences, as individuals and organisations who speak Arabic may be more susceptible to falling for phishing scams written in their native language. Secondly, by translating emails into Arabic, fraudsters can expand their reach and target individuals who may not be proficient in English. Thirdly, ML algorithms are commonly used to detect and block phishing emails based on certain criteria. By translating the emails into Arabic, fraudsters can evade these detection mechanisms and increase their chances of success. Fourthly, translating phishing emails into Arabic can also be a social engineering tactic as the recipient may be more likely to trust it and follow the instructions if it

appears to come from a legitimate source. Finally, fraudsters may use machine translation (MT) services to hide their identity and location by using a different language, making it harder for law enforcement agencies to track them down and hold them accountable for their actions.

The lack of research on Arabic spam/phishing email detection highlights the necessity for advanced solutions. Existing Arabic-centric studies [65]–[72] primarily focus on end-user education, but the challenges posed by Arabic's complex linguistic features and the absence of comprehensive datasets for machine learning impede the development of effective detection algorithms. There is a significant gap in applying natural language processing (NLP) and deep learning (DL) methods to Arabic phishing, particularly in exploring character-level analysis. Addressing these issues is crucial for enhancing cybersecurity in Arabic-speaking regions, safeguarding individuals and organisations from malicious emails, and contributing to a safer digital environment.

1.3 Research problem

The detection of phishing emails in Arabic presents significant challenges due to the lack of dedicated resources and tools tailored specifically for Arabic phishing email detection. Firstly, the scarcity of Arabic datasets for training ML models poses a major hurdle. Existing datasets primarily focus on English phishing emails, leaving a substantial gap in resources required to train ML models to effectively detect phishing attacks in Arabic. This scarcity inhibits the development of accurate and reliable detection models for Arabic phishing emails.

Another challenge is the complex nature of the Arabic language itself. Arabic exhibits rich linguistic nuances, including variations in grammar, sentence structure, and word formation. Consequently, accurately capturing the semantic meaning of Arabic text proves difficult. Understanding the intended meaning behind phishing email content, including the usage of deceptive language and manipulation techniques, is essential for effective detection. Overcoming this challenge and developing techniques to accurately capture Arabic semantic meaning is vital for enhancing the efficacy of phishing detection systems in the Arabic language.

Additionally, the absence of comprehensive stop word lists tailored specifically for Arabic text hampers the accurate analysis and classification of phishing emails. While stop word lists have been extensively developed for English, the lack of a comprehensive Arabic stop word list affects the precision and performance of phishing detection systems in Arabic. Constructing comprehensive stop word lists tailored to the Arabic language is crucial for improving the detection accuracy and efficiency of phishing emails.

Addressing these challenges and developing solutions to bridge the gaps in Arabic phishing email detection is essential. Overcoming the scarcity of Arabic datasets, capturing accurate Arabic semantic meaning, and constructing comprehensive stop word lists for Arabic text will enable the development of more robust and effective phishing detection systems specifically designed for Arabic-speaking users. By tackling these research problems, the study can contribute to improving the cybersecurity landscape and protecting Arabic users from the ever-evolving threats of phishing attacks.

1.4 Research questions

1) What are the challenges and issues in phishing email detection studies, and how can they be improved?
2) What are the characteristics of the datasets and resources available for phishing email detection studies?
3) Can English phishing email detectors classify phishing emails that have been translated from Arabic to English using Google Translate?
4) Can Arabic phishing email detectors classify phishing emails that have been translated from English to Arabic using Human translate?
5) How can feature extraction techniques be employed to enhance the detection of phishing emails in the Arabic language?
6) Is there a significant difference in accuracy between the phishing detection model using the original set of Arabic stop words and the model utilising additional Arabic stop words?

1.5 Research aim and objectives

This research aims to propose and develop an efficient EAPD model specifically designed for the detection of phishing emails in both English and Arabic languages. The EAPD model will focus on analysing the content of email message bodies written in English or Arabic to identify potential phishing threats. To achieve this aim, the research will involve the following key aspects:

1) To identify and address the challenges and issues in phishing email detection studies to enhance the effectiveness of detection methods.
2) To examine the characteristics of existing datasets and resources available for phishing email detection studies, aiming to understand their limitations and explore potential improvements.
3) To evaluate the effectiveness of English phishing email detectors in classifying phishing emails translated from Arabic to English using Google Translate as a translation method.
4) To assess the effectiveness of Arabic phishing email detectors in classifying phishing emails translated from English to Arabic using human translation as a translation method.
5) To explore and develop effective techniques for feature extraction to enhance the detection of phishing emails specifically in the Arabic language.
6) To investigate and compare the accuracy of a phishing detection model using the original set of Arabic stop words against a model that incorporates additional Arabic stop words, examining if there is a significant difference in performance.

1.6 Research hypothesis

This research probes a central hypothesis geared towards understanding the nuances of phishing email detection in Arabic, specifically when considering translations from English. The focal hypothesis delves into discerning the classification accuracy of Arabic phishing detectors, particularly when distinguishing between phishing emails translated from English to Arabic and those originally penned in Arabic. Through this hypothesis, the study intends to shed light on the challenges and distinctions that arise due to linguistic transformations. Ultimately, the study aimed to offer insights that can fortify cybersecurity measures for the Arabic-speaking community. The primary hypothesis under scrutiny is the following:

Hypothesis: There is a significant difference in the classification accuracy of Arabic phishing detectors when distinguishing between phishing emails that have been translated from English to Arabic and those that were originally written in Arabic.

1.7 Research methodology

This section provides an overview of the research methods employed in this thesis, which are crucial for conducting a comprehensive study and gaining valuable insights into the research topic. In terms of data resources, various research methods were utilised. Firstly, an English-Arabic parallel phishing email corpus was developed by translating text data from the IWSPA-AP 2018 anti-phishing shared task using human translation. This corpus served as the foundation for the research, providing authentic examples of phishing emails in both languages. To enhance dataset diversity, an additional collection of 300 genuine emails written in Arabic was created. These emails were translated into English using machine translation, enabling comparative analysis and testing. The resulting dataset, comprising phishing emails in both languages, proved valuable for testing and validating the system's performance in handling multilingual content, particularly in Arabic.

As for the methodology used for phishing email detection, a comprehensive model was proposed, leveraging text features based on email properties. The model combined various research methods to enhance detection accuracy. Feature extraction techniques at the word and character levels were employed to identify the most relevant attributes for phishing email detection. Two datasets were carefully selected: the English-Arabic parallel phishing email corpus from IWSPA-AP 2018 and the Arabic Phishing Email Corpus (APEC) derived from real phishing email cases. ML/DL algorithms, including classical models and DL models like CNN and RNN, were used to classify the selected features. Performance evaluation measures were applied to assess the system's effectiveness in distinguishing between legitimate, suspicious, and phishing emails. This comprehensive research methodology aimed to develop an effective model for phishing email detection and contribute to the advancement of strategies in combating phishing threats.

1.8 Contribution

The study makes several contributions to the field of phishing email detection. These contributions are outlined as follows:

1) **Building an English/Arabic phishing email corpus**: This research contributes to the creation of an English/Arabic phishing email corpus. By collecting and curating authentic phishing emails in both languages, the corpus provides a valuable resource for training and evaluating phishing detection models, especially in cross-lingual scenarios. This corpus facilitates further research and advancements in phishing email detection for English and Arabic languages.
2) **Developing an English–Arabic Phishing Detection (EAPD) model**: The study proposes the **EAPD** model, which is specifically designed for the detection of phishing emails in English and Arabic. By leveraging ML and NLP techniques, the **EAPD** model demonstrates an efficient approach to detect and classify phishing emails in both languages. This model contributes to the development of effective detection systems that can address the specific challenges and characteristics of phishing attacks in English and Arabic.
3) **Creating a new Arabic stop word list (Salloum's list)**: The research introduces a new Arabic stop word list called Salloum's list. This stop word list is tailored specifically for the Arabic language and enhances the accuracy and performance of phishing email detection by filtering out common and non-informative words. Salloum's list contributes to the improvement of Arabic text analysis and classification tasks, including the detection of phishing emails.
4) **Research publications**: The study has resulted in several research publications that contribute to the existing knowledge and understanding of phishing email detection. These publications include "Phishing Email Detection Using Natural Language Processing Techniques: A Literature Survey" [73], "Phishing Website Detection from URLs Using Classical Machine Learning ANN Model" [74], "A Systematic Literature Review on Phishing Email Detection Using Natural Language Processing Techniques" [59], and "A New English/Arabic Parallel Corpus for Phishing Emails" [63]. These publications provide insights, analysis, and evaluations related to phishing email detection and serve as valuable references for researchers in the field.

1.9 Thesis outline

To elaborate, the thesis has six chapters, and each chapter is elaborated in such a manner that it explains the message of each chapter and its subtopics, followed by the next chapter.

1) **Chapter 2. Literature Review**: This chapter introduces the issues related to phishing. A particular definition is mentioned, and an in-depth analysis of phishing losses and the impact of phishing attacks is provided. Along with studying phishing email detection, other anti-phishing techniques used by various researchers are comprehensively discussed.
2) **Chapter 3. English–Arabic parallel corpus generation for email phishing**: This chapter focuses on addressing the limited availability of linguistic resources for Arabic by presenting an English-Arabic parallel corpus of phishing emails. The corpus was developed by combining existing English and Arabic texts with a new collection of authentic phishing emails in Arabic. The chapter also highlights the unique features of the Arabic language and the challenges involved in developing linguistic tools for Arabic. It describes the process of corpus development and examination and how this corpus can serve as a valuable resource for testing and validating multilingual phishing email detection systems. Finally, the chapter concludes with a summary of the main findings.
3) **Chapter 4. A New Model for Detecting Arabic/English Phishing Attacks**: This chapter presents a model for detecting phishing emails using text features based on email content. The proposed model involves selecting an English-Arabic parallel phishing email corpus, followed by feature selection using CNN and FastText embeddings algorithms, and feeding the selected features into ML/DL classifiers. The training will be performed using classical ML & DL models. The best-performing algorithms will be used to differentiate between legitimate, suspicious, and phishing emails. The chapter concludes by outlining the experimental setup and the steps involved in the proposed model.
4) **Chapter 5. Discussion of the Results**: In this chapter, the final phishing email detection model is presented with a detailed explanation of its architecture. The effectiveness of the proposed model is evaluated through various experiments, which compared different classifiers and feature extraction methods in detecting and preventing phishing attacks. The chapter aims to provide a comprehensive overview of the final model and its performance, highlighting its ability to detect phishing emails with a reasonable error rate. The findings

of these experiments provide valuable insights into the most effective techniques for phishing detection, which can help improve the security of individuals, businesses, and governments against these malicious attacks.

5) **Chapter 6. Conclusion and Future work**: In this chapter, the limitations of the research will be discussed, including any potential constraints or shortcomings of the study. The conclusions of the research will be presented, highlighting the main findings, implications, and contributions of the study. Additionally, recommendations will be provided for future research in this area, including suggestions for improving the performance of phishing email detection systems. Finally, the chapter will outline possible directions for future research that can build upon the work presented in this paper and extend it further.

Chapter Two

Literature Review

2.1 Overview

This chapter aspires to execute a comprehensive systematic literature review (SLR), aimed at integrating and synthesizing the corpus of research pertinent to phishing email detection methodologies, which employ ML and NLP techniques. Guided by a pre-specified set of criteria, this endeavor promises a targeted and exhaustive perusal of existing literature. The focal points of the examination include the application of NLP for identifying phishing attempts, the role of ML algorithms in phishing detection, and the optimization techniques applied to improve the performance and accuracy of phishing detection methods.

Additionally, the exploration extends to an analysis of the textual features characteristic of phishing emails and a review of the datasets and resources previously employed in this research domain. Further, the evaluation criteria in phishing detection studies are scrutinized, with an assessment of the range of traditional and nuanced metrics used to measure effectiveness. Through this rigorous SLR, an illuminating overview of the current phishing email detection research landscape is aimed to be provided, potential gaps in understanding are identified, and promising paths for future investigations are suggested. This review study aims to address the following eleven SLR questions:

1) What are the key research areas in phishing email detection using NLP?
2) Which ML algorithms are used most for developing models for detection of phishing emails?
3) What are the main optimisation techniques used in detecting phishing emails?
4) What are the feature extraction methods in phishing email detection using NLP studies?
5) Which NLP techniques are used most in phishing email detection studies?
6) Which datasets and resources have been used in phishing email detection using NLP studies?

7) What are the evaluation criteria of the ML/DL techniques that have been used in phishing email detection using NLP studies?
8) What are the tools used in phishing detection email using NLP studies?
9) Which parts of the email are the most widely used in phishing detection email using NLP studies?
10) What are the trends across time in phishing email detection using NLP studies?
11) What proportion of phishing email detection studies are on mixed language models?

2.2 Process used to carry out the literature review

Conducting a thorough literature review plays a significant role in carrying out a research study. It helps in establishing an accumulated knowledge foundation which then leads to the enhancement and expansion of theories. As a result, research gaps are reduced and earlier research missed areas are uncovered [75]. In the present study, a SLR has been carried out and includes phishing email detection using NLP.

2.2.1 Method

The guidelines followed for performing a systematic review for the current review study can be found in [76]. The following four phases were employed to conduct the review: "identification of inclusion and exclusion criteria, data sources and search strategies, quality assessment and data coding and analysis" [76]. The following sub-sections present the details of these phases. The SLR techniques mentioned in [77] were also followed for this study for better organisation. The introductory procedure of establishing a review protocol is included in the acquired SLR method, whereas planning, carrying out and analysing the review are included in the review process. The following steps were used to conduct the review. The search was identified, the work quality was assessed, the main research was selected, the data was synthesised, the review was recorded, the data was extracted and finally, a verification was performed.

The six steps of the review protocol that are used in this survey are presented in Figure 2.1 In addition, the research question synthesis is an essential segment of the SLR approach since, at the beginning, it determines the terms of reference for this study. Then, the step that includes

combining a search strategy that focuses on establishing the initial studies is highlighted in Figure 2.1. Although this phase is achieved, there should be a way of defining the search terms/criteria and the initial studies must be related to the SLR.

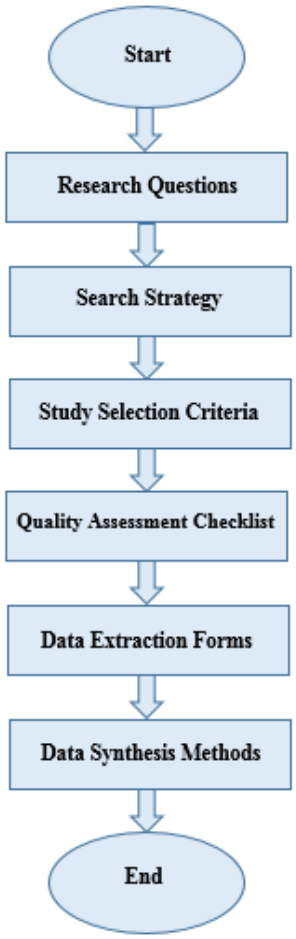


Figure 2.1 Protocol Review Stages [77]

2.2.1.1 Inclusion/exclusion criteria

The review study will involve the analysis of the articles that fulfil the inclusion as well as exclusion criteria stated in Table 2.1.

Table 2.1 Inclusion and exclusion criteria

Inclusion Criteria	Exclusion Criteria
Must involve phishing email detection	Articles without phishing email detection aim
Must involve NLP techniques	Articles without NLP techniques
Must be written in English language	Articles published in languages other than English
Must be published between 2001 and 2022	

2.2.1.2 Data sources and search strategies

The ACM Digital Library, Emerald, Google Scholar, IEEE, ScienceDirect, Springer, Taylor and Francis Online, and Wiley Online Library databases were used to thoroughly explore the extant studies to collect the research articles for inclusion in this systematic review. December 2020 marks the commencement of the search for the studies to be included in this systematic review. Search terms used during the search for relevant studies were based on keywords stated in Table 2.2. Appropriate selection of keywords is imperative for the selection of articles for inclusion in the review, since these keywords serve as the basis for access to relevant articles [78]. The search results obtained through the use of the previously stated keywords allowed access to 1125 articles (see Table 2.3), including 315 duplicate articles which were excluded through filtration. Consequently, 810 articles were obtained. Each study was evaluated by the authors against the inclusion and exclusion criteria, with 100 articles fulfilling the inclusion criteria, which were subsequently included in the analysis process. The preferred reporting items for systematic reviews and meta-analysis (PRISMA) was followed during the searching and filtration stages of the articles for the current review study [79]. The PRISMA flowchart is depicted in Figure 2.2.

Table 2.2 Keyword search

Keyword search
“Phishing” or “Malware" or “Malicious” or & [“detection” or “approaches” or “methods” or “attack”]
“Phish email” or “Malware email" or “Malicious email” & [“detection” or “approaches” or “methods” or “attack”]
“Phish e-mail” or “Malware e-mail" or “Malicious e-mail” & [“detection” or “approaches” or “methods” or “attack”]

Table 2.3 Final search results across the databases

No.	Database	Count
1	ACM Digital Library	69
2	Emerald	20
3	Google Scholar	348
4	IEEE	174
5	SAGE Pub	21
6	Springer	266
7	ScienceDirect	171
8	Taylor and Francis Online	15
9	Wiley Online Library	41
Total		**1125**

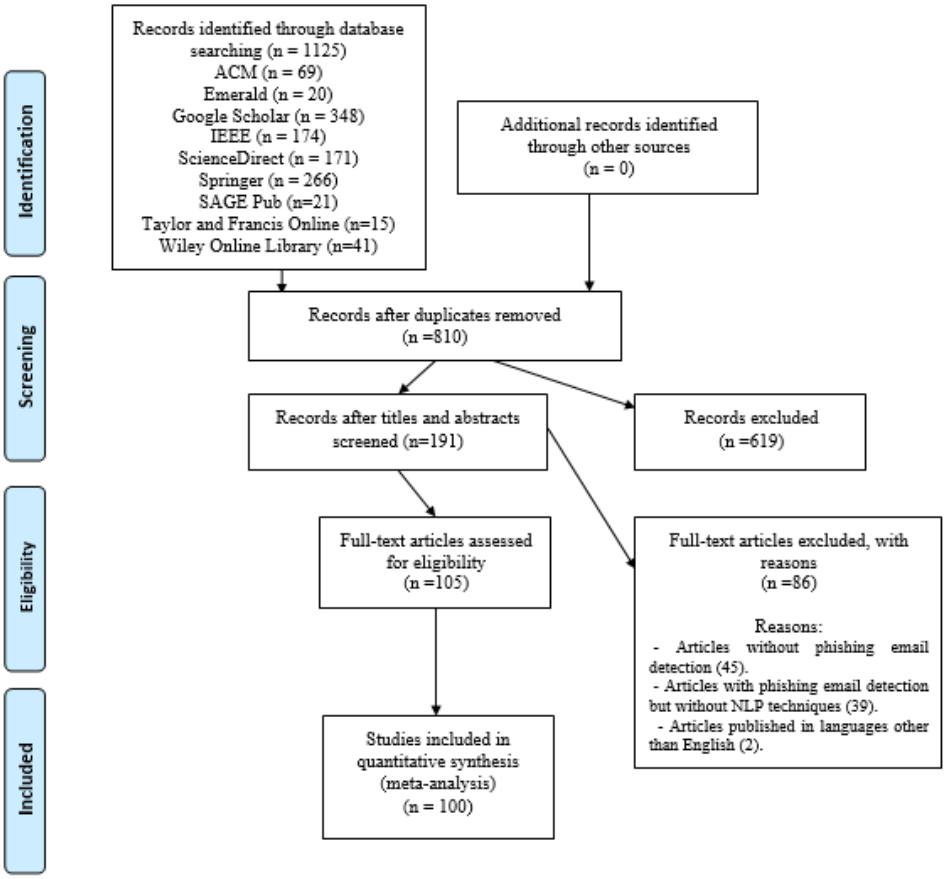


Figure 2.2 PRISMA flow diagram

2.2.1.3 Quality assessment

The factor of quality assessment is as important as the inclusion and exclusion criteria [80]. Seven criteria stated in the quality assessment checklist were employed for assessment of quality of the research articles qualified for inclusion in further analysis after filtration (N = 100). Figure 2.3 shows the quality assessment checklist. The checklist was a modified form of recommendations of [76], and did not mean to disapprove the work of any of the scholars [76]. A 3-point scale was taken as a standard for scoring the questions, whereby 1 point was assigned to 'Yes'; 0 points assigned to 'No' and 0.5 points assigned to 'Partially'. The range of points that could be scored by any study was 0 to 7. The greater the total score acquired by the study suggested a higher degree of the ability of the study to give responses for the research questions. The outcomes obtained from the quality assessment of each study are shown in Appendix Table A8, which indicates the fulfilment of quality assessment criteria by all studies, thereby suggesting the eligibility and qualification of all 100 studies in further analysis.

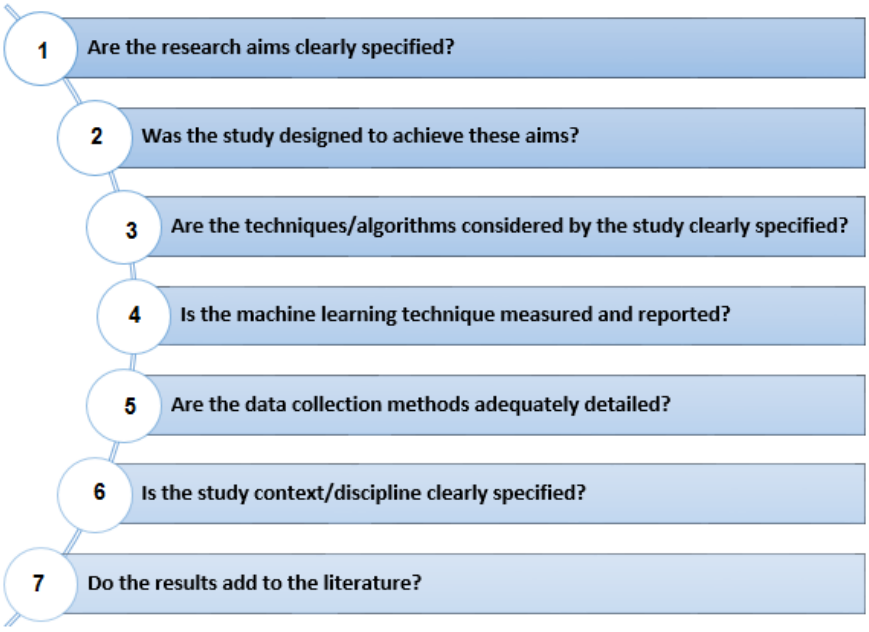


Figure 2.3 Quality assessment checklist [81]

2.2.1.4 Data coding and analysis

For each study identified, the following information was recorded (a) year of publication (b) the main key research area in phishing detection using NLP techniques (c) research techniques (e.g., AB, BN, CNN, etc.), (d) optimisation techniques (eg., Adam optimizer, Cuckoo search algorithm (CS), etc.), (e) text features, (f) NLP techniques used in phishing email detection studies, (g) datasets and resources, (h) evaluation criteria, (i) tools used in phishing detection email using NLP studies, (j) and the parts of the email most widely used in phishing detection email, using NLP studies.

2.3 Phishing email detection

2.3.1 Background

2.3.1.1 Phishing definition

The aim and the scope of the approaches of phishing detection can be examined through the definition of phishing that has been implemented by such approaches. Consequently, when the background of the various definitions of phishing is presented, it can assist the readers in understanding the scope and competencies of these approaches. The literature includes several definitions of phishing which are summarized in Table 2.4. It provides definitions by PhishTank [82], the APWG [83], Xiang *et al*. [84] and Ramesh *et al.* [85] that include the maximum number of cases in which the phishers' goals have been stealing sensitive personal information like the authentication credentials [86]. Table 2.4 shows a comparison of those phishing definitions established on the target and strategy of phishing. The most leading phishing strategies are social engineering (e.g., through fraudulent emails) and technical subterfuge (e.g., malware infection) [86]. On the other hand, classic techniques (e.g., pharming [87]) are also used to yield personal information of the users from the Internet [88]. On the contrary, the definitions of Whittaker *et al.* [89] and Khonji *et al.* [90] are not bound to the attacker's target (e.g., sensitive personal information). They define the phishing strategy (e.g., phishing website or socially engineered messages) without affirming to a precise phishing target (e.g., only state the attackers' benefit). With no scientific agreement, the other sources might deliver a standard definition [91]. In order to initially find the definition of the word, the dictionary is considered the primary source. Table 2.4 shows three definitions from prominent English dictionaries. Furthermore, it lists the definition

of the APWG, a non-profit foundation that keeps a record of phishing [91]. The definition of the APWG is lengthy compared to its dictionary definition. The five definitions differ in the level of detail and range of the phenomenon. For example, the American Heritage definition consists of phone calls, whereas the others do not. Additionally, the aim of phishing varies in the definitions, fluctuating from financial account details (Collins, APWG) to more general personal information (Oxford, Merriam- Webster, American Heritage) [91]. One bank may consider a fraudulent phone call as phishing, whereas another bank may not. Therefore, oppression or countermeasures can be hardly assessed. Consumers may also suffer from the downside of a lack of a standard definition. It is difficult for people who are less computer literate to understand phishing. The aim is to clarify the definition of the phishing phenomenon by analysing the already present phenomenon in comparison to most of the standard definitions that have already been established by experts. The generated definition is dependent on the consensus that is illustrated through the literature, and is enough for assisting further development. In fact, several academics have characterised phishing since the inception of phishing attacks; however, their interpretations differ and often do not coincide. Lastdrager [91] stated that "phishing is an extensible attempt to induce in which imitation is utilized to collect information from a target" after reviewing numerous phishing definitions. To summarise this, the definition of Whittaker *et al*. [89] is considered to be the most general, while APWG [83] defines the most frequently used phishing attacks in a precise manner.

After conducting a comprehensive analysis of existing phishing definitions mentioned in a number of research studies [84], [85], [97], [89]–[96], my understanding of a phishing email has evolved significantly. Phishing is now recognised as a deceptive cyber tactic that involves impersonating trustworthy entities to fraudulently obtain sensitive information, including personal, financial, or login details, from unsuspecting individuals. This expanded comprehension underscores the intricate nature of phishing, emphasizing its reliance on social engineering and digital deception to exploit human vulnerabilities, especially in electronic communication.

Table 2.4 Most popular definitions of phishing

Source	Definition	Target	Strategy
American Heritage Dictionary (2013), USA [92]	To request confidential information over the Internet or by telephone under false pretences in order to fraudulently obtain credit card numbers, passwords, or other personal data.	Personal information	Not specified
APWG (2013) [93]	Phishing is a criminal mechanism employing both social engineering and technical subterfuge to steal consumers' personal identity data & financial account credentials.	Identity data, financial account credentials	Social engineering
Collins English Dictionary (2013), UK [94]	The practice of using fraudulent e-mails and copies of legitimate websites to extract financial data from computer users for purposes of identity theft.	Not specified	Not specified
Khonji et al. [90]	Phishing is a fraudulent act to acquire sensitive information from unsuspecting users by masking as trustworthy entity in an electronic commerce.	Not specified	Social engineering
Lastdrager (2014) [91]	Phishing is a scalable act of deception whereby impersonation is used to obtain information from a target.	Not specified	Not specified
Merriam-Webster (2013), USA [95]	A scam by which an e-mail user is duped into revealing personal or confidential information which the scammer can use illicitly.	Personal information	Not specified
Oxford University Press (2014), UK [96]	The fraudulent practice of sending emails purporting to be from reputable companies in order to induce individuals to reveal personal information, such as passwords and credit card numbers, online.	Personal information	Not specified
PhishTank [97]	Phishing is a fraudulent attempt, usually made through email, to steal your personal information.	Personal information	Social engineering
Ramesh et al. [85]	Phishing is a type of computer attach that communicates socially engineered messages to humans via electronic communication channels in order to persuade them to perform certain actions for the attacker's benefit.	Sensitive information	Not specified
Whittaker et. al. [89]	Phishing page is any web page that, without permission, alleges to act on behalf of a third party with the intention of confusing viewers into performing an action with which the viewer would only trust a true agent of the third party.	Not specified	Not specified
Xiang et al. [84]	Phishing is a form of identity theft, in which criminals build replicas of target Web sites and lure unsuspecting victims to disclose their sensitive information like passwords, personal identification numbers (PINs), etc...	Sensitive information	Not specified
Salloum's definition	Phishing is a deceptive cyber tactic that involves impersonating a trustworthy entity to fraudulently obtain sensitive information, such as personal, financial, or login details, from unsuspecting individuals.	Sensitive information	Social engineering and technical subterfuge

As previously stated, a phishing attack begins with an email sent to an online customer. Crimeware is a kind of malware that is defined as a software which accomplishes illegal activities that are expected to generate monetary gains for the assailant [98]. The technical subterfuge schemes are generally comprehended by the users' actions like opening an attachment (see Figure 2.4).

Four steps are normally followed by the malware's activities: betraying a user to activate it, blocking technical defence, attaining its purposes, and lastly propagating [99]. Considering, for example, when a user opens an attachment file in an email, a keylogger can be installed, or when the link is clicked, the user can be readdressed to the phishing website by DNS attacks. So, the main part of phishing is to deceive the users by giving fake information and bait them to achieve actions in favour of foes.

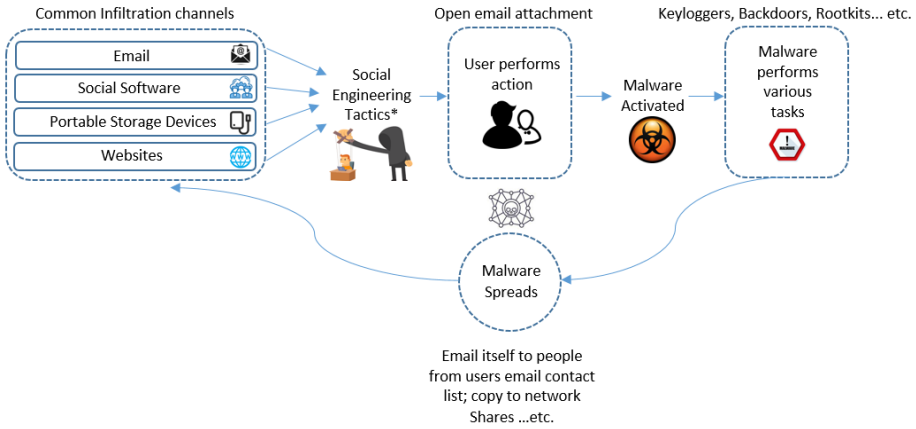


Figure 2.4 Steps taken by malware to infiltrate a system (adapted from [99])

Phishers offer potential victims with fake situations where the users are advised to execute a specific type of significant activity. Some examples of the typical situations are as follows: a user's webmail storage is almost surpassing the limit, a user's bank account is required to be updated because of some security measures, a user's online transaction has not been managed because of inappropriate information that the user entered while the goods are being purchased, etc. [100].

2.3.1.2 The origin of phishing

As mentioned above, phishing is considered a crime that uses both social engineering and technical subterfuge to acquire the personal identity data and financial account credentials of consumers [101]. The schemes of social engineering look for unguarded victims to fool and let them believe that they are doing business with a trusted, legitimate party, for instance, by making use of illusory email addresses and email messages [102]. These are considered to lead the consumers to forge the websites that deceive the recipients into revealing their financial data, such as their usernames and passwords [102]. The scheme of the technical subterfuge employs malware onto computers in order to steal the credentials directly, usually by employing the systems that interrupt the accounts of the consumers' usernames and passwords, or mislead them towards forging the websites [2]. The Interex conference was the first to present the notion of 'phishing' in 1987 [103]. The word phishing originated from the analogy that malicious Internet users bait to 'fish' for credential information from the sea of Internet users by utilising emails [2]. In January 1996, the Internet of phishing was first cited on the alt.2600 hacker newsgroup, or might have been ongoing from the printing of the hacker newsletter '2600' that came earlier [102]. The term 'phishing' started being used in 1996 to define the incidents where hackers were misusing passwords through the unsuspecting America On-Line (AOL) user to steal AOL accounts [102]. Today, this term has been extended and is composed of different attacks to target the personal information [104]. Since originating from 'fishing', therefore it is always spelled as 'phishing' to distinguish it from the original term, and to probably match the phone 'phreaking'. When the definition of phishing is considered, the derivative noun, 'phisher', denotes the committer of the crime. The hackers substituted 'f' with 'ph', and the primary form of hacking was known as 'phreaking'. The word phreaking was initially implemented by the first hacker John Draper, who invented the notorious Blue Box with which he was able to hack telephone systems in the early 1970s [102], [105]. The first form of hacking is known as 'Phone Phreaking' which was the origin of the 'ph' spelling in the organisation of hackers. By 1996, the accounts that were hacked by the criminals were known as 'phish'. This phish went into trading among the hackers. Afterwards, the intensity of the number of phishing attacks kept increasing, and the criminals grew their area of activity from simply stealing AOL accounts towards targeting users of online banking and e-commerce sites [105].

2.3.1.3 Phishing life cycle

As previously stated, a phishing attack begins with an email sent to an online customer (see Figure 2.4). This email contains a fraudulent link that redirects the user to a fake website, which is cloned by the attacker to seem exactly like the original website on which it is based. This persuades the gullible email recipient of the email and website's legitimacy. Figure 2.5 depicts a phishing email and its essential components. This information was acquired from the University of Massachusetts Amherst [106] and will help show how to protect internet users against deceit. The features of the fake website are depicted in Figure 2.6, where the email recipient is required to supply confidential information, which the attacker then obtains and uses illegally.

1) Despite claiming to be 'UMass Amherst<it@umass.edu>, the sender is actually not associated with UMass and the actual email address isn't one of the university's email addresses.
2) In a phishing email, it's easy to identify spelling and language mistakes. In this email, for example, there is a comma before a colon, which is incorrect.
3) A phishing email will also include language that generates a false feeling of urgency. This prompts the recipient to take action without much deliberation.
    - The threat of a 'permanent' error if the receiver takes too long to act is one example found in this email.
4) The message link is crafted to look like an authentic UMass Amherst page address. However, hovering over it reveals that it leads to a different page.
    - Hovering over a link reveals where it leads, which in this case is not to a trusted UMass website. As a result, double-check the links before clicking them!
5) Another flaw in the mail is that it claims to be from both UMass Amherst and Microsoft Corporation. The sender is a phony if they are unsure of their own identity and affiliation.
6) The link included in the message leads to a fake SPIRE login page with the web address being "tantechholdings.com".

Figure 2.5 Fresh Phishing email example [106]

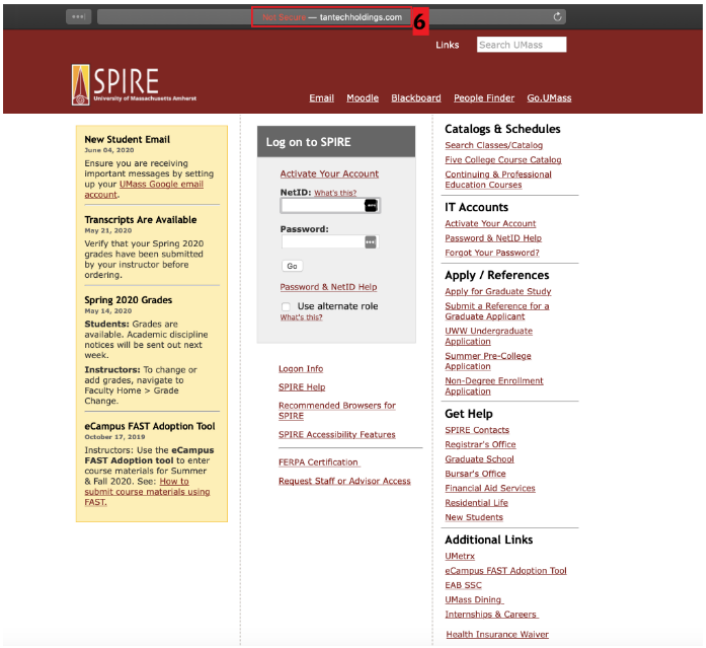


Figure 2.6 Phishing website with annotations [106]

2.3.2 Taxonomy of email message

Filtering e-mails is a method of distinguishing between legitimate and phishing email messages. This technique uses either a phishing e-mail filter, which examines and categorises e-mails into their appropriate groupings, or a learning-based filter which analyses a collection of labelled coaching data (previously collected messages with upright evaluations) [107], [108]. Another method of analysing e-mail messages is to examine each one individually for the existence of any unique words. Any degree of associated degree e-mails are divided into two sections: the body and the header. The e-mail headers contain several fields, such as from, subject, to, and so on [108]. The header lines not only give information about the message's subject, receiver, and sender, but they also give explicit routing data. The body of the email follows the header lines and contributes to the message's content. Figure 2.7 depicts the structure of an e-mail, and Figure 2.8 illustrates the structure of an e-mail message for the purposes of feature extraction and selection [109].

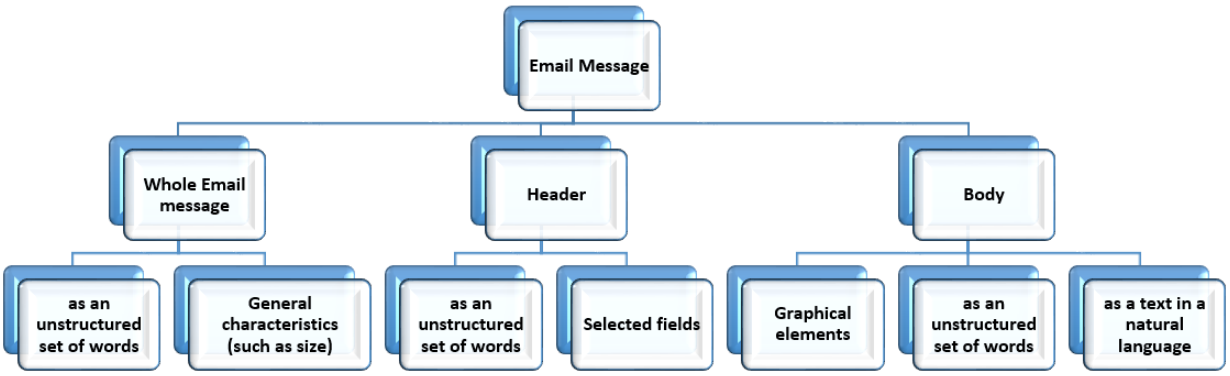


Figure 2.7 Taxonomy of email message structure (adapted from [110])

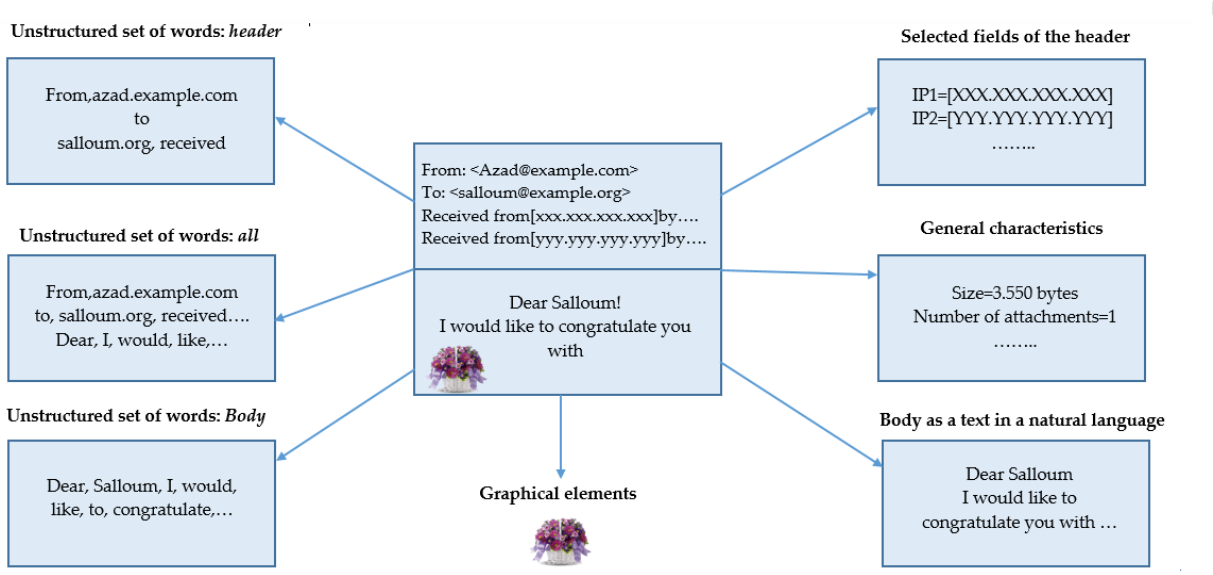


Figure 2.8 An example of the structure of e-mail messages (suggested for purposes related to feature extraction and selection (adapted from [110])

To comprehend the various strategies of e-mail filtering, it is critical to gather operating data regarding the format and structure of an e-mail [107]. This also aids in the identification of the pre-processing stage. The employment of envelopes in this modern method, similar to ancient communication mail, is an interesting feature. An e-mail envelope is not visible to the naked eye since e-mail systems remove it before delivering the e-mail message. Figure 2.9 shows an e-mail envelope and source code for an e-mail, respectively. To identify extra envelope recipients, more RCPTs are used.

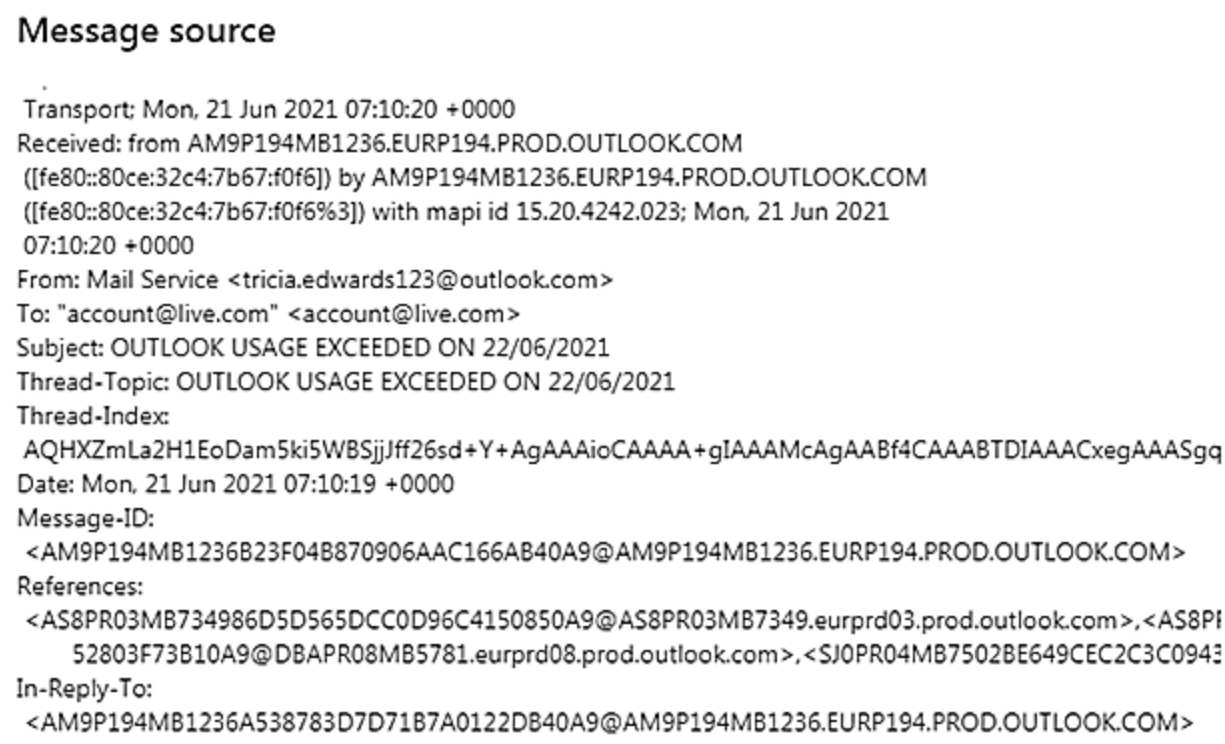


Figure 2.9 E-mail envelope and source code

2.3.3 Features for detecting phishing emails

Phishing emails are sent with the intent of stealing personal information from the recipients. The majority of users fell victim to phishing attacks as a result of their careless internet surfing. Companies should educate their staff about phishers' traps and strategies. In this section, the methods of defending against phishing attacks and identifying phishing will be discussed. To weed out phishing emails, some spam filters employ hundreds of features. These features [111] for detecting phishing emails are classified as follows:

1) Email body-based characteristics: These attributes are taken from the email body. They have binary features like shapes, HTML, and specific phrases and links in the email body.
2) Subject-based features: Certain aspects of an email are derived from its subject, such as whether it is a reference to a previous email or the use of terms like verify or debit.

3) URL-based characteristics: These attributes examine when an IP address is used instead of a domain name, the inclusion of @ in links, the number of photos, external and internal links in the email document, the number of cycles in links, and so on.
4) Script-based features: These features look for JavaScript, pop-up window code, on-click activities, and other script-based features in the email.
5) Sender-based characteristics: These characteristics provide information about the sender, such as the difference between the sender's address and the reaction to the address.

2.3.4 ML techniques

Figure 2.10 depicts the various phishing email detection methods utilised in the literature, and also the volume of publications that use each method. The most prevalent phishing email detection algorithms are supervised approaches, such as SVM [13], [16], [19], [22], [23], [26], [27], LR [14], [19], [22], [26], [29]–[33], DT [11]–[13], [15], [17], [22]–[25], and NB [18], [26], [38], [112]. Unsupervised approaches such as k-means clustering [22], [34], [113]–[116] and DL methods have also been adopted [18], [19], [48], [49], [117]–[120], [20], [21], [28]–[30], [45]–[47].

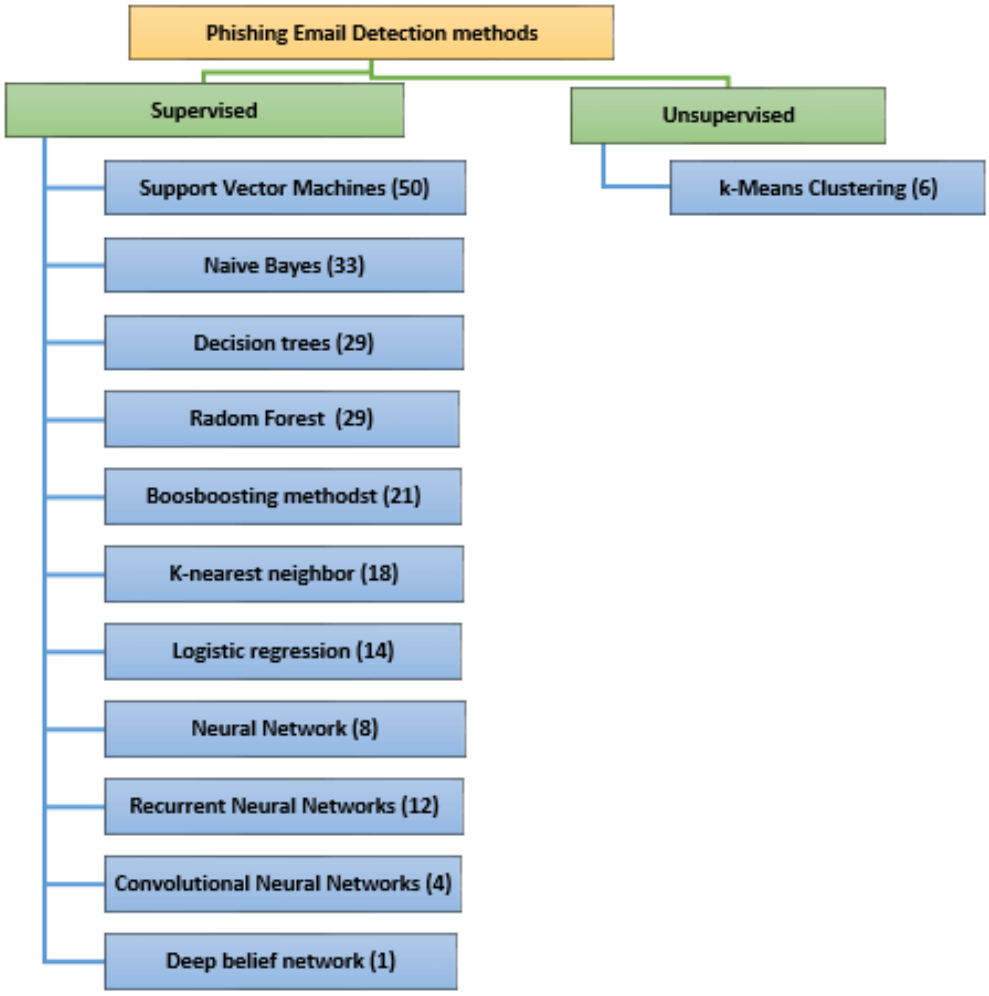


Figure 2.10 Methods used in phishing email detection

2.3.4.1 Supervised classical ML algorithms

2.3.4.1.1 Decision Tree (DT)

A commonly used ML algorithm that can be applied for regression and classification is the DT. A recursive partitioning algorithm is applied to test the availability of attributes or features considering specific purity indexes [121]. The Gini Index and Entropy are the most commonly used indexes, where the former is applied to measure the probability of a randomly chosen feature that is incorrectly classified [13]. The uncertainty amount that is proportional to the information

gain is referred to as Entropy [13]. By means of these indexes, the required position of the features, either internal node or root, can be determined. The DT can be applied to the categorical or continuous variables. Instances of research in the literature using DT are [13], [22], [23], [56]–[62], [64], [67], [75], [87]–[101].

2.3.4.1.2 Random Forest (RF)

A RF is an ensemble classifier that makes predictions using a variety of DTs. It works by fitting a variety of DT classifiers to different subsamples of the dataset. In addition, each tree in the forest was constructed using a random selection of the best attributes. At the time of the training phase, the DTs are created (as defined by the developer), and these are applied for the class prediction use. They are attained through consideration of the voted classes for each specific individual tree, and the class which attains the highest number of vote is considered the output. Similar issues within literature are resolved using the RF method [11], [13], [39], [44], [50], [51], [117], [119], [122]–[125], [14], [126]–[134], [15], [16], [24], [26], [33], [35], [37]. RF details can further be attained using [135], [136].

2.3.4.1.3 Naïve Bayes (NB)

The Bayes rule of conditional probability is applied by this classifier, and all data features are applied. They are individually analysed based on the assumption that they are not only independent but also as important as one another. Quick convergence and simplicity are the classifiers benefits, yet it is not possible to understand the associations and interactions amongst the features of each of the samples. The following papers [11], [14], [34], [35], [37]–[41], [44], [47], [65], [18], [117], [119], [123], [126], [127], [131], [137]–[140], [23], [141]–[143], [24], [26], [28], [31]–[33] have reported the use of NB to enhance the textual features in phishing email detection.

2.3.4.1.4 Support Vector Machine (SVM)

SVM is usually applied for classification activities as well as regression activities. Each data item within the SVM is plotted as the point within the n dimensional space (*n* is the feature number for each sample within the training set). The mission of the algorithm is to extract the most appropriate

hyper-plane which can be split into two classes. The nonlinearly separable data is classified by SVM through transformation into higher dimensional space, with the help of a kernel function, in which a separating hyperspace is present. Yet, it is difficult to interpret the SVM, and it is quite memory sensitive. It was noticed that numerous scientific papers, such as [11], [13], [27], [31]–[35], [37], [39], [41], [44], [14], [47], [65], [116], [117], [119], [125]–[127], [130], [131], [16], [133], [137]–[140], [143]–[147], [18], [148]–[157], [19], [22]–[24], [26], have utilised the SVM algorithm to detect phishing emails.

2.3.4.1.5 Neural Networks (NN)

The structure of the NN is formed by a set of interconnected identical units (neurons). Through these interconnections, signals are sent from one neuron to another [158]. Furthermore, weights are attached to the interconnections so that delivery between the neurons is enhanced [159]. On their own, the neurons aren't powerful; however, when they are connected, complex calculations can be carried out. At the time of network training, the interconnection weights are updated, therefore, during the testing phase, interconnection plays a significant role. The NN example can be observed in Figure 2.11. Within the figure, the NN includes "an input layer, hidden layer and output layer". The network is referred to as feedforward as the interconnections do not skip or loop back to the rest of the neurons. The nonlinearity present within hidden neurons helps provide the NNs power. Furthermore, the network must include nonlinearity so that complex mapping can be learnt.

The NN model fitting needs experience, even though it is competitive as part of the learning ability. The local minima are quite standard, and there is need for delicate regularisation. Many papers [14], [31], [65], [116], [127], [141], [34]–[37], [40], [42]–[44] have used NN differently.

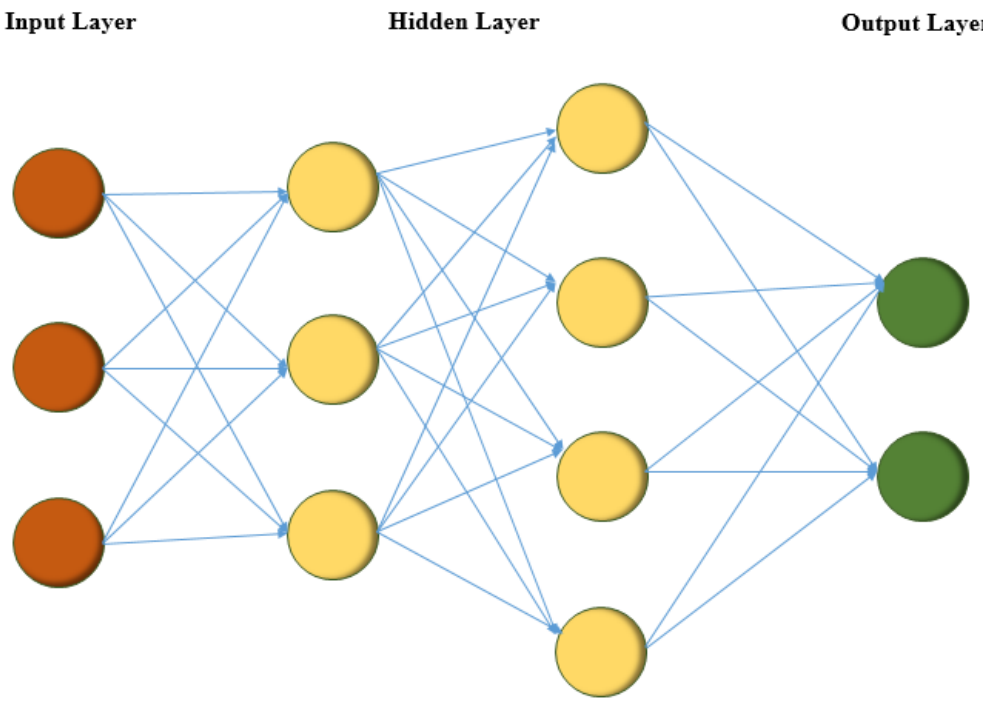


Figure 2.11 Neural Network

2.3.4.1.6 Linear Regression (LIR)

LIR is a supervised learning ML technique. It carries out a regression task. Based on independent variables, regression models a goal prediction value. It is mostly utilized in predicting and determining the link between variables. Few researchers [26], [32] have used LIR to train and test their models to detect a phishing emails.

2.3.4.1.7 K-Nearest Neighbours (KNN)

A commonly applied supervised learning algorithm is the KNN, which usually helps in classification. The assumption here is that similar aspects maintain close proximity. Similarity measures are applied to check for the similarity degree, most commonly the Euclidean distance. Implementation is easy with KNN, as tune parameters and model parameters are not built. The

KNN is referred to as a non-parametric algorithm, which is why fundamental assumptions regarding the distribution of data are not required. The algorithm will perform slower based on the increase in size and dimensionality of the dataset. It was noticed that several studies, such as [13], [14], [44], [49], [65], [125], [126], [130], [131], [137], [160], [15]–[17], [23], [25], [26], [28], [32], have employed the KNN algorithm in phishing email detection.

2.3.4.2 Supervised DL algorithms

DL is an ML branch that uses multilayer artificial neural networks (ANNs) to achieve state-of-the-art accuracy in complicated problems such as computer vision [161]–[163], speech synthesis [164], [165] and recognition [166], [167], language translation [168] , and several others such as fraud detection [169], [170]. DL differs from classical ML methods, in that it has the unique capacity to learn depictions instantly from a variety of data types like audio, video, text, or images without the requirement for hand-written constraints or subject technical expert knowledge. Because of the adaptable design, they can learn straight from raw data and improve prediction accuracy as more information is available. GPU-powered inference systems are necessary to improve performance and provide low latency inference for the computationally demanding deep neural network (DNN). The most often used DL models are CNNs and RNNs.

2.3.4.2.1 Convolutional Neural Network (CNN)

Various convolutions layers along with nonlinear activation function such as ReLU are referred to as the CNNs. As compared to the traditional NN where the layers are fully connected, the CNN convolution upon the input is carried out for computation of the output, and it provides the outcome of a local connection. For each layer, there is a significant number of filters that are applied, and its output is combined to attain outcomes. At the time of the training phase, the CNN learns filter values. In the case of NLP tasks, the CNN input are documents or sentences. A matrix row is used to represent the character or word, and this provides the vector which is aligned to the word, referred to as word embedding. The matrix column space is stated by the embedding dimension. The CNN differences amongst NLP and image is the choice of filter and size. For images, the filter is the slide over the input's local patch, but in NLP it will slide over the complete row, as the word is represented entirely. Hence, the filter matrix column space would be similar to the input matrix

column space [171]. It was noticed that several studies, such as [18]–[21], have employed the CNN method.

2.3.4.2.2 Recurrent Neural Networks (RNNs)

The hidden sequential associations in variable-length input sequences are learned by a RNN, which is mostly utilised for sequential data modelling. Many noteworthy successes in the areas of NLP and speech synthesis and recognition can be attributed to recurrent NN methodologies [30]. The RNN, on the other hand, has a long-term dependency issue, which could exacerbate the gradient exploding and vanishing issues. Polymorphisms of RNN have been developed to overcome the difficulties with it, one of which is the long short-term memory (LSTM) [48], [172]. To achieve the goal of learning this "long-term dependence" data, the LSTM utilises gates on the input and recurrent input to influence the state and also output at multiple intervals. Similar to the convolutional network, the LSTMs needs the same size inputs, hence, for the network input, it is only necessary to have the initial N email words [18]. RNN have been utilised by several academics [18], [19], [29], [30], [45]–[49] to train and test their phishing email detection models.

2.3.4.3 Unsupervised learning algorithms

2.3.4.3.1 K-means clustering

The technique applied for dataset partitioning or clustering into *k* groups is referred to as cluster analysis. Random selection of the *k* data points (clusters) is done and then passed through an iteration series using the mentioned methods.

1) For a specific word, *w*, they will be aligned to the closest cluster centre *Cj* with $1 \leqslant j \leqslant k$.
2) Each cluster centre (*Cj*) value will be updated using the mean value from the words that are part of the cluster [173].
3) Till the time the cluster cannot be changed any further, the algorithm will continuously run.

Usually, topic modelling term frequency–inverse document frequency (TF-IDF) and clustering procedures like the k-means are complementary, and may be used integrated, specifically the TF-IDF vectorisation as the precursor to k-means clustering, in order to present an in-depth assessment

[115]. Earlier research studies, like Ruiz-Casado *et al.* [174] and Rong [175] have been carried out with the help of Wikipedia keywords to indicate that the TF-IDF vectors and clusters of words outcomes are strictly aligned with the groups expected, allowing them to be an effective article classification tool [115]. NLP, topic modelling, and clustering techniques were combined to analyse and assess the persuasive techniques/strategies used by cybercriminals when fraudulent emails are created (Stojnic *et al.* [115]).

The experimental results indicate that when these techniques are applied, it is possible to understand the mindset of the cybercriminals along with the ability of the techniques to attain consistency, even though there has been a strategy evolution from early scams towards the modern phishing emails.

2.3.5 Feature extraction

Feature extraction is the process of converting raw data into numerical features that may be processed while maintaining the information in the original data set. It yields better results than just applying ML to raw data [176]. There are some techniques to extract features from text like principal component analysis (PCA) and latent semantic analysis (LSA), in which the input data is DTM, keeping in mind the TF-IDF measure (*F*). The following are the primary perspectives.

2.3.5.1 Principal component analysis (PCA)

PCA, as stated by [177], aims to extract mapping from within inputs of the original dimensional space towards the developed smaller dimensional space, ensuring minimum information loss [14]. Using the available data structure, it is possible to understand the structure and extract them through eigenvectors and eigenvalues, which help maximise the projected data variance and spread them out over the new dimensional space [14]. The objective of the technique is to alter the variables that can be correlated, into the linearly uncorrelated variables and the principal components by applying the orthogonal transformation, as stated by [178]. The principal component direction is represented by the eigenvectors, and the direction variance is brought forward by the eigenvalues. It was noticed that several studies, such as [14], [33], [34], [126], [139], [179], have utilised the PCA technique in phishing email detection.

2.3.5.2 Latent semantic analysis (LSA)

For NLP, the mathematical procedure applied is the LSA. Its objective is to embed the topics within the input data (documents) explicitly, extracted from the highest values, and attained based on the feature number required [14]. Removal is conducted for the features not selected and having values lower than the threshold value. They are not used in the following activities. The input data for CS and the mutual information (MI) measures is DTM, keeping in mind the TF-IDF measure, (*F*) are applied. These are univariate feature selection procedures [14]. The initial one is the CS to measure the linear dependency amongst the two random variables (input feature and target). The second is the MI, which integrates the nonlinear associations amongst input features considering target and analysis [14]. Many studies [14], [34], [179] have used LSA differently.

2.3.5.3 Chi-Square (Chi-S)

A commonly used feature selection procedure is the Chi-S ($\chi 2$) [180] which assesses individual features through computation of Chi-S statistics within the context of classes. Hence, the Chi-S score can analyse the term and class dependency [181]. If the class and term are independent, then the score would be 0 otherwise 1 [181]. The score is informed depending on the term having a high Chi-S. It was observed that some studies, like [14], [179], have used the Chi-S technique in phishing email detection.

2.3.5.4 Mutual Information/ Information Gain (MI/IG)

The measure used for the quantification of mutual dependence amongst two variables is MI. It is based upon random variable entropy (within information theory). Through MI, the information amount that is attained within the random variable is calculated, using a different random variable. Considering the work proposal, the information of each feature is identified, thereby stating whether the email is phishing or legitimate [178].

A procedure that uses the information gain measure to rank features can be used to select the most useful features [182]. For the metric, a threshold is then decided, and the attributes are kept with a value attached—only the top-ranked ones are kept. Furthermore, the features are selected by information gain through scores. Such a technique remains simple as compared to the earlier. The

concept is that each feature score is computed, which then indicates the class discrimination and the features are then sorted based on the score. The top-ranking ones are the only ones retained. MI has been utilised by several researchers [14], [142] to extract/select the features in phishing email detection.

2.3.6 Tools and techniques

This section discusses the numerous tools used for experimental purposes as well as the evaluation of an anti-phishing system's accuracy. A researcher's tool selection is influenced by a variety of parameters and algorithms. Figure 2.20 shows several tools that can be used for phishing detection evaluation. Python is the most commonly used one for phishing email detection [87], [91], [98]–[100], [183], [184]. Table 2.5 delves deeper into these tools and their uses in many sectors.

Table 2.5 Most popular tools of phishing detection

No.	Tool	Description
1	Python	One of the easiest to learn and most valuable programming languages is Python. Python is a sophisticated language with enhanced data structures and a straightforward approach to object-oriented programming. Its refined syntax, dynamic typing, and interpreted semantics make it a perfect language for scripting and quick application development across a variety of platforms [185].
2	Weka	Weka is a tool that provides a graphical user interface (GUI) that aids the functions of an algorithm by allowing the user to import a dataset and apply various functions/rules to the algorithm [186]. As a result, categorisation, regression, and grouping of algorithms are possible, as well as data visualisation and algorithm performance.
3	KERAS	Keras, a NN API, works with DL algorithms to provide simple and quick techniques [37], as well as CPU and GPU running features so models may be processed simultaneously.
4	TensorFlow	TensorFlow is a Google-developed end-to-end ML platform that allows users to run programs on multiple CPUs. This program includes GPU access, and the website is user-friendly for beginners as well as a learning tool for professionals [37]. TensorFlow can be readily combined with Keras to conduct DL experiments [187].
5	SCIKIT-LEARN	It is a library environment that provides not only a large selection of supervised algorithms appropriate for the project at hand [37], but also high-level implementation to train using 'fit' methods and 'predict' via an estimator or a classifier. Cross-validation, feature selection, feature extraction, and parameter tuning are among the other features available in this program [188].
6	NLTK	The Natural Language Toolkit, also known as NLTK, is a tool that generates interfaces for text processing, access to huge corpora collections, and linguistic structure. It is a Python package for NLP [189] that comprises libraries and programmes for parsing, chunking, tokenisation, PoS tagging, semantic analysis, clustering, and classification, among other NLP functions.
7	MATLAB	MATLAB is a high-performance technical computing language which combines features like computation, visualisation, and programming in a user-friendly environment whereby the issues and their solutions are written in recognisable mathematical notations [190].

2.3.7 Evaluation metrics

A confusion matrix depicts a table that shows a general summary of the classification and segmentation performance. Furthermore, certain binary classification problems need a two-group confusion matrix which is often utilised to present the positive and negative classes. There are four groups of the matrix included in this study, as follows: False negatives (FN), false positives (FP), true positives (TP) and true negatives (TN), as displayed in Table 2.6 [191]. These can be utilised to attain the four measures of classification performance. The complete negative incorrect predictions represent FN, the complete positive incorrect predictions represent FP, complete positive correct predictions represent TP and complete negative correct predictions represent TN.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2.1}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2.2}$$

$$\text{F1-measure} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.3}$$

$$\text{Accuracy} = \frac{(\text{TP} + \text{TN})}{(\text{TP} + \text{FP} + \text{TN} + \text{FN})} \tag{2.4}$$

To perceive these measures, the confusion matrix provided in Table 2.6 can be assessed.

Table 2.6 Confusion matrix

Confusion matrix	Predicted positive	Predicted negative
Actual positive	TP	FN
Actual negative	FP	TN

There are some more metrics recorded in certain papers, for instance, area under curve (AUC), and receiver operating characteristics (ROC) [12], [24], [29], [32], [35], [43], [128], [132], [179], [192]. Moreover, when taking into account imbalanced datasets, the utilisation of relevant evaluation metrics is an essential aspect with which to reckon. There are some grounds on which accuracy is unsuitable, for instance, asymmetric costs, base-rate fallacy, and imbalanced datasets. The same situation is seen with the ErR metric and in this plan, the preferential values should be the confusion matrix, ROC, and AUC. In addition, the metrics particularly suggested for imbalanced investigations must be utilised by the researchers [193], for instance, balanced accuracy, Matthews correlation coefficient (MCC), geometric mean, and balanced detection rate, and others. Although, certain papers utilise greatly imbalanced URL datasets, [194], [195], error rates [146], [154], make use of detection rates and malicious missing rates [196], and employ metrics in demand such as recall, *F*1-score, accuracy, precision, and ROC [197] utilising FP and FN rates, when the class grouping and dataset size is recorded, they can have an advantage of working out different metrics. Therefore, in the literature studies, mainly unsuitable imbalanced dataset metrics were noticed [198].

2.3.8 Dataset Properties

The datasets utilised by the authors [97], [199], [208], [209], [200]–[207] in order to test and train their models carry a vast impact on the credibility of their models, even though an essential feature of a suggested system is the detection process. Furthermore, the datasets utilised in website detection are nearly similar to the one utilised in the email detection methods, hence revealing the absence of a variety problem. In order to train/test the models, sometimes malware and spam emails are used, even though the papers are only regarding phishing email detection. These types of papers are categorised in the *malicious* class (*spam* dataset represents URLs taken from the body of spam emails). The papers that contain legitimate, phishing, and malicious sources are listed in Appendix Table A2.

The ground truth datasets are utilised by the different approaches, which they gather it from various cyber intelligence sources, and the evaluation is firmly combined with it. In addition, there are various testing methodologies which are been used by these sources. These sources also target various kinds of phishing activities, therefore shield various phishing domains. As seen, there is a

contrast between the evaluations relying on one dataset from another. For this reason, there is a debate on how essential it is to have a publicly available reference dataset in order to classify the evaluation of different approaches. Moreover, this essential part can present a benchmark in order to contrast the efficacy of different approaches, and eventually make it easier for the analyst to make improvements in the field in an additional systematic manner. However, the rerun of experiments for the systematic contrast of efficiency is difficult to achieve due to the missing of reference sets along with the complexity of code sharing. The determining features of the datasets utilised in the literature are listed in Table 2.7.

Table 2.7 Phishing email dataset features

No.	Dataset feature	Description
1	Dataset source	The generally utilised data sources of legitimate and phishing websites along with the approaches that grip every source are mentioned in Appendix Table A2. However, the insufficient understanding regarding the methodologies utilised in the collection and preservation of every source results in no concord at all in terms of the quality of various sources.
2	Dataset size	The evaluation dataset size differs between various approaches. As seen, the reliable outcome depends on the size of the dataset; the bigger the better
3	Dataset redundancy	There is not sufficient information in the literature regarding datasets redundancy. Although numerous presentations and overlay between various sources of datasets, particularly of phishing websites, can be seen.
4	Dataset timeliness	Although if a similar source of data and size of the dataset is utilised in two plans, their phishing website information might not be the same. The phishing blacklist supplier generally amends their data plan weekly, daily, or even hourly, because phishing websites last for short-terms.
5	Ratio of legitimate to phishing websites	The ratio of legitimate to phishing example displays the level at which experiments portray an actual world distribution ($\approx$ 100/1) [86].
6	Training set to testing set ratio	The extensibility of the approach is seen in the ratio of training to testing examples [86].

2.3.9 Datasets used for evaluation

Several datasets employed for the evaluation of phishing detection algorithms are available freely on the internet. Some of the most renowned phishing and ham datasets are summarised in Table 2.8.

Table 2.8 Most popular dataset

No.	Dataset	Description
1	Phishing Archive [210]	Phishing Archive is an archive of phishing attacks maintained by the APWG. The attacks recorded in this archive were either reported to or detected by APWG [93]. The evaluations of Dhamji et al. [211] and Abburous et al. [212] make extensive use of this dataset.
2	PhishTank [97]	The phishing data reported by the user is stored in the PhishTank website. This information is accessible via API [213] and is shared via a website.
3	Corpora [202]	There were, initially, two components of corpora of the SpamAssassin project: easy ham, as the name suggests, were easily differentiated from spam, and hard ham which were hard to distinguish from spam [214]. There has been a new addition to this corpus in the form of easy ham_2, a ham dataset, spam_3, and a spam dataset [215]. This dataset has been employed by both Fette *et al*. [147] and Khonji *et al*. [216] to evaluate the algorithm PILFER and implement the LUA algorithm, respectively.
4	Enron dataset [201]	Personal emails are included in the Enron dataset [217], which was generated by 150+ employees involved in project CALO [218]. The dataset had integrity difficulties at first, but Bryan Klimt and Yiming Yang [219] were able to repair them. It is regarded as a benchmark dataset, because it contains about 50,000 spam and 43,000 ham emails [215]. The collection of ham messages involves six Enron workers and the TREC 2005 Spam Track public corpus [215]. Georgala *et al*. use the Enron dataset as well for their research [220].
5	TREC [221]	The TREC corpus [221], utilised by Al-Daeef *et al*. is another extensively used dataset [215]. The copyright of this dataset is held by the Waterloo University. The TREC 2005 corpus, which contains 92,189 emails arranged chronologically, and was generated for spam evaluation [215]. There are 39,399 legitimate emails and 52,790 spam emails in the collection. TREC 2006 and 2007 can also be found on their respective websites [215].
6	IronPorts [222]	IronPorts is a defensive mechanism devised by Scott Banister and Scott Weiss in 2000 against Internet threats. In 2007, the Iron Port's corpus [222] was taken by Cisco and has also been employed by Moore *et al.* [223]. A dataset is a collection of data that appear in their spam traps and emails sent to them by consumers. Iron Port's SpamCop [224], created by Jullian Haight in 1998 and acquired by Iron Port in 2003, is a service that keeps track of spam reports from commercial email or UBE recipients (Unsolicited Bulk Emails) with several spam traps in various areas, making it a significant contributor to the Iron Port corpus [215]. SpamCop also analyses all of the reported spam and compiles a list of the systems that were used to send the emails that SpamCop blacklisted [215].
7	Phishload [225]	Phishload is a phishing database produced by Max-Emanuel Maurer in 2012 [225]. Apart from comprising around a thousand legitimate websites, it also contains HTML code, URL, and other data relevant to phishing websites [215].
8	Nazario/Phishing Corpus [200]	The Nazario/Phishing Corpus consists of 7315 emails that were initially collected from 2004 to 2007 and last updated in 2015. The dataset has been used mainly for phishing email detection.
9	SMS Spam Collection [226]	Is used as the public set of the SMS labelled messages, with 5,574 tagged (ham/spam).
10	The Spambase Data set [215]	The UCI data repository of the Spambase Data set has 57 features and 4,601 instances (2,788 emails labelled as spam and 1,813 ham emails) [215]. Mark Hopkins, Erik Reeber, George Forman and Jaap Suermondt from the Hewlett Packard Labs established the dataset [215].
11	Csmining [227]	This dataset includes the emails from six Enron employees extracted from the Enron corpus. One thousand emails were formed and divided into 20% spam and 80% ham. Selection is made from the Enron dataset as it attains a mix of official and personal emails. It does not include the problems present in the rest of the email datasets.

2.3.10 Optimisations techniques

The term ‘optimisation’ alludes to a method for determining the input parameters or arguments to a function that produces the function’s minimal or maximum output. Continuous function optimisation, in which the input variables to the function are numeric is the most prevalent form of optimisation problem faced in ML. The function’s output is a real-valued assessment of the input parameters as well. These challenges, when separated from functions that take discrete variables and are referred to as combinatorial optimisation issues, may be called continuous function optimisation issues. The population optimisation algorithms are stochastic optimisation algorithms that keep a pool (a population) of potential solutions that is utilised to select, examine, and narrow in on an optimal solutions. This sort of algorithm is designed for more difficult unbiased issues with noisy function assessments and multiple global optima (multimodal), when choosing a suitable or satisfactory adequate approach is difficult or impossible using other approaches.

In phishing email detection, optimisation algorithms have been used in several studies: [12], [15], [156], [27], [30], [37], [40], [114], [116], [144], [155], and the most important of them is the bio-inspired computing (BIC) optimisation technique [228]. The attributes of self-correction and enhancement are inherent in Bioinspired computing (BIC) algorithms, along with a natural tendency to adjust according to the consistently changing environments. BIC is capable of offering flexible, efficient and multifaceted computational algorithms. BIC algorithms have been used in different fields in the past few years to solve issues. BIC solution can be obtained by selecting appropriate dimensions of the problem, assessing the significance of the comparative solutions and describing the operators. Research is being carried out on BIC processes to resolve complicated computational problems [229].

Because each situation is unique, BIC algorithms necessitate various algorithm-dependent parameters [228]. Further BIC may require a high number of iterations to optimise the objective function, which can be inefficient. These algorithms, on the other hand, have two major advantages; the first is an effective information-sharing technique aiding the algorithm's quick convergence, and the other is a lesser probability of becoming trapped in a locally best solution [228]. Other benefits of utilising BIC include the detection of previously undiscovered patterns

and a lower reliance on mathematical modelling or extensive training [230]. Several BIC methods have been employed in the literature to find solutions for phishing email detection [27], [37], [155]. One algorithm known as grey wolf optimisation (GWO), which is based on the natural hunting behaviour of grey wolves. Another optimisation technique is chicken swarm optimisation (CSO), which is based on the behaviour and lifestyle of roosters, hens, and chicks in a chicken swarm. Firefly optimisation algorithm (FOA) is the third method, and it works by measuring the attraction of fireflies by their flashing behaviour. The grasshopper optimisation algorithm (GOA) simulates and mathematically models the behaviour of grasshopper swarms in nature [228]. Whale optimisation algorithm (WOA), which simulates humpback whales' hunt for prey, encircling prey, and bubble-net foraging behaviour [228], is also included in the study.

Figure 2.15 shows how often these methods are used in studies in phishing, suggesting that there is scope for further research in adoption of these optimization methods.

2.3.11 Phishing email detection using NLP

One type of phishing is through spoofing emails, where the phisher emails the user using a fake email address to deceive people so that they end up opening the email [231]–[235]. This allows the phisher to influence the user and gain from their private information [236]. Several anti-phishing technologies have gained traction in countering the problem such as phishing blacklist [237], phishing email detection based on NLP and ML approaches [39], [48], [183], [238]–[241]. The efficiency of the phishing blacklist was investigated by Sheng et al. [237]. The method is based on blacklists of senders and links. Detection involves extracting sender address and link address from the message and cross-checking with the blacklists, for verifying whether the email constitutes a phishing attempt. Blacklists consist of the sender blacklists and link blacklists. The major drawback is the manual revision of the blacklist and users' indication of the website as a phishing website by reporting it. Among all the databases of phishing websites, two major phishing sites are PhishTank [242] and OpenPhish [243]. The efficiency of blacklisting for identifying phishing emails depends greatly on blacklists.

AI has developed a lot, and now phishing email detection has also adopted ML. NLP as well as ML have contributed considerably to combating phishing emails [48]. Earlier, features related to

semantics [238], syntax [239], and context [183] had a major role in phishing detection. A study by [39] developed the simplest ML strategies using RF, DT, LR, and SVM containing supervised classification for phishing email identification. One technique using hybrid feature selection simultaneously analyses content as well as behaviour [240]. ML-based anti-phishing techniques train classification algorithms from both phishing and legitimate emails helped by the ML algorithm to attain classifier model email classification. An investigation by [244] divided features into three categories: basic [244], [245], latent topic model (LTM) [146], [244], [246] and dynamic Markov-chain features [244]–[246]. Basic features can be collected through email and need not be processed further. Topic model features are sets of words connected to each other and can occur together; these are not easily detectable in an email. Text features developed on the basis of Bag-of-Words (BoW) are also known as Dynamic Markov Chain (DMC) features, which involve the modelling of message content to determine the probable association of an email with either legitimate or spam groups. One major flaw with the NLP phishing e-mail detection developed on ML is its high dependence on emails' surface text instead of deep semantics. So, when the structure of a sentence is altered or different synonyms of words are chosen or if different changes are made, it is nearly impossible for NLP built on ML to analyse these changes [247]. This ML method primarily uses feature engineering to create features to accomplish tasks and present emails. Blacklisting and feature engineering both operate manually and a large workforce and expertise are necessary for this task, which restricts the efficiency of detection. Various NLP tasks consisting of text categorisation [248], information extraction [249], and machine translation [250] have been heavily influenced by DL. It can also create features from emails, which will pinpoint phishing attempts automatically, doing away with the need for the manual extraction feature of emails. In phishing email identification, DL helps process emails' text more accurately and efficiently. A study by [251] used DL along with word embedding techniques for reintroducing structure in free text email conversations. This is not used to detect phishing emails but the mode still comprises DL with word embedding to analyse emails. In [20] suggested a phishing detection model prepared on Keras [252], word embedding and also CNN.

NLP technology is currently being employed for detecting phishing emails, rather than using DL techniques, completely disregarding how anti-phishing email and other objectives differ, and partially ignoring contextual information, thus limiting the progress in phishing email detection.

The current literature review reveals that the previously mentioned issues are of considerable importance to grasp the research trend of phishing detection utilising NLP and ML. For the most part, a survey was performed to synthesize the research pertaining to the detection of phishing through Natural Language Processing (NLP) and Machine Learning (ML), with the aim of comprehensively analyzing these research works. Some research has been conducted to identify phishing emails utilising diverse ML approaches. Numerous features have been developed for the categorisation of emails into malicious or safe emails [14], [16], [126], [148], [253], [254]. The researchers in [253] detected phishing emails adopting the best list of features that has high accuracy, but using the least number of features. Their paper suggested a binary search feature selection (BSFS), which assessed with greater accuracy using the least features as well as search time. The findings revealed that the BSFS technique weighed the accuracy of 97.41% in comparison with SFFS (95.63%) and WFS (95.56%). The study still needs more features and sophisticated feature selection methods set to derive the best feature set. An investigation by [126] used document embedding utilising Doc2Vec and performed a parallel arrangement that employed SVM, LR, RF, and NB. The results of the conducted tests indicated a good classification rate with accuracy and F1 score of 81.6% and 76.6%, respectively using the SVM classifier. Despite considerable study on detecting phishing emails, the research is lacking the use of features that permit an easier interpretation and provide a deeper insight into phishing and legitimate emails. The model put forward in [16] has been developed based on a dynamic approach that adopted an inbuilt dataset gathered from various web sources to equip the work with a dynamic dimension and improve accuracy. A hybrid approach has been proposed for phishing detection integrating feature extraction and classification of the mails using SVM. While compared the proposed hybrid method along with SVM (accuracy- 87%, sensitivity-88.5 % & specificity-91%) & Neural Network (accuracy-90.5%, sensitivity-92%, specificity-93.5%) method, it was found that the proposed hybrid method, with an accuracy of 98%, sensitivity of 97%, and specificity of 97.5%, performed better [16]. By increasing the dataset, the predicted method would be strengthened. By using a variety of emails, both phished and ham, the scheme will be closer to the real-world scenario, where fraudsters are constantly improving their methods. A study by [14] used a multi-stage method involving both normal NLP and ML to detect phishing emails; the suggested multi-stage strategy comprises of feature engineering inside NLP, "lemmatisation, feature selection, feature extraction", improved learning strategies for resampling and cross

validation, and the arrangement of hyper parameters. Gualberto et al. [14] introduced two techniques, with the first employing CS statistics and MI to improve dimensionality, whereas the subsequent strategy uses a combination of PCA and LSA. These approaches produce reduced feature sets that, combined with the XGBoost and RF algorithms, lead to an F1-measure of 100% success rate. Validation experiments were conducted using the SpamAssassin Public Corpus and the Nazario Phishing Corpus datasets". An investigation by [254] expects to order spam messages effectively as well as with low latency. The research applied distinctive ML models like "XGBoost, LightGBM, and Bernoulli Naive Bayes" that are extremely quick and are also marked by lower time unpredictability. Another feature taken for this purpose is the length of messages; Unigram, Bigram, and TF-IDF matrix helped pull such features. CS feature selection enabled diminishing space complexity. The main findings of this study indicate that "Bernoulli Naive Bayes followed by LightGBM with the TF-IDF matrix produced the highest accuracy of 96.5% in 0.157 seconds and 95.4% in 1.708 seconds respectively". In order to boost performance, the models would be more stable when the study use ML models that can be trained using datasets from various resources, as well as datasets with a huge number of documents. An email spam detection strategy that uses NLP and ML techniques is shown in [148]. They depict the outcomes obtained from classification and assessment. Validation experiments have been conducted using the SpamAssassin Public Corpus. SVM algorithm used for detecting spam emails. The accuracy of the applied technique is calculated to 0.90, the precision is equal to 0.90, the recall is equal to and finally, the F-measure is equal as 0.90. The weakness of this study is that it is used only one algorithm to train the proposed model (SVM) for detecting spam emails.

THEMIS is a new DL model for detecting phishing e-mails [48]. The model runs on a better convolutional neural network (RCNN) model including multilevel vectors mechanism, which allows concurrent modelling of an email at the header, body, character and word levels. THEMIS has an accuracy of 99.848% according to the outcomes of their study. The only flaw of the model is that it cannot detect phishing in emails with an e-mail body but no email header. A model targeting phishers based on the CharEmbedding was executed by a group of researchers in [241]. In the proposed model, URL is used to extract features of emails that completely outdate the manually created handcrafted features; the model does not depend on network access, which renders it more reliable for clients owing to low response time. It has an accuracy of 95.02%, but

this model still has some downsides. The major drawback is that it does not recognise if the URL of the website is active or has any error; it is really important to examine the URL of the website before any major conclusion. The model sometimes misclassifies phishing sites in case of shorter URLs or URLs containing sensitive words such as “login” or “registered”, which may result in the misclassification of URLs as phishing websites. Additionally, some URLs of misleading websites which aren’t actually replicas of other websites can go un-scanned by the model depending upon the URL string.

2.3.12 Distribution of phishing email detection studies across perspectives

The current systematic review considered 100 research studies published between 2007 and 2022 on the topic of phishing email detection using NLP techniques, to find answers to 11 SLR questions.

RQ1: Which ones are the key research areas in phishing email detection using NLP studies?

The main research area in phishing detection studies is feature extraction/selection with 86 studies. Phishing email detection is another significant research area with 68 studies, while 33 studies investigated the classifications of phishing emailx, as shown in Figure 2.12. Only 14 studies were on improving/optimising algorithm topics, which shows that this research area is not as significant as compared to the other three areas in phishing detection studies, as illustrated in Figure 2.12.

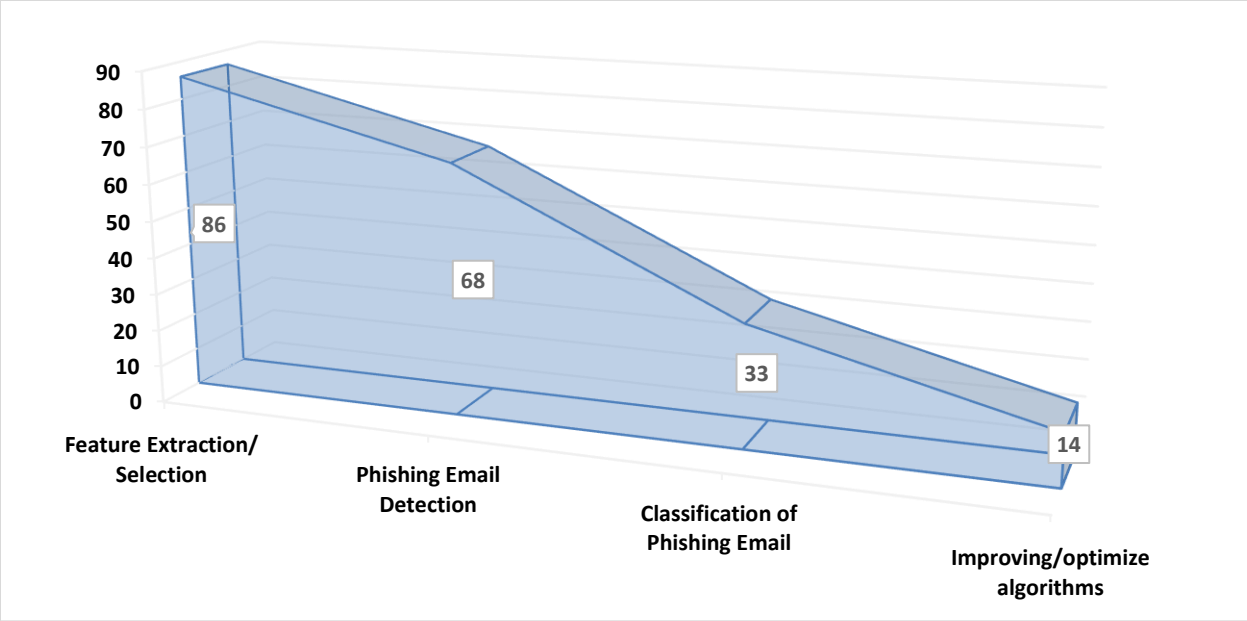


Figure 2.12 Research area distribution

RQ2: What are the ML algorithms used in phishing detection email using NLP studies?

Figure 2.13 shows the distribution of all the analysed articles over the key research techniques in phishing detection studies. The most common research technique used in studies includes SVMs which have been used in 50 studies. The next most common research technique used is NB, used in 33 studies. This is followed by DT and RF with 29 occurrences for each study. Artificial RNN, and NN and CNN are the most applied DL techniques, occurring in 13, 8, and 5 studies, respectively. Figure 2.14 illustrates a summarised view of a 'sample' of 30 techniques drawn from the ML-based techniques discussed in this section. Out of these 30 techniques, nine are supervised, two are unsupervised, and 19 are semi-supervised.

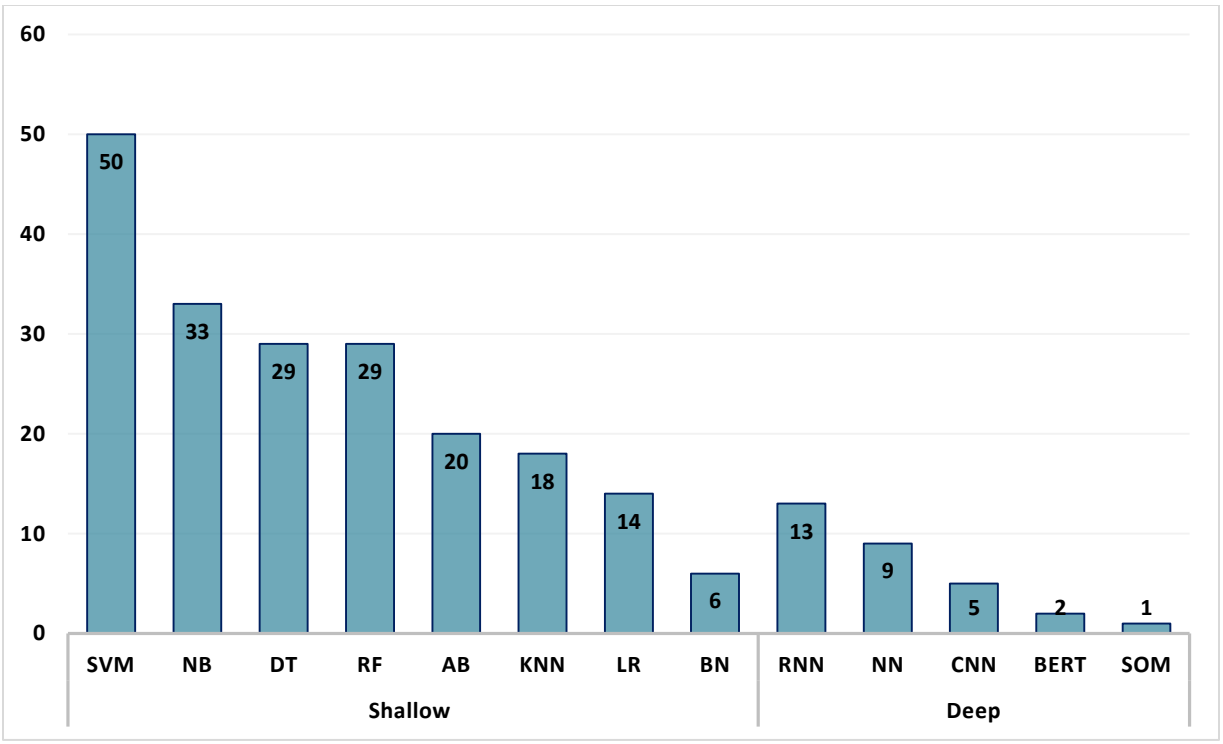


Figure 2.13 The popularity of various ML-based techniques

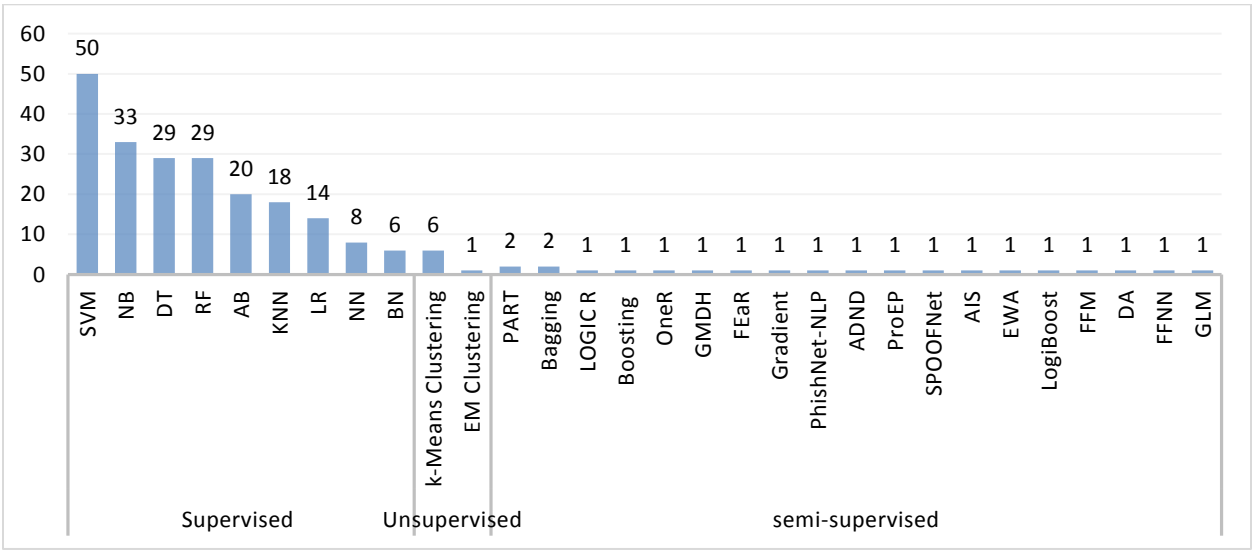


Figure 2.14 The popularity of various techniques

RQ3: What are the optimisations techniques used in phishing detection email using NLP studies?

Figure 2.15 illustrates the most popular optimisation techniques used in phishing email detection studies. The most frequently used technique is the Adam optimiser, constituting more than 26% of the optimisations techniques in the reviewed literature. Second in popularity is the sequential minimal optimisation (SMO), with 21% instances of use in the papers reviewed. SMO reveals the significance of a word to a document in the textual datasets.

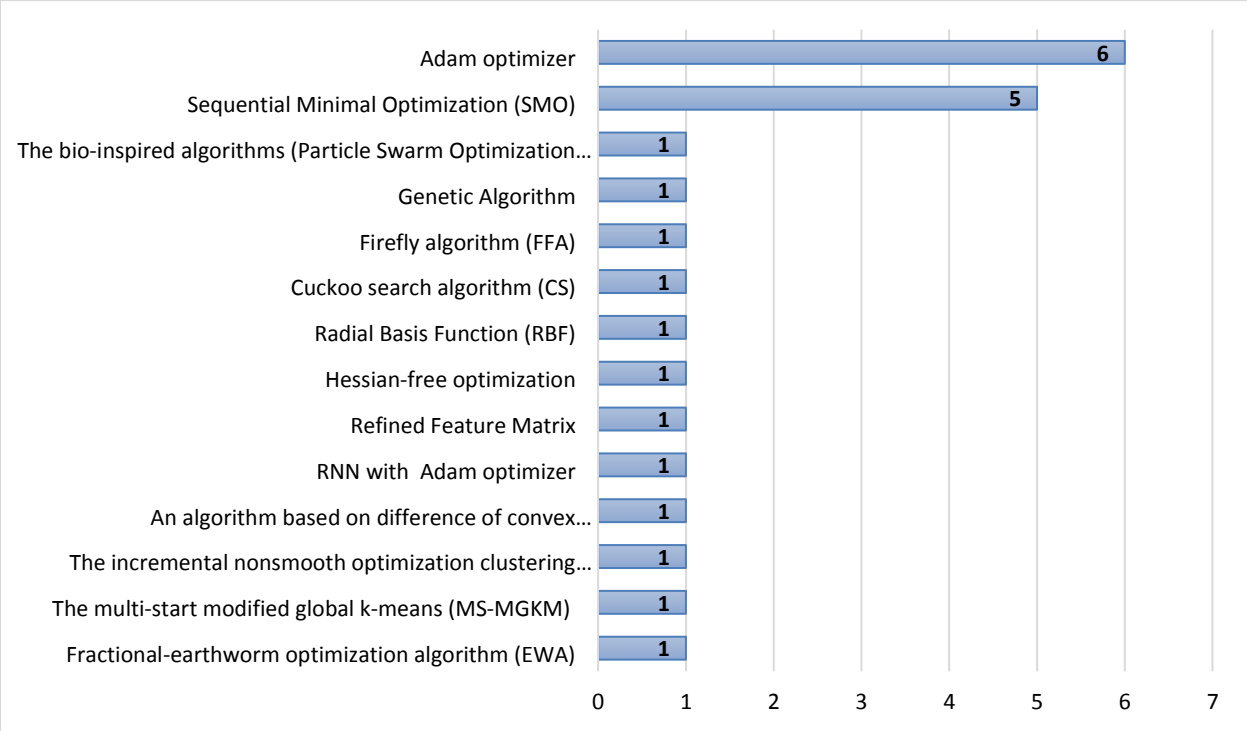


Figure 2.15 The popularity of various optimisations techniques

RQ4: What are the text features in phishing email detection using NLP studies?

Figure 2.16 shows the text features used in phishing email detection studies include the BoW and information gained (IG) that have been used in 10 studies each and Word2vec has been used in nine studies. Other less common but significant features used in studies were PCA, part-of-speech tagging (POS), doc2vec, latent dirichlet allocation (LDA), LSA, and CS, identified in six, four, three, three, three, three studies, respectively.

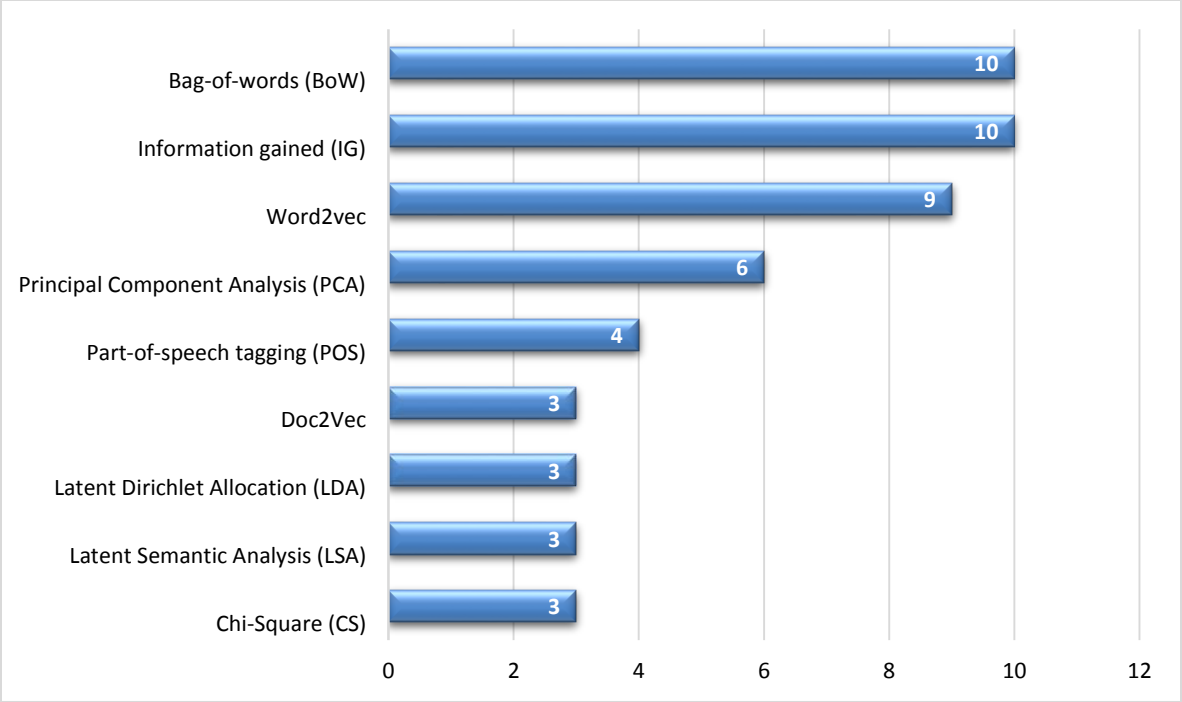


Figure 2.16 The popularity of various techniques

RQ5: What are the NLP techniques used in phishing email detection studies?

In terms of NLP techniques, Figure 2.17 depicts the most popular NLP techniques used in phishing email detection studies. The most frequently used technique is Basic NLP tasks at 59 studies; the basic NLP tasks include '*stopword removal, punctuations, special characters, stemming, and*

tokenisation'. Second in popularity is TF-IDF at 36 studies. TF-IDF reveals the significance of a keyword to a document in the textual corpus. Finally, word embedding was found in 12 studies. See Appendix A1 for details.

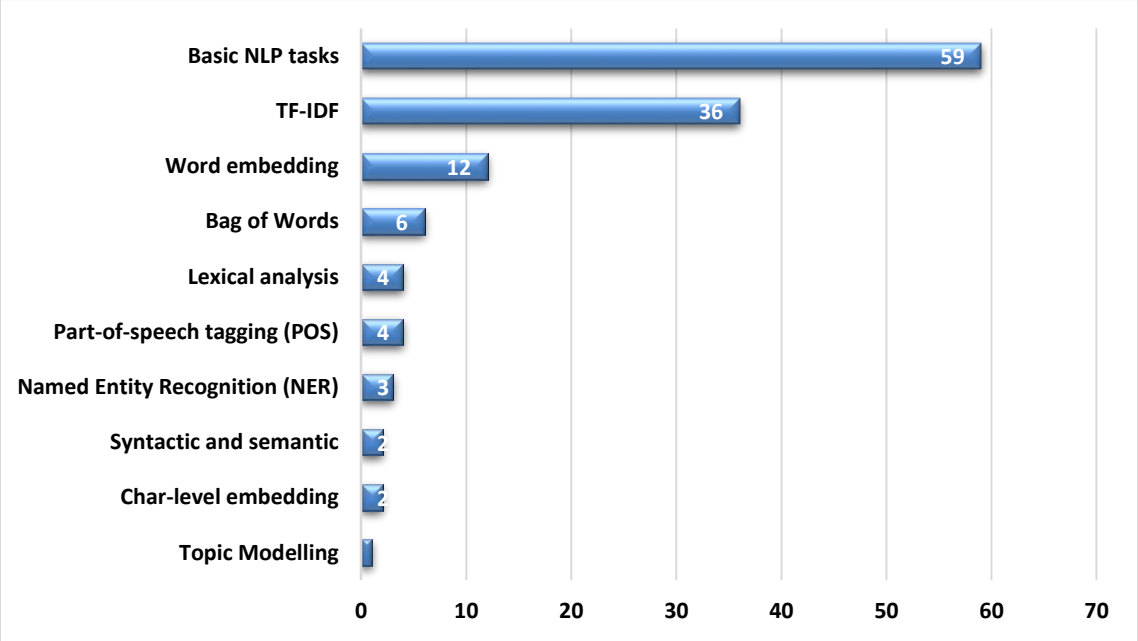


Figure 2.17 The popularity of various NLP techniques

RQ6: Which datasets and resources have been used?

Figure 2.18 shows the distribution of the analysed articles over the resource type. The Nazario phishing corpus has been in the majority of studies at 42 studies. The next common resource is the SpamAssassin Public Corpus, identified in 40 studies. The Enron dataset has also been discussed in 23 studies. This is followed by PhishTank, IWSPA-AP, and TREC corpus at seven, six, and six studies, respectively.

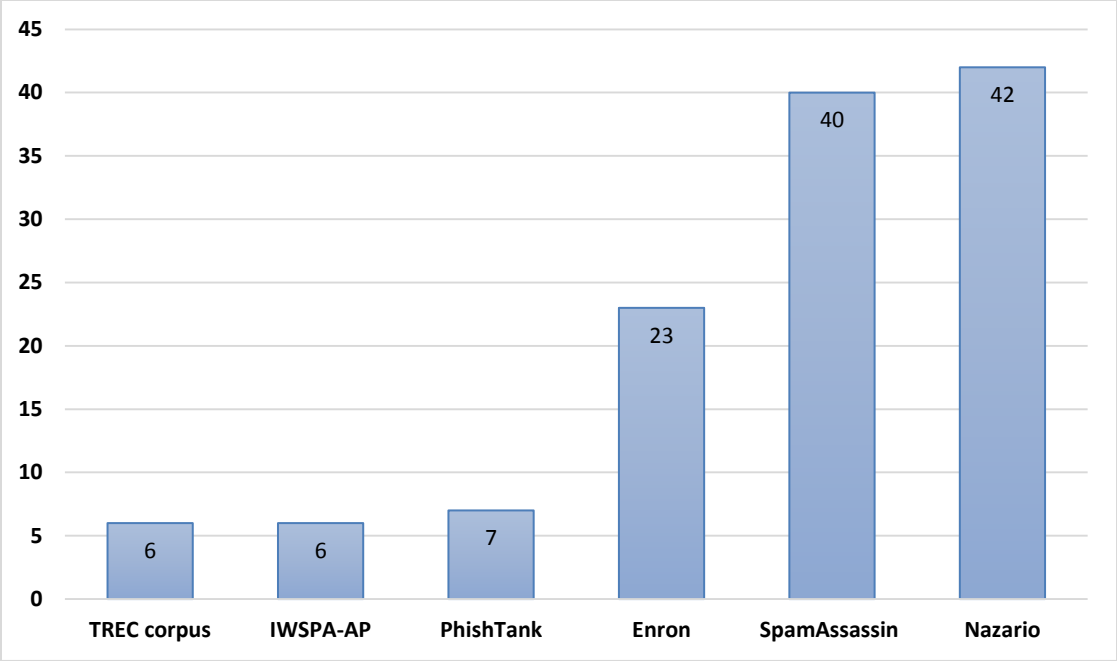


Figure 2.18 Resources type among selected papers

.

RQ7: What are the evaluation criteria of the ML/DL techniques that were used in phishing email detection using NLP studies?

According to [86] accuracy, precision, recall, and F1-score are the four ML/DL methods that serve as metrics for the evaluation of the quality of outcome obtained from various phishing detection techniques [86]. The popularity of applying various ML/DL methods as evaluation metrics for phishing email detection has been shown in Appendix Table A1. Multiple metrics were also used in some experiments for evaluation of the quality of the phishing email detection employed. Eighty-three experiments were performed, where researchers opted for employing the accuracy standard as a metric, while 38 experiments employed the F1-measure. The metric holding 3rd place in popularity was the confusion matrix (TPR, FPR, TNR, FNR, specificity, and sensitivity, which was used in 36 experiments while the metric of recall and precision standards, used in 35 experiments each, held fourth place, as depicted in Figure 2.19. It has been found that the measurement of examined classifiers in terms of their quality is done based on accuracy, recall,

F1-measure, precision, sensitivity, and specificity. Their computation is shown subsequently. Moreover, any other information regarding the clarification of these measures using a confusion matrix can be viewed in Appendix Table A1.

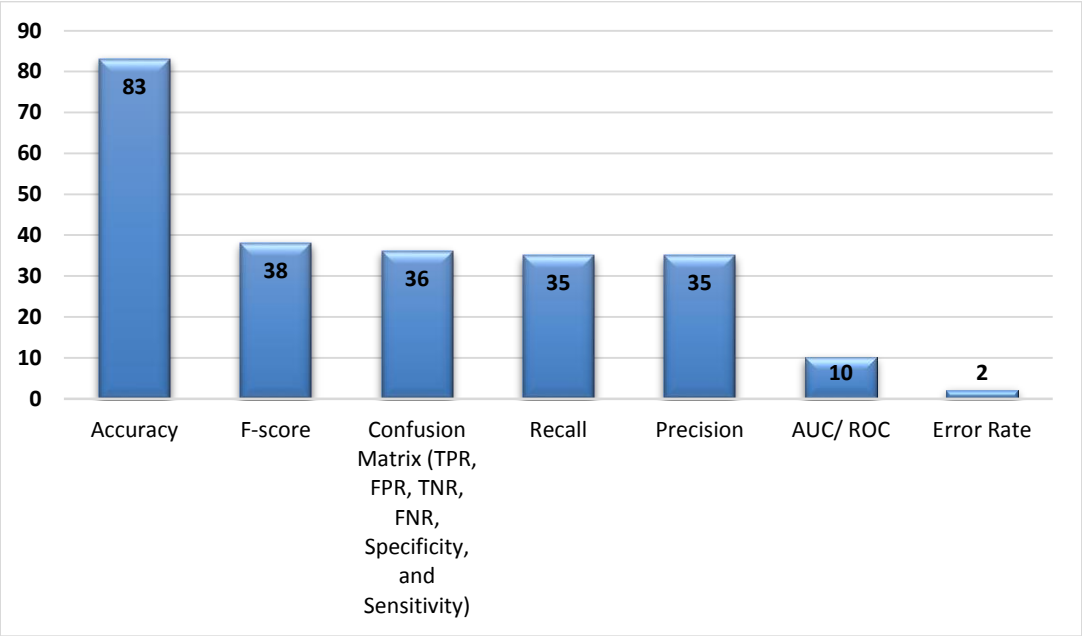


Figure 2.19 Popularity of Various Evaluation Criteria in Researches

RQ8: What are the tools used in phishing detection email using NLP studies?

In this section, popular tools used for phishing email detection are described. Figure 2.20 describes tools/applications used in the reviewed articles. General tools were collected by the authors of each study. For instance, The Python programming language [255], Weka [256], Keras library for the training of deep-learning models, Scikit-learn library, and Java programming language were utilized. It can be clearly seen that most of the surveyed studies were conducted using Python (N = 37), followed by Weka (N = 17), then Scikit-learn library (N=12), Keras (N = 9), Java (N=8) and TensorFlow (N = 6).

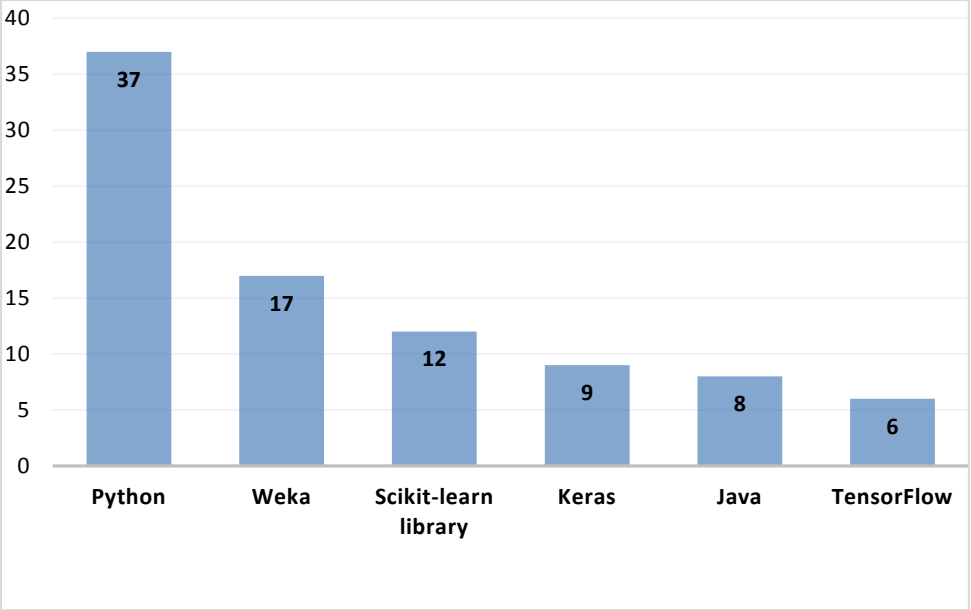


Figure 2.20 Popularity of various tools in researches

RQ9: Which parts of the email are the most widely used in phishing detection email using NLP studies?

Figure 2.21 shows that 38% of the phishing detection email studies mainly relied on email body text (N = 93) for data collection, followed by both email header and URL (N = 75, N = 66, respectively).

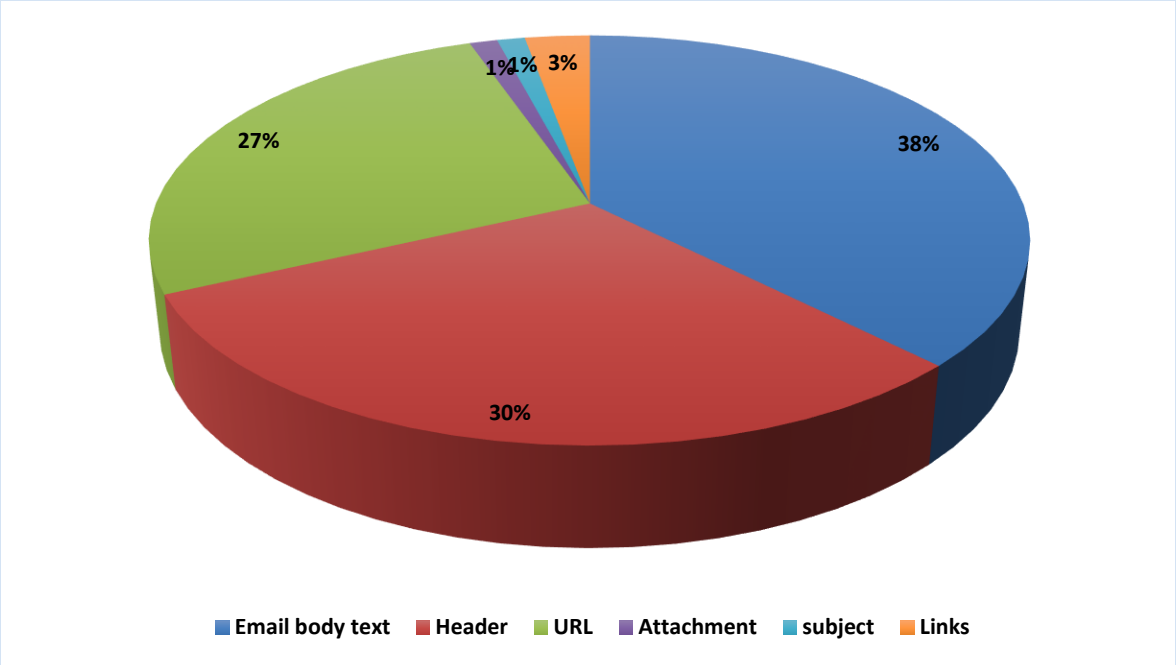


Figure 2.21 Distribution of phishing email detection studies per part of the email

RQ10: What are the trends across time in phishing email detection using NLP?

Figure 2.22 shows the distribution of phishing email detection studies in terms of publication year, indicating that studies have increased over the years. As can be observed, in the studies from 2006 to 2022, the highest number of publications rapidly grew from one publication in 2007 to an average of 12 studies in the last four years. It can also be noticed that the number of articles increased from six studies in 2012 and 2013. Moreover, there is a drop-down ratio of five publications in 2014 and 2015, and this has decreased to 4 in 2016 and 2017. There have been a total of 57 studies almost 57% of these 100 studies published during the period from 2018 to 2022. The highest number of studies was published in 2020, with 22 publications. The next highest publication year was 2018 and 2019, during which a total of 11 studies were published in phishing email detection using NLP techniques.

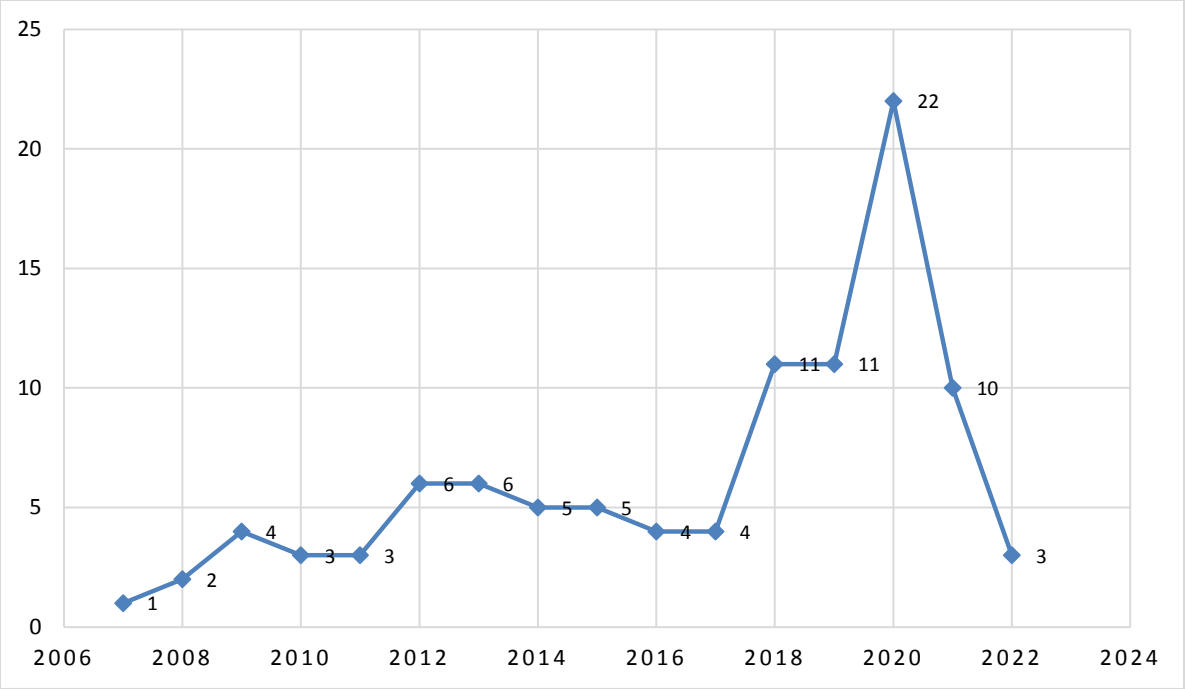


Figure 2.22 Distribution of phishing email detection studies per publication year

RQ11: What proportion of phishing email detection studies are on mixed language models?

Considering the outcomes of Figure 2.22 that show the distribution of phishing email detection studies in terms of publication year, the proportion of studies on mixed language models is 2%. Mostly, there are studies on English datasets for phishing email detection. There are two papers on Arabic phishing email detection using classical ML on mixed language models [65], [142]. Due to a lack of resources for Arabic spam/phishing emails, and the limited amount of progress achieved in tackling Arabic NLP in general, studies on Arabic phishing email detection are insufficient.

2.4 Discussion

There are only a few studies [65]–[72] on Arabic spam/phishing email detection. Because there are not enough resources to deal with Arabic spam/phishing emails, most of the Arabic research to date has aimed to educate end-users on how to detect phishing emails. Some experts suggest that educating end-users to detect phishing emails and websites is a good approach [257]. Some firms, such as Symantec and Microsoft, provide their users with educational resources on phishing. Traditional text-based resources or instructions, on the other hand, are ineffective for learning how to detect phishing emails or websites. Users require more appealing, dynamic, and enjoyable methods for education. Furthermore, they must put their newly acquired knowledge to trial in a secure environment. Certain researchers created educational games to offer a quick fix [67]. Regrettably, most of these educational materials and games were created in English. This makes it difficult for non-English speakers to take advantage of these resources, particularly in the Arabic and Middle East regions, where security awareness is lacking.

Al-Mohannadi et al. [66] performed a study on the risk of email phishing in Qatar and related it to the risk of email phishing in the United Kingdom. She found that the threats posed by phishing are growing and becoming increasingly widespread in Qatar. One of the causes for the effectiveness of phishing email attacks in Qatar, according to the research, is specific ignorance, or perhaps the inclination for Qataris to be misled online. This is due to the absence of understanding of email phishing threats and how to prevent them. This is still a reason for worry because hackers can readily access personal information by utilising phishing email techniques.

Baiomy and Youssif [67] dealt with the issue of inadequate security knowledge in the context of phishing attacks in Egypt and the Middle East. To educate Arabic users regarding phishing URLs, they created an anti-phishing game. They designed and executed their game using a well-known framework. They then put the researchers' execution to trial to see how useful it was as a training technique. They began by determining phishing site URL attributes that aid in the detection of phishing sites. Then, to create their anti-phishing game, they used a well-known game design framework (EDPE). They used a pre-test and post-test analysis to identify the degree of phishing awareness among 56 individuals before and after playing the game. They employed a paired t-test and a one-way analysis of variance (ANOVA) statistical study to see how much an anti-phishing

game could assist users to recognise and prevent phishing attacks. The findings of the pre-test confirmed that security awareness in the Arabic region is still in its infancy, whereas the post-test results show that meaningful educational games in the Arabic language can be utilised to educate Arabic users regarding security principles and increase their understanding of security.

Another study [235] introduced a phishing detection framework in Arabic aimed at user education and gamification to assess learned knowledge. Similarly, Ahmad and Erdodi [69] discussed the analysis of phishing attacks involving Arabic domains, focusing on homograph attack vectors using internationalised domain names (IDN) of the Arabic character set.

Another study [70] aimed to see how well Saudi Arabian users could detect phishing emails. The findings of this study confirm the hypothesis that numerous personal aspects influence users' recognition stages, as indicated in the deception-detecting framework. The first task was to translate the survey from English to Arabic. Because some students had not yet activated their university email, the following phase encouraged them to do so. The initial survey was sent to gather information on the students' particular characteristics, and then normal emails were sent to them before the trial phishing email was sent. The next stage focused on delivering the test phishing email and a concluding survey to gather information about the individuals' behaviours. Approximately 200 Saudi Arabian students were randomly separated into two categories: those who clicked on emails and those who replied to emails in this study. The phishing email was replied to by 14 students (7% of the population). The 3–11% victim percentage of the population attacked by phishing emails is consistent with this result. The *click email* generated the majority of the responses (86% of the victims). This suggests that individuals were hesitant to share sensitive information over email or on explicit demand. People are not asked to provide personal information explicitly in *click emails*; instead, they are directed to a login webpage for information. This may lead people to overlook the importance of verifying the email's legitimacy.

A couple of studies [71], [72] conducted two phishing tests in different languages (English and Arabic). For two causes, they discovered that the number of respondents who were unable to detect the phishing email in the first test significantly decreased in the second experiment. The first explanation was that the users' phishing awareness was enhanced as a result of the phishing

information given to the victims when they clicked on the link. The email's language was the second factor. The second experiment was conducted in English, even though most of the respondents were non-English speakers. It led to the discovery that the email's language has a considerable impact on the outcomes of tests.

Spam/phishing emails in Arabic can be identified using ML algorithms. This is accomplished by improving the algorithms using various kinds of training datasets. Computer scientists can use datasets to develop new ML approaches and techniques. To train the algorithms to identify such emails, datasets are needed.

El-Halees [65] described an approach for filtering spam from mixed Arabic and English corpus email messages. ME, DT, ANN, NB, SSM, and KNN were among the supervised ML algorithms examined by the system. The system's efficiency was adequate while using personal email messages in English alone, with the best approach being SVM with an accuracy rate of 99.03% and F1-measure of 98.32%. Only in Arabic, where the maximum accuracy was 89.77% by ANN and F1-measure 76.25% by SVM, the performance was not precise. These outcomes can be attributed to the fact that Arabic is a strongly inflected language. As a result, they stemmed Arabic messages before classification, and ME, with an accuracy of 92.92%, and NB, with an F1-measure of 92.42%, performed better. While the study presented promising results in detecting spam emails from mixed Arabic and English corpus, there are several limitations to this work. Firstly, the study utilised a relatively small database consisting of about 1047 email messages containing 41,883 tokens. This small dataset size may affect the generalisability of the study's findings and limit the applicability of the proposed approach to larger datasets. Secondly, the study relied solely on the MI method for feature extraction. While MI is a widely used method for feature extraction, its use in isolation may result in a limited set of features, which can adversely affect the performance of the ML classifiers. Thirdly, the study only used traditional ML classifiers for classification and did not incorporate any DL classifiers. The absence of DL classifiers may limit the study's ability to identify more complex patterns in the dataset, which could have improved the accuracy of the classification. Furthermore, it would have been useful to compare the performance of the proposed approach with other feature extraction methods and ML classifiers to validate the effectiveness of

the proposed approach. This would have provided a more comprehensive analysis of the proposed approach and its effectiveness in comparison to other approaches.

The researchers discovered that the majority of previous research focused on educating end-users on how to detect phishing emails in Arabic. ML models were used by El-Halees [65] to eliminate spam from email messages in a mixed Arabic and English corpus. As previously stated, none of the following methods were evaluated in a parallel Arabic–English phish email corpus. A parallel corpus is a group of texts that have been transcribed into one or more languages other than the original. Where only two languages are present, the basic example is when one of the corpora is an exact translation of the other. Yet, some parallel corpora persist in multiple languages. Furthermore, the translation orientation does not have to be consistent, so some texts in a parallel corpus may have been translated from language A to language B while others were converted the other way around. The translation's orientation may be unknown. Parallel corpora are currently attracting attention due to the possibility of aligning original and translation content and gaining information about the nature of translation. They are important in translation and contrastive research, and with the emergence of the data-driven learning (DDL) approach, they are now becoming prominent in translation training and language education. Despite their importance, Arabic appears to have a suitable parallel corpus resource for phishing email detection. Minimal Arabic–English parallel corpora have been reported in the literature, and those that have been described are frequently erroneous and/or expensive. Of the available ones, some are tiny [65], while some are genre-restricted [66]–[72], which makes them unsuitable for the needs of academics. Based on a translated imbalanced phishing data set (the First Security and Privacy Analytics Anti-Phishing Shared Task) (IWSPA-AP), this research describes a framework for detecting phishing text in Mixed Language Arabic–English. The IWSPA-AP training dataset contains 5721 phishing and legitimate emails (Arabic–English parallel corpus) that can be utilised for ML to detect phishing emails in mixed languages. The bidirectional corpus can be employed to compare and contrast translation and the source language. At several phases, involving translation, text segmentation, alignment, and file production, the corpus was manually checked.

The SLR showed that phishing email detection has been extensively researched in common languages such as English. However, these findings cannot be generalised to non-English-speaking

nations, such as the Arab world, due to significant cultural and linguistic differences. With Arabic being a Semitic language with rich morphology, there are few papers on Arabic spam/phishing email detection using classical ML techniques [65], and no work has been done on content-based phishing email detection for the Arabic language. This research gap highlights the need for a more rigorous study on semantic analysis of email body text to understand the intention of the sender and detect phishing attacks in Arabic language emails. Therefore, there is a critical need to develop detection systems that can identify phishing emails in the Arabic language by removing language barriers and enhancing semantic analysis.

While email phishing detection has been significantly researched and developed for many of the world's major languages, notably English, there remains a glaring shortfall when it comes to emails in the Arabic language. Several challenges contribute to this problem. The richness of the Arabic language in terms of its morphology and the lack of resources dedicated to Arabic spam/phishing email detection exacerbate the issue. Moreover, many existing detection systems and educational resources are developed in English, making them less accessible to non-English speakers in the Arabic and Middle Eastern regions, where security awareness is relatively low. There is a critical need for the development of phishing detection systems and educational games that can effectively cater to Arabic language users, and this should involve a more robust semantic analysis of the email body text to understand the sender's intent accurately. It is also essential to address the specific cultural and linguistic differences that impact the generalisability of phishing detection strategies across different languages.

2.5 Knowledge gap

The present state of research in phishing email detection reveals a significant knowledge gap concerning the Arabic language. The uniqueness of Arabic, characterised by its rich vocabulary, varying dialects, distinctive writing system, synonyms, homographs, and diacritics, makes it a challenging task for conventional filtering systems, which are predominantly oriented towards the English language, to effectively detect and filter phishing attempts in Arabic emails. The problems are compounded by the scarcity of extensive, high-quality, and diverse Arabic-language datasets that could be used to train ML models for robust phishing detection. The limited availability of such resources poses a severe constraint to the development of advanced algorithms capable of

accurately classifying and detecting phishing emails in Arabic. This leaves users exposed to potential cyber threats and heightens the need for comprehensive, Arabic-specific phishing detection solutions.

Moreover, there is a significant lack of research focusing on leveraging advanced NLP techniques and DL models specifically for Arabic phishing detection. The development of models tailored to the characteristics of the Arabic language has yet to be adequately explored. Such models could potentially offer a more effective approach to detecting phishing attempts in Arabic by understanding the morphological and syntactical intricacies of the language.

Additionally, while certain studies have proposed using character-level analysis for improved phishing detection, there is still a lack of exploration regarding their effectiveness when applied to the Arabic language and its unique script characteristics. Thus, the field of Arabic phishing detection calls for urgent, focused research attention, to pave the way for the development of sophisticated, language-specific tools and systems that can provide effective protection against phishing threats.

2.6 Summary

This chapter describes the intent to perform a meticulous SLR, dedicated to amalgamating and synthesising the collective body of research pertinent to the methodologies involved in phishing email detection. These methodologies predominantly utilise ML and NLP techniques. With the application of pre-established selection criteria, a total of 100 scholarly articles published in the timeframe of 2006 to 2022, were examined. The focal points of this exploration include key research domains in phishing email detection that make use of NLP, the ML algorithms and optimisation techniques employed in phishing email detection, text features unique to phishing emails, the datasets and resources utilised in this field, and the evaluation criteria established. This chapter also endeavours to shed light on the varied perspectives involved in phishing email detection studies. The primary findings of this chapter have been consolidated and deliberated in the discussion section. In summary, the findings of the systematic review highlight significant knowledge gaps that currently obstruct progress in the field of Arabic phishing email detection. Foremost, there exists a pressing demand for expansive and comprehensive datasets specific to Arabic, to augment the training and refinement of ML models, given that the bulk of existing datasets are designed around English. Additionally, the intricacy and linguistic richness of the Arabic language present distinctive challenges in accurately decoding the semantic meaning of phishing emails, indicating an area necessitating deeper investigation. Addressing these knowledge gaps is paramount for enhancing cybersecurity protocols for Arabic-speaking users and for the development of a robust phishing detection framework.

Chapter Three

English–Arabic Parallel Corpus Generation for Email Phishing

3.1 Overview

As delineated in Chapter 2, Section 2.5 ("Knowledge gap"), the task of filtering phishing emails in Arabic encounters numerous complications, primarily arising from the unique intricacies of the Arabic language and the dearth of appropriate resources for ML model training. Arabic, characterised by its abundant vocabulary, diverse dialects, and distinct writing system, stands as a complex language. This complexity often baffles conventional filtering systems designed with an English-centric focus, hampering their ability to efficiently detect Arabic phishing attempts. Adding layers to this intricacy are the Arabic language's properties such as synonyms, homographs, and diacritics, offering phishing content ample opportunities to slip past detection mechanisms. The paucity of comprehensive and diverse Arabic-language datasets for ML model training is another considerable hurdle. This lack of resources restricts the creation of sophisticated algorithms capable of robustly classifying and identifying Arabic phishing emails, leaving users exposed to potential cyber threats. To tackle these impediments and enhance the efficiency of Arabic phishing email detection, substantial efforts must be directed towards creating all-encompassing datasets and crafting ML methodologies tailored specifically to the Arabic language's complexities. In response to the limited resources available for training and testing ML models that can effectively classify Arabic phishing emails, an inventive approach was embarked upon. Recognising the imperative need for a comprehensive dataset, 1,258 emails from the esteemed dataset were manually translated, "The Leading Security and Privacy Analytics Anti-Phishing Shared Task (IWSPA-AP 2018)." This rigorous translation process led to the creation of a substantial corpus of simulated phishing emails in Arabic, considerably expanding the data resources for Arabic-language phishing emails. In addition, the existing data was supplemented with a collection of 300 Arabic emails designed for testing purposes. This new dataset aims to support the development of more precise and effective phishing detection systems for Arabic. This approach is geared towards addressing the data availability gap and surmounting the challenges tied to Arabic phishing attack detection. The resulting dataset, both diverse and comprehensive,

provides a plethora of valuable insights, permitting ML models to train with an emphasis on Arabic content. Consequently, the generated emails accurately capture the complexities and nuances of the Arabic language while encapsulating various phishing tactics and strategies frequently used by attackers targeting Arabic-speaking users. The chapter begins by first highlighting the distinguishing features of the Arabic language that make it different from English. Afterwards, the chapter sheds light on the parallel Arabic/English corpora. The other sections of the chapter describe the process of development of a new corpus and its examination.

3.2 Arabic: a global language

Arabic is the primary language spoken by more than 400 million individuals worldwide [258], [259]. Over the years, much advancement has been made in the field of Arabic-related computing research and applications. Specifically, a large number of individuals use the Arabic language for Internet access [260]. Several users speak only the Arabic language, which is why they are unable to comprehend vast amounts of English data at present [261]. Furthermore, there has been a worldwide expansion of interest within Arabic nations in terms of economics, politics, culture, and other aspects. Since the Arabic speakers are in large numbers and English holds significance, the language translation must be carried out using high-quality parallel corpora. Yet, MT is faced with challenges due to the structural differences amongst the languages. As compared to the European languages, Arabic needs separate treatment since it has an exclusive morphology, and as indicated in Table 3.1, in terms of graphology features, English and Arabic are quite different from each other, not only in terms of syntax and grammar but also in terms of cultural context and writing styles [262]. This can have significant implications for detecting phishing attacks in each language. Phishing attacks are often designed to exploit vulnerabilities in human behaviour, such as fear, urgency, and curiosity. The Arabic language is characterised by a rich and complex cultural context that shapes how people perceive and respond to different types of messages. For example, a message that appeals to religious beliefs may be more effective in Arabic than in English, where religion plays a less central role in daily life.

Furthermore, the Arabic language has a different writing style that can make it harder to detect phishing attacks. For instance, Arabic is written from right to left, which can make it difficult to spot fake URLs or malicious links in emails or other online messages. Moreover, many Arabic

speakers may not be familiar with the common phishing tactics that are prevalent in the English language, such as the impersonation of trusted organisations or the use of urgency and scare tactics. This lack of awareness can make Arabic speakers more vulnerable to phishing attacks, especially those that are tailored to their cultural context.

In conclusion, Arabic is very different from English when it comes to detecting phishing attacks. Therefore, it is essential to take into account these differences when designing anti-phishing measures and educating Arabic speakers about the risks of phishing attacks.

Table 3.1 Difference between English and Arabic

	English	**Arabic**
Connection	Usually, diagonal strokes link each character to the next.	In Arabic letters, the baseline is connected with horizontal strokes.
Character versions	Characters have limited shape variations in English.	According to their relative position in the word, Arabic letters might have up to four distinct shapes.
Capitalisation	Yes	No
Direction	Follows the left-to-right direction in reading/writing.	Follows the right-to-left direction in reading/writing.
Features	English-writing has a specific geometrical feature.	The letters or segmented sub-letters are different from the segments in English.
Gender differentiation	No differentiation.	Verb and sentence structure.
Language codes	en | eng	ar | arb
Plural forms	Singular and plural.	Singular, dual and plural.
Position of adjective	Before the noun.	After the noun.
Place with most speakers	The United States of America.	Egypt

Segmentation	Handwriting can be segmented into different letters or sub-letters using any analytical segmentation method.	The letters or segmented sub-letters vary from those in English.
Size of alphabet	26 letters	28 letters
Types of sentences	Verbal	Nominal and verbal
Total speakers	1.348 billion	274 million

3.3 Non-phishing English–Arabic parallel corpora

Recently, Arabic researchers have been focusing extensively on parallel corpora. However, there are not many Arabic–English parallel corpora available in the literature. Research indicates ([263], p. 327) that the shortage of these corpora may be attributed to the limited availability of financial and material resources and the prevailing uncertainty of the concerned authorities about the effectiveness and significance of corpora. Among the most prominent English–Arabic corpora projects is "the English–Arabic Parallel Corpus of the United Nations Texts (EAPCOUNT)", based on 341 paragraph-aligned texts [264]. A compilation of a couple of sub-corpora, it includes 5,392,491 words. One subset includes English content in its original form, while the other includes the corresponding Arabic translations. The corpus was developed by compiling textual content from UN resolutions and annual reports, as well as texts extracted from the literature issued by international institutions [264]. Likewise, another such project was sponsored by the European Commission, in which the experts at the Language Technology Lab in Germany developed a multilingual parallel corpus, MultiUN [265]. This 300-million-word long parallel corpus was actually a compilation of chunks of data obtained from the UN documents issued at the UN's official website during the period 2000–2009 (see [265]). Another parallel corpus, namely the Open Parallel Corpus (OPUS), was developed by Tiedemann (2012) [266], who offered it to be used as a free multilingual parallel corpus; he obtained online translated texts and compiled them into the corpus. OPUS allowed parallel, as well as monolingual, data to be processed through its open-source tools; it also facilitated the research process by offering several search interfaces. OPUS was developed automatically, that is, without involving any manual processing, as mentioned on the website. The European Union sponsored the development of a multilingual

parallel corpus, namely the EuroMatrix, based on texts taken from the European Parliament proceedings; the extracted texts were originally in English and translated into Arabic and other languages. This EU-developed corpus contained 1.5 million Arabic words out of a total of 51 million words. This corpus intended to facilitate and support machine translation systems. Considering the corpus development in Arab countries, experts at the Kuwait University obtained extracts of Arabic translations from the book series "World of Knowledge" and compiled them to formulate a parallel corpus; this book series was issued in Kuwait by the National Council for Culture, Arts and Letters (NCCAL). There were a total of 3 million words in this corpus; the corpus could only be accessed and used by staff and students associated with Kuwait University, specifically those enrolled in the lexicography and translation programs [263]. Several projects of parallel corpora (including the Arabic language projects) had been initiated by the Linguistic Data Consortium (LDC). GALE Phase 2 Arabic Broadcast News Parallel Text is among their prominent projects containing data obtained and recorded under the LDC's supervision. The data contained extracts of news aired from 2005 to 2007 in the form of Arabic source texts with English translations. There were 60 source–translation document pairs containing 42,089 Arabic source text words with corresponding English translations in the corpus (See [267]). Moreover, the LDC automatically incorporated texts from a couple of monolingual corpora, including the Arabic Gigaword Second Edition (LDC2006T02) and English Gigaword Second Edition (LDC2005T12), to come up with Arabic-English Automatically Extracted Parallel Text. This corpus was based on new articles in Chinese and French issued by the Xinhua News Agency (Chinese) and Agence France-Presse (French). There were 1,124,609 sentence pairs in the corpus with about 31 million English words (See [267]). Another multilingual corpus was developed at UMIST by [268]. The corpus contained texts pertaining to IT in the English language with Arabic and Swedish translational corpora. There were 1 million tokens of Arabic text and 2.7 million tokens of Swedish text. The IT content was extracted from multilingual IT websites and included guides and manuals meant to instruct the users of computer systems, hardware, and software. The corpus can only be used by researchers after obtaining prior copyright approval, as it cannot be freely accessed by the public. The Qatar Computing Research Institute also created the corpus of AMARA [269], [270], which extracted data from educational platforms such as Technology, Entertainment, and Design (TED) and the Khan Academy in the form of video captions developed by the community. The corpora contained both Arabic (2.6 million) words and English (3.9 million) words. This corpus

was designed for the machine translation of data. The corpus was equipped with an editor which allowed the generation of subtitles (see [271]). In his work, Izwaini [272] collected a substantial amount of data comprising 27.8 million Arabic words and 30.8 million English words to create a parallel corpus. The data was extracted from reliable sources such as the Al-Hayat newspaper and the OPUS corpus. The corpus was anticipated to facilitate researchers exploring machine translation. Similarly, Hassan and Atwell (2016) [273] compiled about 2 million holy words of the Prophet Mohammad (Peace Be Upon Him) to develop a Hadith corpus in Arabic with translations in multiple languages of English, French, and Russian.

As established earlier, a wealth of critical corpora has been developed to enable the translation process between Arabic and English in various fields. However, an apparent deficit of Arabic-language representation emerges when examining the existing literature on phishing email detection. This chapter is dedicated to addressing this data availability void and navigating the intricacies associated with phishing detection in the Arabic language. Through these efforts, the aim is to lay a solid foundation for the development of sophisticated Machine Learning (ML) models, specifically tailored for detecting Arabic phishing emails. The product of this endeavour will be a diverse and comprehensive dataset – a significant tool for bridging this existing knowledge gap. The dataset, rich with critical insights, will allow for the focused training of ML models on Arabic content. This comprehensive approach not only fosters a nuanced understanding of the Arabic language but also propels the development of precision-targeted solutions to combat phishing attempts. By undertaking this progressive initiative, it is anticipated that a potential paradigm shift in the realm of Arabic-language phishing email detection will occur, contributing significantly to the safety and security of Arabic-speaking digital communities.

3.4 Building Arabic–English phishing email corpus

Detecting phishing emails in Arabic presents a formidable task, largely due to the scarce availability of data in this language. To surmount this obstacle, the development of two distinct data sources designed to bolster the capacity to identify Arabic phishing emails is proposed. The first data source will primarily serve as a training ground for ML models. Leveraging this source will enable the construction of robust models that can proficiently detect Arabic phishing emails, delivering high accuracy levels. Simultaneously, the second data source will function as a testing

platform to assess the trained ML models' performance. Through this iterative process, the models can be refined to optimise their precision continually. Utilising these two data sources concurrently promises to significantly augment the capability to intercept phishing emails in Arabic. This enhancement is pivotal in safeguarding the organisation and its users from such malicious cyber-attacks. Therefore, this dual-source strategy stands as a beacon of the organisation's commitment to fortifying digital security within the realm of the Arabic language.

3.4.1 English–Arabic parallel corpus for email phishing

To tackle the scarcity of resources essential for training ML models adept at classifying Arabic phishing emails, a renowned dataset was strategically selected via a systematic review in Chapter 2. This dataset, termed the "Leading Anti-phishing and Security Analytics Joint Task Force (IWSPA-AP 2018)," was utilised to produce an expansive quantity of simulated Arabic phishing emails, effectively enriching the available data pool for Arabic phishing emails. The endeavour of constructing a parallel corpus for phishing emails in both English and Arabic presents its unique set of challenges, primarily due to the limited data availability. However, this task was approached innovatively through the generation of parallel synthetic data, employing human translation. This method involved translating English phishing emails into Arabic using human translators, followed by pairing them with their original English versions. This approach allowed us to build a comprehensive, bilingual dataset, serving as a significant resource in the field of phishing detection. In the following section, the methodology adopted for the creation of this parallel corpus of phishing emails in both English and Arabic is expounded upon. The insights derived from this process could potentially catalyse the advancement of multilingual phishing email detection methods.

3.4.1.1 Original dataset description

This study presents a new English–Arabic parallel phishing email corpus that has been developed from the anti-phishing share task text (IWSPA-AP 2018) corresponded with the 8th ACM Conference on Data and Application Security and Privacy in detecting phishing email through an anti-phishing shared task [274], which is a common practice associated with ML and text analysis in the area of cybersecurity. The organisers of IWSPA-AP 2018 [EDMB+18] provided the email

corpus. Several researchers [11], [20], [39], [112], [275]–[280] evaluated their models by using the IWSPA-AP 2018 dataset. The point of the anti-phishing shared undertaking is to assemble a classifier to differentiate phishing emails from spam and authentic emails. The two sub-tasks can be accommodated within unconstrained categories, which implies that members undertaking training may use any other external corpus. The anti-phishing shared tasks involve two sub-tasks: the first one is associated with the testing of emails with a header, while the second is associated with the testing of emails without a header. The descriptive statistics of training and testing email corpus related to these tasks are summed up in Table 3.2 and Table 3.3.

Table 3.2 Training email corpus details

Training Dataset	Legitimate	Spam	Total
With header	4082	501	4583
Without header	5088	612	5700

Table 3.3 Testing email corpus details

Testing Dataset	Data Samples
With header	4195
Without header	4300

3.4.1.2 Creating the corpus

For this study, the header-less version of the IWSPA-AP 2.0 training dataset was selected as the cornerstone to develop the English–Arabic parallel corpus. To translate the English content into Arabic, two potential strategies can be employed:

(i) Machine translation (MT), which leverages free APIs offered by service providers such as Google or Microsoft. This technique benefits from cutting-edge machine translation technologies, providing a means to accelerate the translation process.

(ii) Human translation (HT), which involves the translation of the English text from its onset. This approach guarantees high linguistic fidelity, preserving the subtleties and intricacies inherent in the original content.

By opting for either of these translation methodologies, the aim is to establish a precise and robust English–Arabic parallel corpus. This can significantly augment the understanding of phishing detection in order to bridge the gap between languages and increase the accuracy of detection mechanisms across linguistic boundaries. Human translation (HT) requires interpretation since it is not feasible to translate each word from the source document directly into the target language. Literal translations can alter the intended meaning of the source document, and thus, translators must exercise their linguistic and cultural understanding to convey the message accurately. For example, consider the English phrase "kick the bucket," which figuratively means "to die." A literal translation of this phrase into another language may not convey the intended meaning effectively. Therefore, human translators employ their expertise to comprehend the context and choose appropriate expressions or idioms that capture the essence of the source document while maintaining its intended meaning in the target language.

The goal of document translation is to effectively convey the same message from the source document using the target language. While there are MT tools available, HT is still considered to be the best method for translating written documents such as books, legal documents, manuals, product information, websites, personal documents, magazines, letters, and advertisements. This is because human translators take into account the grammar, idioms, conventions, and most importantly, the context of the original language when translating it to the target language, and preserving the meaning as close to the original as possible. MT can only translate the text from one language to another, that is, it cannot provide the same level of understanding and cultural nuances that a human translator can provide. See examples in Table 3.5.

So, a good translation is not provided by the first technique (MT) because of the translation quality. Therefore, the second one (HT) is used in this work, that is, translating from scratch with the aid of 10 volunteers who are English and Arabic language experts. The 10 translators were divided into two groups, with one group translating and the other checking the translation for grammatical

and spelling errors. The roles were then exchanged for the next batch and the process repeated until all the emails were translated. The translators were asked to make sure that

1) The Modern Standard Arabic language is followed.
2) A valuable sentence is written, which must end using a period (.).
3) Multiple phrases or words should not be typed.
4) The speech style used should be polite, and punctuation marks must be correct.
5) Factual data should only be provided when commenting on the email.
    a. One must not mention aspects that may occur in the future.
    b. Imagination and speculation shouldn't be present.
    c. Feelings related to the email scene should not be stated.
    d. Poetic style shouldn't be used excessively.
    e. Must not mention the nationality or names of places or persons, such as American Flag or Washington City.
    f. All essential details are to be mentioned and non-essential ones are to be ignored.

The translation and proofreading process took six months – from 15 September 2021 through 15 March 2022 – 12 weeks for translation, and 12 weeks for proofreading and quality control. Every month, 200 emails were translated and audited (100 legitimate emails and 100 phishing emails). In the first week of every month, 100 emails presented in an MS Word file were distributed equally to the 10 volunteers, with 10 new emails for each volunteer (5 legitimate e-mails and 5 phishing e-mails). In the first week of every month, the first group checked the emails translated by the second group, and the second group also checked the emails translated by the first group and ensured the level of accuracy of the translated text. The sixth volunteer checked the emails translated by the first volunteer, while the seventh volunteer checked the emails translated by the second volunteer, and so on. The translators completed the process of translating the text and its content to ensure its credibility and integrity. They should not delete or add anything on their own or according to their whims; they could add a few margins for clarification if the translator wanted. They could not highlight or show their personal view on the content of the text to be translated. Further, they have to take into account the nature of the repeated words and their meaning. The process was repeated in the third and fourth weeks of every month.

3.4.1.3 Judging the quality of the translation

Figure 3.1 illustrates the process of translating texts from English to Arabic, including the quality control procedures followed. The following criteria are used to judge the level of quality of the translation process:

- Coherence of meaning and work to achieve consistency.
- Integration and comprehensiveness.
- Matching the method.
- Grammar and spelling.

Table 3.4 shows a sample of the translation correction process, which includes checking the punctuation, rewording some sentences, and solving the ambiguity of some cultural expressions. Figure 3.2 also depicts a sample sentence taken from the corpus along with the sentence ID number, as well as English and Arabic translations of the sample sentence.

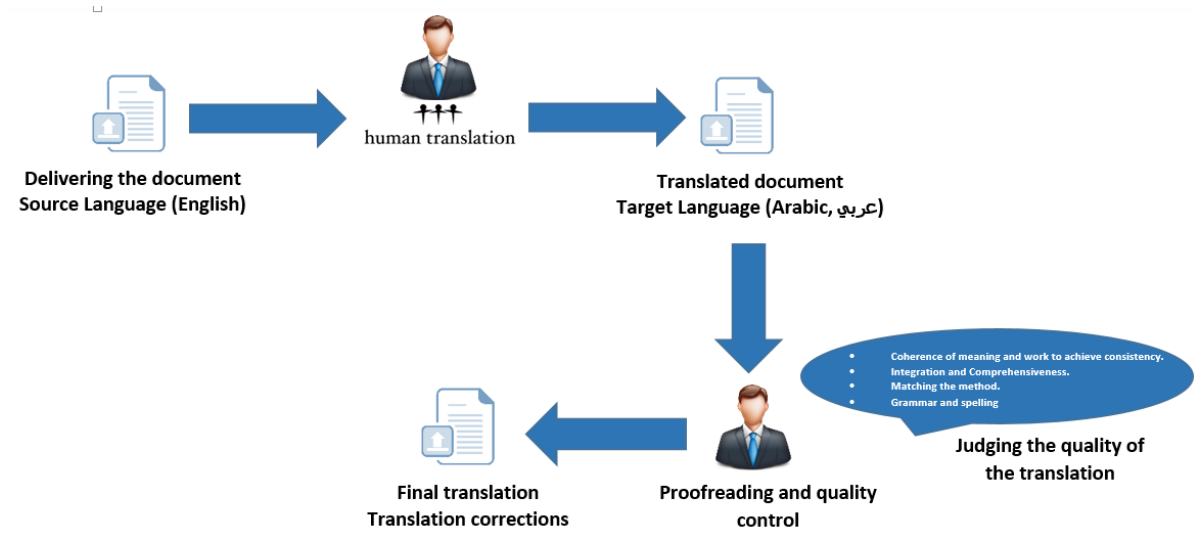


Figure 3.1 Human translation from English to Arabic

Table 3.4 Examples of translation corrections

No.	Source (English Text)	Translation (Arabic Text)	Corrected (Arabic Text)
1	This is an urgent notice from the board of governors federal reserve bank Washington DC. Open attached letter and read carefully and respond accordingly.	هذا اشعار عاجل من مجلس محافظي بنك الاحتياطي الفيدرالي في واشنطن العاصمة. افتح الرسالة المرفقة واقرأها بعناية واستجب وفقا لذلك.	هذا إشعار عاجل من مجلس إدارة محافظي البنك الاحتياطي الفيدرالي في واشنطن العاصمة. افتح الرسالة المرفقة واقرأها بعناية واستجب وفقًا لذلك.
2	As part of our duty to strengthening our security and improving your overall mail experience, we have detected your mail settings is out of date. We want to upgrade all email account scheduled for today. To Complete this procedure, CLICK HERE to upgrade your account. If your settings is not updated today, your account will be inactive and cannot send or receive message any longer.	كجزء من واجبنا في تعزيز أمننا وتحسين تجربة بريدك الإجمالية، اكتشفنا أن إعدادات بريدك قديمة. نريد ترقية جميع حسابات البريد الإلكتروني المقررة اليوم. لإكمال هذا الإجراء، انقر هنا لترقية حسابك. إذا لم يتم تحديث إعداداتك اليوم، فسيكون حسابك غير نشط ولا يمكنه إرسال أو استقبال الرسالة بعد الآن.	كجزء من واجبنا المتمثل في تعزيز الأمان وتحسين تجربة البريد بشكل عام ، اكتشفنا أن إعدادات البريد لديك قديمة. نريد ترقية جميع حسابات البريد الإلكتروني المجدولة لهذا اليوم. لإكمال هذا الإجراء ، انقر هنا لترقية حسابك. إذا لم يتم تحديث إعداداتك اليوم ، فسيكون حسابك غير نشط ولا يمكنه إرسال أو استقبال الرسائل بعد الآن.
3	This is to notify all Students, Staffs of organization that we are validating active accounts. Kindly confirm that your account is still in use by clicking the validation link below:	هذا لإخطار جميع الطلاب وموظفي التنظيم بأننا نتحقق من صحة الحسابات النشطة. يرجى التأكيد على أن حسابك لا يزال قيد الاستخدام من خلال النقر فوق رابط التحقق أدناه:	هذا لإخطار جميع الطلاب وموظفي المؤسسة بأننا نتحقق من صحة الحسابات النشطة. يرجى التأكد من أن حسابك لا يزال قيد الاستخدام من خلال النقر على رابط التحقق أدناه:

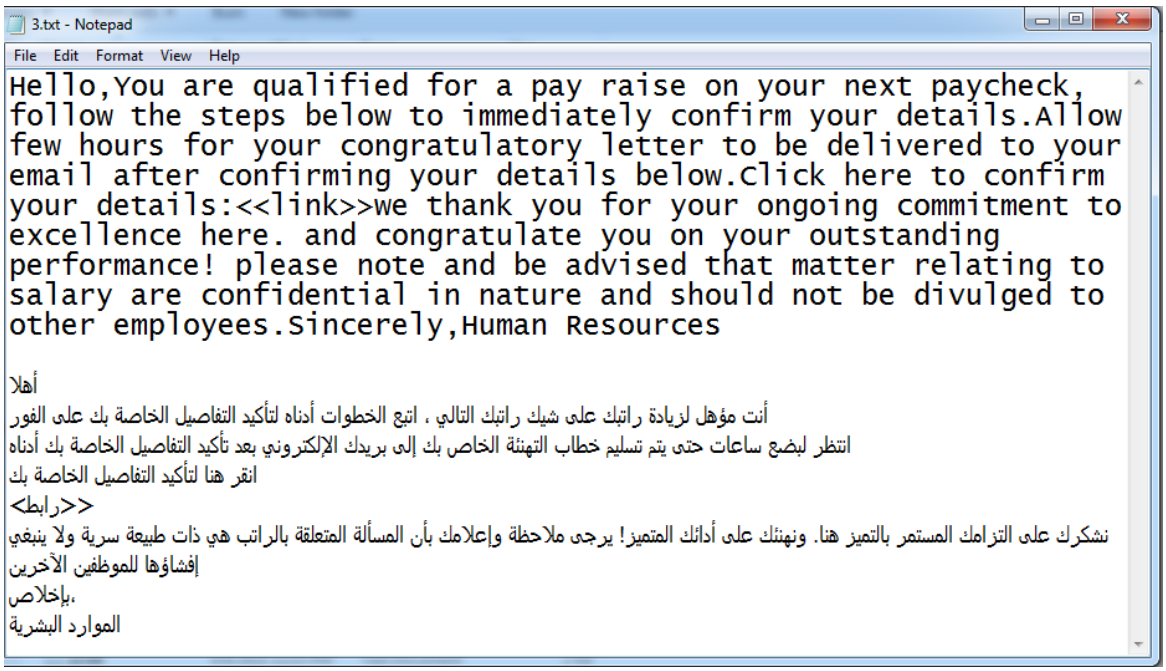


Figure 3.2 Example from the new English /Arabic parallel corpus

3.5 Arabic Phishing Email Corpus (APEC)

The creation of a corpus containing genuine Arabic phishing cases can serve as an indispensable tool for organisations seeking to bolster their defences against phishing attacks conducted in the Arabic language. Given the pervasive and increasingly sophisticated nature of such cyber threats, it is essential to have a robust database that is representative of these specific attacks. In the context of this study, the goal was to augment capabilities to identify Arabic phishing emails by harnessing an assemblage of authentic instances of these cyber threats. This corpus, a comprehensive collection of phishing cases, was painstakingly curated to ensure it adequately represents the multifaceted nature of phishing attacks that users face. Such a corpus, enriched with diverse examples of phishing attacks, can be utilised as a pivotal benchmarking resource for testing. It provides a practical, real-world platform to assess the efficacy of the ML models, which have been trained specifically to identify and flag phishing attempts. By comparing the outputs of these models against the corpus, the models' performance and accuracy can be monitored, which can help to make necessary adjustments to improve the overall precision of their predictive capability. Therefore, the corpus plays a crucial role not only in strengthening the cybersecurity measures of organisations but also in advancing the understanding of the evolving strategies deployed by phishers and helping to design more resilient ML models against such cyber threats.

3.5.1 APEC collection

Collecting a sample of Arabic phishing emails via snowball sampling was a challenging task, but it was necessary to obtain a high-quality sample for scientific research purposes. Snowball sampling is a non-probability sampling technique commonly used in social science research when the population of interest is rare, hard to reach, or difficult to identify. This sampling technique is also known as chain referral sampling, network sampling, or referral sampling. In snowball sampling, the researcher begins with a small group of participants, often referred to as the "seed sample," who are chosen based on their relevance to the research question (see Figure 3.3). After data collection from the initial sample, the researcher then asks the participants to refer other individuals they know who meet the inclusion criteria for the study. The new participants are then asked to refer others, and the process continues until the desired sample size is achieved. Here are

the steps followed in this study to collect phishing emails via snowball sampling from individuals who have received them:

1. **Identify an initial set of phishing emails**: The first step was to identify an initial set of Arabic phishing emails by asking cybersecurity experts or individuals who have received phishing emails.
2. **Contact individuals who have received phishing emails**: After the collection of an initial set of phishing emails, snowball sampling was used to identify additional individuals who have received phishing emails. They started by contacting the individuals who had sent them the initial set of phishing emails and asked the latter if they knew of anyone else who had received similar emails. Social media platforms and online forums were also used to reach out to individuals who may have received Arabic phishing emails.
3. **Analyse the data**: After a set of Arabic phishing emails was compiled, the data was analysed to identify patterns and characteristics of Arabic phishing emails. It can be difficult to determine whether an email is a phishing attempt or a legitimate message. However, several steps were followed to help determine the authenticity of the email:

    a) **Check the sender's email address**: Phishing emails often use a fake email address or one that is similar to a legitimate organisation but with slight variations (e.g., "amaz0n" instead of "amazon"). For this study, the sender's email address was checked carefully to ensure its legitimacy.

    b) **Check for spelling and grammar errors**: Many phishing emails contain spelling and grammar errors, which can be a red flag, signalling that the email is not legitimate. Legitimate organisations typically take the time to proofread their emails before sending them out.

    c) **Click on links or download attachments**: Phishing emails often contain links to fake websites or attachments that contain malware.

    d) **Check the message content**: Phishing emails often try to create a sense of urgency to get you to act quickly.
4. **Evaluate the quality of the sample**: Evaluating the quality of the amassed sample is a critical aspect of this study. This was accomplished by conducting an in-depth assessment of the sample emails' characteristics, which was led by a team of cybersecurity experts. The experts scrutinised the sample and rendered feedback concerning its representativeness

and suitability for research objectives. Figures 3.4 and 3.5 are graphical depictions of the procedure employed to appraise emails dispatched by individuals for their legitimacy or phishing propensity. These figures illustrate the multiple stages entailed in this evaluation process. The assessment protocol typically commences with a thorough analysis of the email content, considering numerous factors, including the structure, language, and embedded links, among others. Subsequently, the ultimate decision of acceptance or rejection of the sample is made. The outcome of this methodical evaluation not only strengthens the validity of this corpus but also ensures that it accurately reflects the diverse nature of phishing attempts encountered in real-world scenarios.

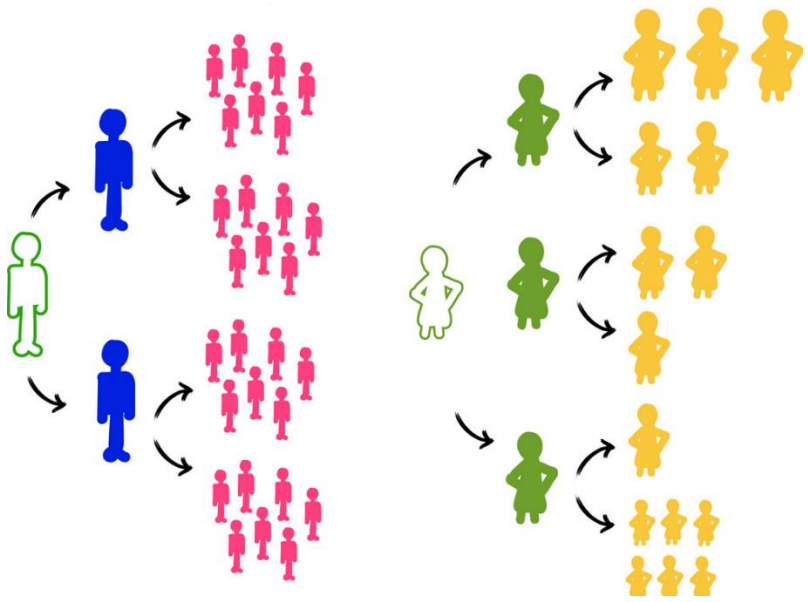
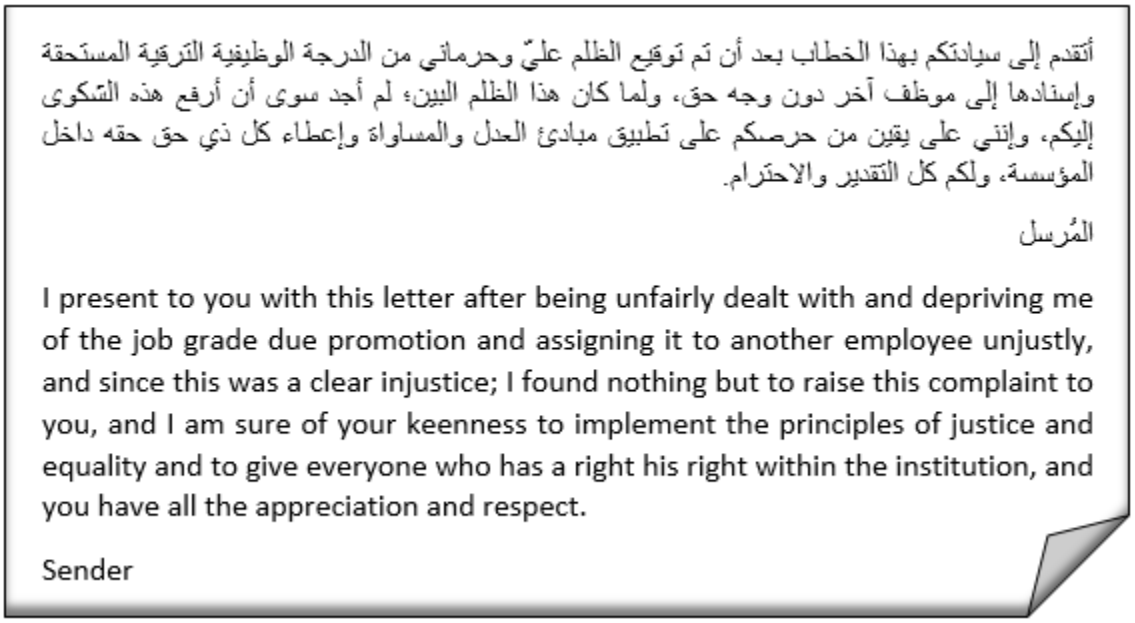


Figure 3.3 Snowball sampling


أتقدم إلى سيادتكم بهذا الخطاب بعد أن تم توقيع الظلم عليّ وحرماني من الدرجة الوظيفية الترقية المستحقة وإسنادها إلى موظف آخر دون وجه حق، ولما كان هذا الظلم البين؛ لم أجد سوى أن أرفع هذه الشكوى إليكم، وإنني على يقين من حرصكم على تطبيق مبادئ العدل والمساواة وإعطاء كل ذي حق حقه داخل المؤسسة، ولكم كل التقدير والاحترام.

المُرسل

I present to you with this letter after being unfairly dealt with and depriving me of the job grade due promotion and assigning it to another employee unjustly, and since this was a clear injustice; I found nothing but to raise this complaint to you, and I am sure of your keenness to implement the principles of justice and equality and to give everyone who has a right his right within the institution, and you have all the appreciation and respect.

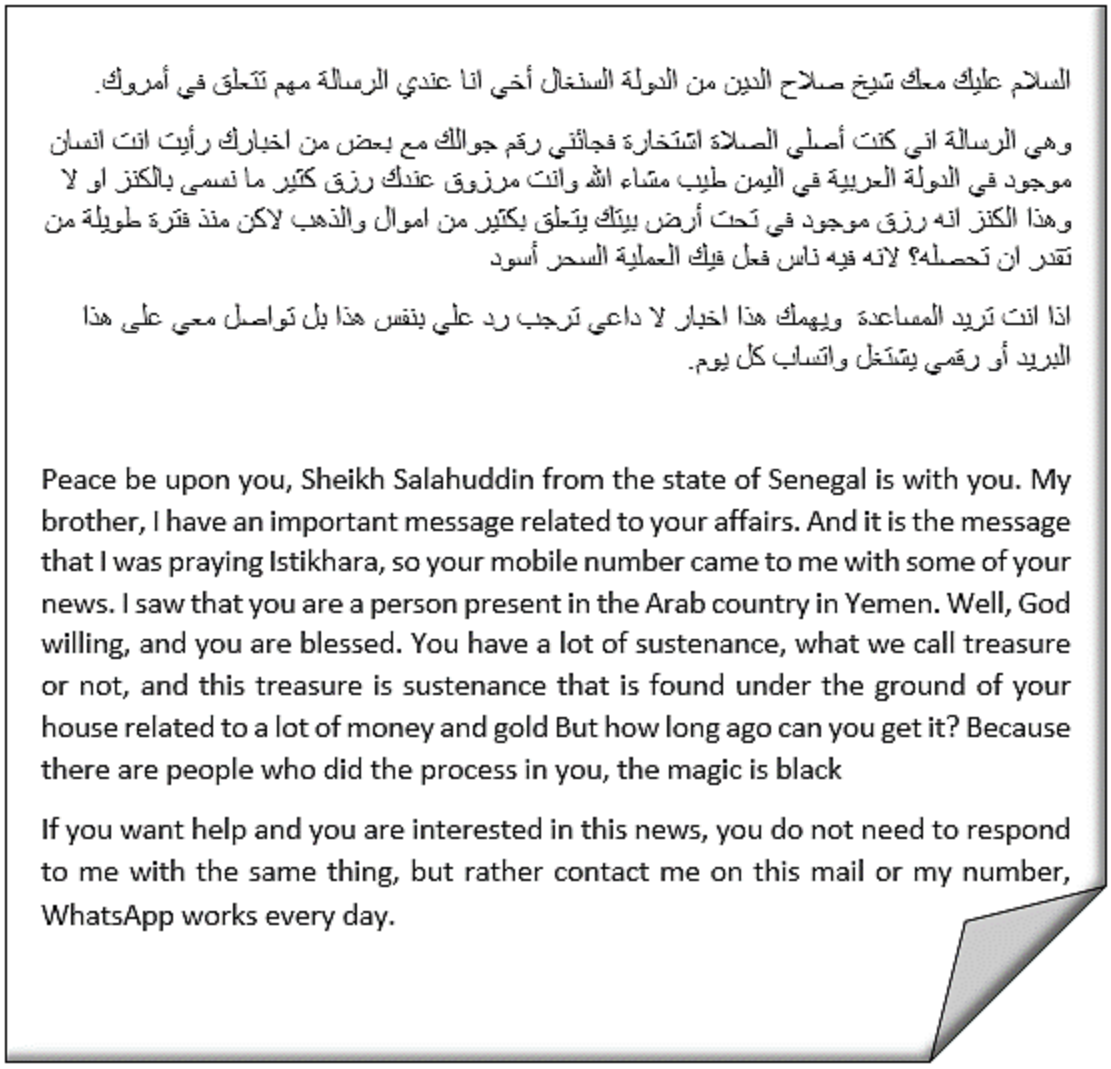
Sender

Figure 3.4 Example of the rejected sample

Analysis of phishing emails:

Based on the content of the email provided, it appears to be a legitimate email. The sender has addressed a concern regarding being deprived of a promotion and is seeking assistance in addressing the issue. The language used in the email is respectful and professional, and there is no sense of urgency or pressure to respond quickly. Additionally, the email does not contain any suspicious links or attachments, nor does it ask for any personal information.

Decision: Rejected.



السلام عليك معك شيخ صلاح الدين من الدولة السنغال أخي انا عندي الرسالة مهم تتعلق في أمروك.

وهي الرسالة اني كنت أصلي الصلاة اتستخارة فجائتني رقم جوالك مع بعض من اخبارك رأيت انت انسان موجود في الدولة العربية في اليمن طيب مشاء الله وانت مرزوق عندك رزق كثير ما نسمى بالكنز او لا وهذا الكنز انه رزق موجود في تحت أرض بيتك يتعلق بكثير من اموال والذهب لاكن منذ فترة طويلة من تقدر ان تحصله؟ لانه فيه ناس فعل فيك العملية السحر أسود

اذا انت تريد المساعدة ويهمك هذا اخبار لا داعي ترجب رد علي بنفس هذا بل تواصل معي على هذا البريد أو رقمي يشتغل واتساب كل يوم.

Peace be upon you, Sheikh Salahuddin from the state of Senegal is with you. My brother, I have an important message related to your affairs. And it is the message that I was praying Istikhara, so your mobile number came to me with some of your news. I saw that you are a person present in the Arab country in Yemen. Well, God willing, and you are blessed. You have a lot of sustenance, what we call treasure or not, and this treasure is sustenance that is found under the ground of your house related to a lot of money and gold But how long ago can you get it? Because there are people who did the process in you, the magic is black

If you want help and you are interested in this news, you do not need to respond to me with the same thing, but rather contact me on this mail or my number, WhatsApp works every day.

Figure 3.5 Example of the accepted sample

Analysis:

Based on the content of the email provided, it appears to be a phishing email. The email has an informal and unprofessional tone, and the sender has used a name that is not their own. The email also contains unusual requests and claims about a treasure or wealth that the recipient has under their house. This is a common tactic used by phishing emails to create a sense of urgency and convince the recipient to respond quickly. Additionally, the email contains spelling and grammatical errors, which is another red flag.

Furthermore, the email asks the recipient to contact the sender through a different email or WhatsApp number, which is a common tactic used by scammers to avoid detection and continue their fraudulent activities.

Decision: Accepted.

Upon completion of the email appraisal procedure, typically encompassing a comprehensive analysis of the email's content followed by a final adjudication, 150 samples constituting phishing emails composed in Arabic were accepted. Concurrently, a balanced corpus was compiled that incorporated an equivalent number of rejected legitimate emails from the sample pool. Thus, the final corpus stood at an aggregate of 300 emails, equally partitioned between phishing and legitimate instances. A balanced corpus ensures equal representation of all data categories or classes, thereby reducing potential analytical or training bias. It underpins more dependable and widely applicable conclusions, as the models trained on this data can accurately mirror and manage a variety of real-world instances. Consequently, a balanced corpus significantly enhances the calibre of predictive modelling, enabling superior outcomes in operations such as spam/phishing detection, sentiment analysis, and various automated classifications.

3.5.2 Arabic–English parallel corpus for email phishing

In an endeavour to establish a parallel corpus of Arabic and English emails, the task of translating authentic Arabic phishing emails into English was embarked upon. This procedure not only enabled us to conduct a comprehensive analysis of the email content but also assisted in discerning recurrent characteristics and patterns.

To facilitate the translation process, machine translation tools, especially Google Translate, were leveraged. This platform possesses the capability to process a substantial volume of emails swiftly and efficiently. While Google Translate demonstrates reliability and effectiveness for general use, its translations are not invariably flawless, given its programmatic limitations in discerning subtler linguistic nuances such as metaphors, symbolism, and regional dialects. Therefore, its application to tasks involving contractual, business, legal, or medical documents is not recommended; these instances would benefit from a translation service involving expert native speakers.

In this study, several measures were implemented to confirm the accuracy of the translations, including a manual review of a subset of translated emails compared with their original Arabic versions. The translation of real Arabic phishing emails into English provided valuable insight into the strategies employed by attackers in these cyber threats.

The knowledge gleaned from this translation process was subsequently utilised to augment and refine phishing detection systems, thereby bolstering the organisation's defence against phishing attacks' deleterious impacts. Google Translate-based machine translation was employed to inspect translation outcomes with the Arabic machine translation system, noting its handling of agreement and word order. The forthcoming steps detail the procedure adopted in this respect:

STEP 1: The source text (phishing / legitimate email) is entered in the Arabic language.
STEP 2: The source text is passed to Google Translate, and the output is obtained as (English text).
STEP 3: Review the translation to assess its quality. Google Translate is a machine translation system, which means that it uses algorithms to automatically translate text from one language to another. While it can be a helpful tool, it is not always 100% accurate, and the quality of the translation can vary depending on the complexity of the text and the specific nuances of the languages involved.
STEP 4: If necessary, make edits to the translated text to improve its accuracy and readability. This can be done by clicking on the pencil icon below the translated text box, which will allow you to make changes to the translation.
STEP 5: Check the grammar and spelling of the translated text to ensure that it is correct. You can use an online grammar checker or spell-checker to assist with this step.
STEP 6: Review to identify irregular word(s) (if any).
STEP 7: Once you are satisfied with the quality of the translated text, copy and paste it onto a document.
STEP 8: On the next email, repeat Steps 1 to 7.

Figure 3.6 depicts the process of translating Arabic texts into English, while Figure 3.7 showcases a sample email from the corpus. The sample email is identified by its unique email ID number, and both Arabic and English translations of the email are provided.

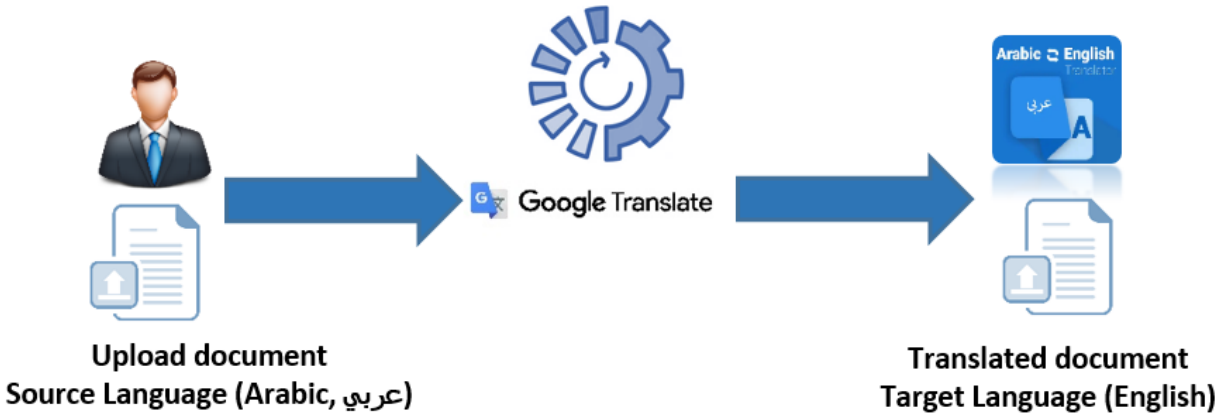


Figure 3.6 Automatic translation from Arabic to English

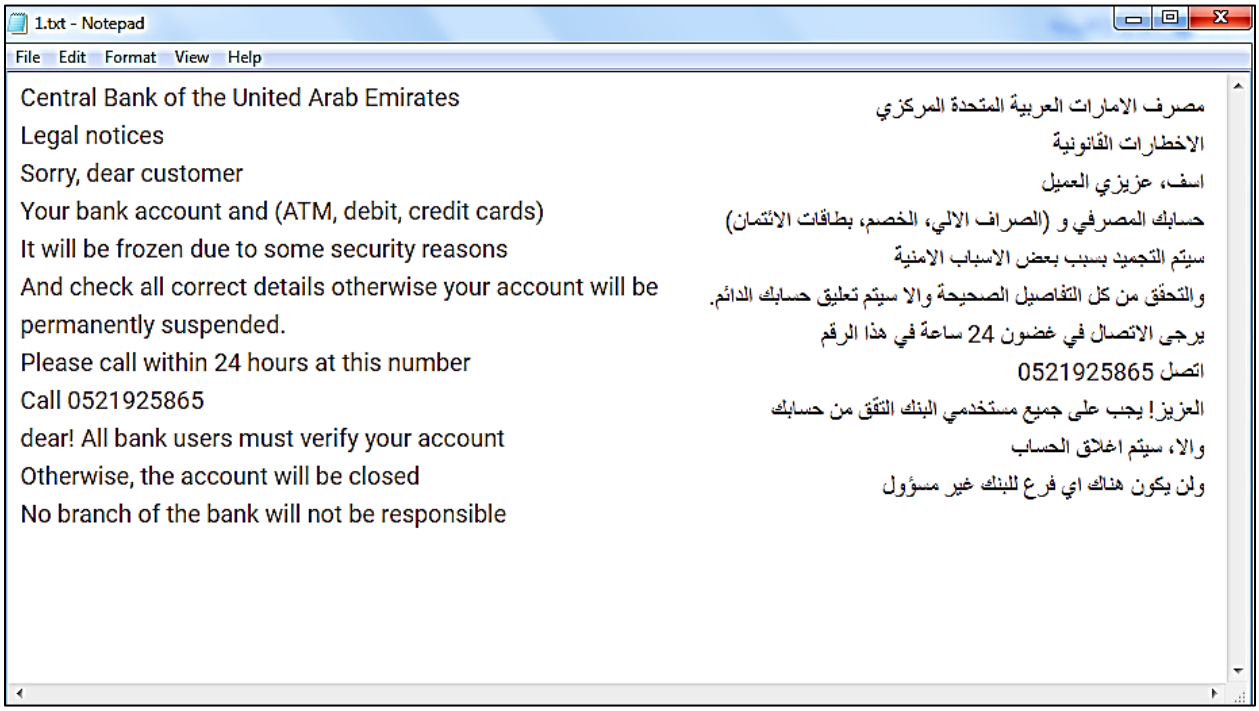


Figure 3.7 Example from the new Arabic/English parallel corpus

3.6 English-to-Arabic translation: challenges and solution

The Arabic language belongs to the Semitic language family, whereas the English language is part of the Indo-European language family. It is significantly difficult to translate Arabic into English or the other way round. The issues related to translation can be categorised into two groups: linguistic and cultural. The linguistic group contains pragmatic rules, morphology, lexicon, syntax, and textual and rhetorical conflicts. For this purpose, Modern Standard Arabic (MSA) is used in academia, literature, print, and mass media. MSA is a systemised form of the Arabic language that slightly contradicts classical Arabic and local dialects.

Since many words from the Arabic vocabulary have no profound English alternatives, Arab translators find it problematic to put forward a word-to-word translation, and that is where cultural issues appear. For instance, تيمم tayammum is an Arabic word that corresponds to "an Islamic ritual of performing ablution with clean mud or sand for purification in the absence of water", but there is no similar notion for it in English [281].

Translating the Arabic language into the English language is a semantically troublesome process. In this regard, machine translation has proven to be more difficult than human translation. Students from the Saudi University of Translation listed the following pitfalls of using machine translation [282]:

1) Arabic sentences are relatively long.
2) Compound structure of a sentence.
3) The content of Arabic phrases is organised in such a way that it is syntactically ambiguous for machine translators to understand correct grammatical relations among words.
4) There are various definitions of Arabic words.
5) The Arabic language is constructed from 28 letters, a handful of which do not have any English alternative.

According to research [283], there is a stark problem experienced by graduate students in translating cultural phrases from Arabic to English. These are,

1) Being oblivious to the culture-related statements
2) Absence of equivalent vocabulary in another language
3) An ambiguity of cultural references and statements
4) Non-proficient translation strategies and methods.

Translation involves a source language and a target language, and these languages significantly vary in their religious cultures. A translator will be able to correctly translate the actual meaning of the text only if he/she is aware of certain linguistic factors and has a profound grip over the target language to comprehend cultural values correctly. There exist two sorts of meanings: 1) denotative meaning, which refers to the verbal explanation of a particular word; for instance, the denotative meaning of "dear" is "غزال"; 2) connotative meaning, which points to the metaphorical idea behind the word; for instance, the connotative meaning of "dear" is "جمال" أو "رشاقة". These linguistic concepts make the Arabic-to-English translation quite tricky and challenging.

Translators should have a good understanding of denotative and connotative meanings. For instance, the pronoun "it" in English can be used in various manners as both subject and object or while mentioning non-living entities or animals. Meanwhile, in the Arabic language, there is no equivalent to "it"; every entity is described as masculine or feminine. So, denotative meaning is one of the major causes of making translation problematic, as it does not provide the deliberate meaning of the sentence. Refer to Table 3.5 to understand translation dissimilarities between denotative and connotative meanings for the Arabic and English languages.

Table 3.5 Examples of translation differences between Arabic and English languages

No.	English	Arabic equivalent
1	It rains cats and dogs	أنها تمطر بغزارة
2	Hello	السلام عليكم
3	Mailbox	صندوق بريد
4	Thesaurus	معجم الالفاظ المترادفة والمتضادة
5	Hepatitis	التهاب الكبد الوبائي
6	Mediterranean fever	حمى البحر الابيض المتوسط
7	Time lapse	تقنية التصوير السريع
8	It is the milkman	أنه بائع الحليب
9	Forbidden fruit is sweet	كل ممنوع مرغوب
10	I don't want to step on your feet	لا أريد أن أتخطاك
10	Two minds are better than one	رأيان أفضل من رأي واحد
11	Diamond cut diamond	لا يفل الحديد الا الحديد
12	Air time	مدة المكالمة الفعلية
13	MP3 Player	مشغل وسائط صوتية
15	Bluetooth	تقنية الاتصال القصير
16	Focus Pixel	دقة العرض/دقة الشاشة
17	Time lapse	تقنية التصوير السريع
18	Chickenpox	جدري الماء
19	You warm my heart	أثلجت قلبي
20	Diabetes	مرض السكري
21	Please	من فضلك
22	Dear all	أعزائي

All problems must have some solution. A good translation requires the translator to be specific with language points. Every language has its own specialty and grammatical structures. A translator must have a good understanding of the denotative and connotative meanings behind a sentence to the abstract accurate meaning of every word. Only then, they will be able to translate a piece of text accurately. A prominent issue with the translation is the plurality of meanings, that is, there can be different meanings of a single word. Therefore, a proficient translator must know different cultures along with being bilingual. To rectify translation-related problems, below mentioned criteria are followed to propose a sound solution [284].

The type to which text belongs can be either general, political, technical, or religious, etc.

1. Linguistic context, such as the former and latter word or clause in a sentence.
2. Possible translation of a certain word in the target language, such as “fat salary” can be translated as "راتب ضخم", which means “huge salary”, but "راتب سمين" cannot be used, which verbally means “fat salary”.

3. Check whether a similar target language grammatical formation can be used or not.
4. How comprehensive an expression can be? For instance, "tall order" is comprehensible if it is translated as "مهمة شاقة", meaning “a daunting task”, but it cannot be explained by using " طلب طويل", which means “long request”.

3.7 Summary

This chapter described the creation of a novel corpus – the English–Arabic Phishing Email Corpus. This resource was established through the meticulous human translation of English email bodies drawn from the IWSPA-AP v2.0 dataset. The corpus will predominantly serve as a training platform for ML models, allowing for the designing of robust models adept at detecting Arabic phishing emails with high levels of precision. In addition to this, a collection of 300 Arabic emails was amassed, intended to serve as a test bed for the said model. To ensure the robustness of this study’s validation techniques, these emails were translated into English using GT. This new collection is poised to be a key asset in assessing this system's proficiency in handling multilingual content. By simultaneously leveraging these two data sources, the aim is to significantly enhance the ability to intercept phishing emails in both English and Arabic.

Chapter Four

A New Model for Detecting Arabic–English Phishing Attacks

4.1 Overview

This chapter provides a comprehensive overview of the proposed model for detecting phishing emails using text features derived from the inherent properties of the emails. The devised model comprises a series of carefully constructed steps aimed at optimising phishing detection. The initial phase of this process emphasises the critical choice of data sources. Data selection is central to the model as it determines the quality and scope of the information on which the subsequent steps will operate. A blend of diverse datasets is ensured, incorporating a wide array of legitimate, suspicious, and phishing emails. By doing so, numerous possible phishing strategies and benign email formats are accounted for. The next critical step after data selection is data pre-processing. This stage involves a series of operations to convert the raw email data into a more digestible format for the models. These operations may include lowercasing all the text, punctuation removal, tokenisation (converting sentences into individual words) and stop word elimination. This step also includes handling unstructured text data, removing any potential noise in the data, such as HTML tags, email headers, non-textual content and so on, and dealing with misspellings often found in phishing emails. After data pre-processing, feature extraction is carried out. In the context of this framework, this consists of two primary techniques – word embeddings and character embeddings. Word embeddings, such as FastText, convert words into fixed-sized dense vectors so that words with similar meanings are placed close together in the vector space. This method allows us to capture the semantic context of words, including those in phishing emails. Character embeddings go one step further, breaking down words into individual characters or subwords, thereby capturing the data's morphological nuances, spelling variations and more granular structures. This is particularly beneficial when dealing with phishing emails that intentionally employ misspellings or uncommon formulations to evade detection. After processing and transforming the raw email data into a suitable format, these features are fed into the Machine Learning/Deep Learning (ML/DL) classifiers. These classifiers, trained on the extracted features, distinguish between legitimate, suspicious and phishing emails. A range of classifiers, including

DTs, RF, SVM, and NN, were experimented with, and the parameters and architectures were optimised as needed. The final step of the proposed model involves the selection of the best-performing algorithms based on their respective performance metrics, such as accuracy, precision, recall and F1 score. These optimal algorithms form the core of the phishing detection model, capable of accurately identifying and distinguishing between legitimate, suspicious and phishing emails. Through this comprehensive and multi-step framework, the aim is to create a robust, effective, and scalable solution for detecting phishing emails, thereby minimizing the risks posed by such threats in today's digital world.

4.2 Proposed multi-stage approach to detect phishing emails

To address the pressing issue of the absence of an effective system for filtering English–Arabic phishing emails, an innovative solution was devised: the EAPD model. The primary objective of the EAPD model is to explore the capabilities of ML and NLP methods in effectively detecting and thwarting phishing attacks in English and Arabic. By integrating a versatile array of techniques, this model can effectively handle the complexities inherent in each language, catering to their unique linguistic features, syntax and semantics. The inclusion of word-level techniques, such as TF-IDF and DTM, allows the model to understand the importance of individual terms in the context of phishing emails, enabling it to differentiate between legitimate and malicious content more accurately. Meanwhile, FastText embedding can represent words as continuous numerical vectors, facilitating semantic analysis and similarity detection. Recognising the specific challenges of Arabic text, the model's incorporation of a character-level CNN (CharEmbedding) becomes paramount. This component enables the EAPD model to learn meaningful patterns and relationships between Arabic characters, compensating for the scarcity of resources and linguistic complexities prevalent in the Arabic language. By developing the EAPD model, there is an aspiration to significantly improve phishing detection by extending its effectiveness to Arabic phishing emails. This approach addresses a critical gap in email security and highlights the potential for multi-lingual solutions to combat phishing attacks effectively by pushing the boundaries of ML and NLP. As phishing attempts evolve in terms of sophistication and diversify across languages, the EAPD model represents a crucial advancement in the defence against such threats, safeguarding users in English and Arabic-speaking communities.

Figure 4.1 illustrates the comprehensive system architecture of the EAPD model. This architecture encapsulates the intricate design and integration of various components, including word-level techniques such as TF-IDF, DTM and FastText embedding and the CharEmbedding. Together, these elements synergistically enable the EAPD model to effectively address the challenge of phishing detection in English and Arabic. The figure represents the model's sophisticated framework, demonstrating its potential to significantly advance phishing email detection research in multi-lingual contexts.

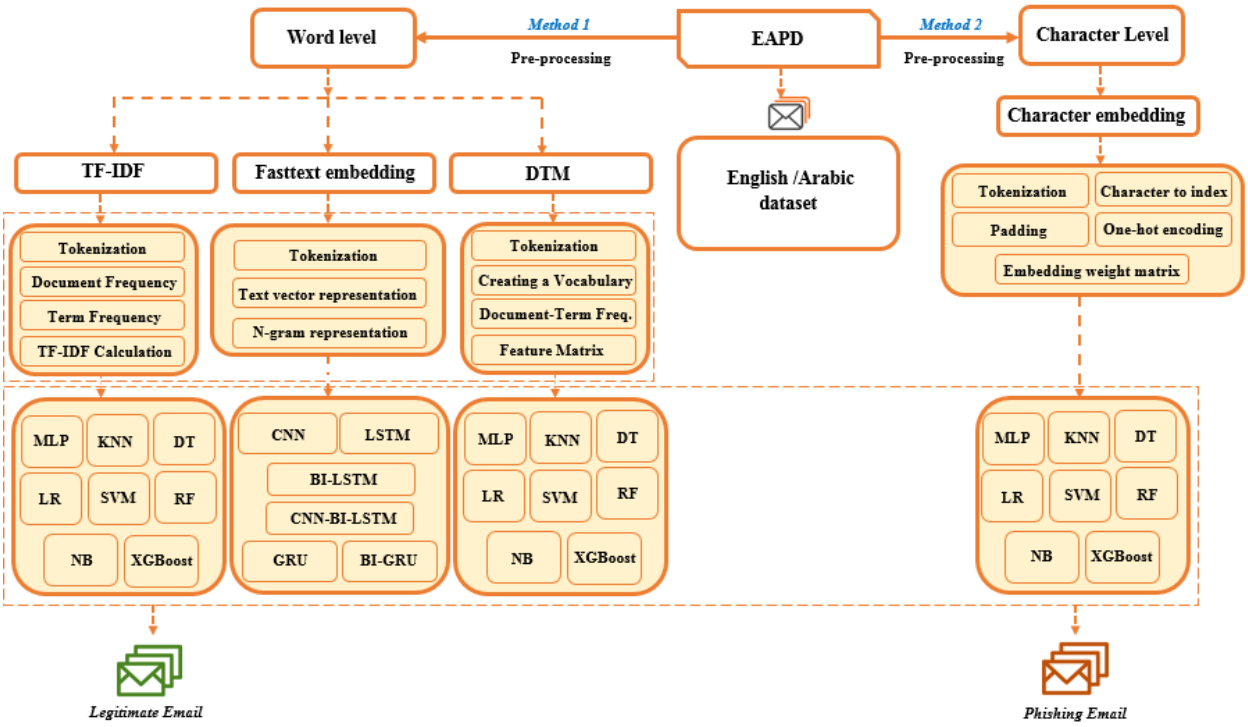


Figure 4.1 System architecture for EAPD model

4.3 Text pre-processing and feature extraction

The general methodology and resources needed to complete the phishing email detection task have been detailed in this section. Data source, data pre-processing, feature extraction (word level and character level), model training and model evaluation are the five primary stages of any NLP activity. The sequence of these stages is depicted in Figure 4.2. In this section, it will be demonstrated that the judicious application of text pre-processing techniques can significantly enhance the efficacy of text classification (TC) tasks for Arabic and English corpora. To this end, the examination focused on how popular strategies, such as the removal of stop words, affected the classification accuracy of the following ten acknowledged algorithms: NB, RF, DT, MLP, XGBoost, LR, SVM, KNN, CNN, and RNN.

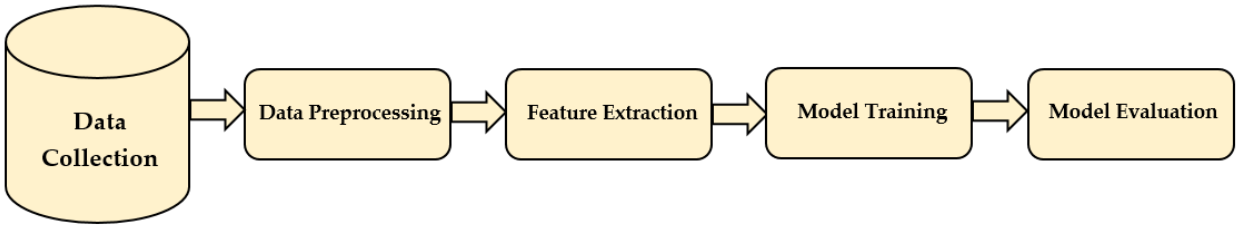


Figure 4.2 General flowchart of main stages for NLP task

4.4 Parallel corpus before text cleaning

Table 4.1 presents the statistics on the English–Arabic parallel corpus, which was created from the human translation (HT) of the English text provided by the (IWSPA-AP 2018) English corpus, 300 real email cases in Arabic and their translated counterparts in English using machine translation (MT) before text cleaning, including Arabic and English words, numbers and special characters for phishing and legitimate emails. Subsequently, numbers, special characters will be removed, and both English and Arabic words will be stemmed before using them as features in the classification process.

Table 4.1 English–Arabic parallel corpus for email phishing

Language	Type		Total
	Legitimate emails	**Phishing emails**	
English Words **(IWSPA-AP 2018)**	112320	44334	**156654**
Arabic Words **(HT)**	107035	43474	**150509**
Arabic Words **(Real case)**	23849	11191	**35040**
English Words **(MT)**	27839	12697	**40536**
Numbers	21161	3466	**24627**
Special characters	60001	22801	**82802**
Total	**352205**	**137963**	**490168**

4.5 Text cleaning and pre-processing

Textual data, especially when extracted from digital communication such as emails, is often riddled with inconsistencies, irregularities, and noise. Cleaning and pre-processing this raw data are pivotal steps in any NLP pipeline, particularly when focusing on the intricacies of phishing email detection. The quality of the input data significantly influences the performance of subsequent steps, including feature extraction and model training. The process becomes even more critical and complex when working with bilingual datasets, such as English and Arabic text, given their linguistic and structural differences. English and Arabic represent two distinct script systems and linguistic structures. English, a Germanic language, follows a left-to-right script with relatively straightforward tokenization based on spaces. On the other hand, Arabic, a Semitic language, employs a right-to-left script with non-trivial morphological intricacies [285]. Owing to Arabic's rich morphology, a single word can convey information that, in languages such as English, might be spread across several words [286]. Thus, pre-processing strategies that work for English may not necessarily be effective for Arabic, necessitating a specialized approach for each. Phishing emails further add layers of complexity to this task, as they often employ obfuscation techniques, such as homoglyphs and domain spoofing [287], which makes the cleaning process vital for accurate detection. Moreover, the dialectal variations in Arabic can pose challenges in standardizing and pre-processing the text [288]. Given these intricacies, this section delves into the tailored methodologies and strategies adopted for text cleaning and pre-processing, aiming to enhance the accuracy and efficiency of phishing email detection in English and Arabic text. The following steps were adhered to for each email, to eliminate irrelevant characters and symbols:

- Retrieve content from phishing and legitimate emails
- Lowercase all the words as the models are not designed as case-sensitive
- Eliminate punctuation and numbers in English and Arabic text
- Remove any strange characters that were used on the keyboard
- Change numbers to text and eliminate them in English and Arabic text
- Employ an improved version of the Natural Language Toolkit (NLTK) English corpus, eliminating stop words from the content such as "*its, is, a, an, the, for* and *that*" and many others (see Appendix A10 for details)
- Use the NLTK Python library to tokenise the texts
- Employ dediacritise, a function that eliminates Arabic diacritics found sporadically in Arabic text and often regarded as noise. Shorter vowels, for example, shadda (germination marker) and the dagger alif (for example, مُدَرّسَـةُ mudar~isaħu to مدرسة mdrs~ħ).
- Normalise Arabic text (alef_maksura, alef and teh_marbuta) using CAMeL Tools. Because of the rich morphology of Arabic, it is important to normalise text using a variety of methods to eliminate noise and sparseness. The following are the most prevalent normalisations: dividing concatenated segments is a part of Unicode normalisation (for example, لا to ل and ا), merging multiple different incarnations of a character into a singular established (for example, ع , عـــ and عـ to ع) and transforming extensions to the equivalent Arabic character in the Arabic character set utilised for Persian and Urdu (for example, گ to ك).
- Consider the keyword pattern and transform all the keywords to lowercase.

4.5.1 Elimination of English stop words

A stop word is a term that occurs repeatedly in a text but has no meaningful details or signals of the processed text's topic. Two strategies are used to compile a collection of stop words. The first is a rule-based technique that utilises morphological analysis (for example, [289]). The produced list is a domain-independent list that is intended for broad usage. On the other hand, the second method comprises a statistical approach involving utilising a corpus's frequency feature (for example, [290]). This is frequently the situation when a domain-dependent list needs to be built

for a particular sector. Stop word deletion is important in TC and other text-processing systems' pre-processing phases. Information retrieval [291], [292], text summarisation [293], [294] and MT [295] are only a few examples. Herein, it is important to remember that stop word deletion in TC does not just pertain to elements; nouns and verbs are also regarded as stop words.

Furthermore, according to [296], the arbitrary elimination of stop words can considerably degrade TC accuracy and produce unexpected findings. The strategy involves using a stop word list for common terms that contains all fundamental stop words and their derived variants, as the goal is to increase pre-processing effectiveness without harming TC accuracy. The Natural Language Toolkit (NLTK) library extracted 179 fundamental English stop words (see Appendix A10 for details). The NLTK is a collection of symbolic and statistical NLP source codes, data sets, lessons and tasks [297]. NLTK is a Python package released under the GPL open-source license, which has grown in popularity in education and academia over the last three years as a comprehensive library for experimenting with natural language.

Pre-processing, the initial stage in detecting phishing attacks, ensures reliable detection by eliminating superfluous words from emails. Stop word elimination and stemming are conducted in the pre-processing step. Take a database *D* that contains *f* number of emails, with *Ei* being the ith email. Each email comprises words grouped into sentences or paragraphs. Therefore, the stop words that exhibit inappropriate words in the text document, such as *a, in, an, the* and so on, are eliminated from every individual mail. Stop word elimination entails deleting stop words from an email through search. This is accompanied by the stemming concept, which involves converting some words in a text document to their root words. The extension of root words increases the intricacy associated with reading. Lastly, the pre-processing step generates dictionary words, which, in turn, are tagged as dictionary words wherein the features are extracted. Let us define the dictionary words scale as [$f \times d$], where *f* denotes the total quantity of emails in the database and *d* is the number of words in each email.

4.5.2 Elimination of Arabic stop words

Arabic stop words are common Arabic words with little semantic value and are often excluded from analyses to reduce the size of the data set and enhance its quality. Examples of Arabic stop

words include articles, prepositions, conjunctions and pronouns, such as *"the", "a", "an", "of", "in", "and", "or", "to", "from", "that", "this", "it", "he", "she", "they", "with" and "without"*. The removal of these words can shift the focus to the more important words and phrases, which, in turn, can improve the accuracy of TC, clustering and other NLP tasks. The following is a list of commonly used Arabic stop words [298]:

أن، في، على، من، إلى، و، هذا، هذه، هنا، هناك، هو، هي، كان، كانت، لا، ما، مع، ماذا، مايو، مساء، مع، معه، نحن، نهاية، هذا، هذه، وإلى، والذي، والذين، ولكن، وليس، وهو، وهي، يكون، يمكن، يوم، يوما، يوجد، يكن، يكونون، يمكن، يمكننا، يمكنه، يمكنها، يمكنهم، يمكنهن، يجب، يكون، يكونوا.

Many online lists of Arabic stop words can be referenced when working with Arabic text. Some popular sources include the NLTK Arabic stop word list (see Appendix A11 for details) and the "Stopwords Arabic Extended" list, which contains a new list of around 751 stop words [298] (see Appendix A12 for details). The NLTK Arabic stop word list and "Stopwords Arabic Extended" list are utilised to obtain a more comprehensive list of Arabic stop words. These two resources combine to generate a list of 857 commonly used Arabic words typically excluded from text analysis. The NLTK Arabic stop word list and the "Stopwords Arabic Extended" list are valuable resources for anyone working with Arabic text data, as they provide a reliable and standardised way to pre-process and clean textual data for further analysis. However, these resources do not provide an exhaustive list and other words that could be considered stop words based on the context in which they are used. Additionally, the use of stop words in Arabic may vary depending on the dialect and regional variations.

The initiative of expanding the Arabic stop word list may or may not improve the performance of ML models that use Arabic text as input, depending on the specific application and data set being used. Stop words comprise common words often removed from the text before analysis, as they are unlikely to carry meaningful information. However, the specific set of stop words can vary depending on the application and the language being analysed.

Expanding the list of Arabic stop words could potentially improve the performance of ML models sensitive to the inclusion of irrelevant words. By removing more unnecessary words, the model

may be better able to focus on the most important features of the input data. However, using a larger stop word list could also remove words that may be important in certain contexts. Expanding the list may also require more computational resources and time for pre-processing the data. In summary, the impact of expanding the Arabic stop word list on model performance will depend on the specific application and data set being used. Therefore, it is important to carefully consider the potential benefits and drawbacks before changing the stop word list.

The choice of stop words in any language, including Arabic, is typically based on certain foundations and considerations. The following are some common factors that influence the selection of Arabic stop words:

1. Frequency: Stop words are often chosen based on their frequency of occurrence in the language. Words that appear frequently but carry little semantic meaning, such as articles, prepositions and pronouns, are good candidates for inclusion in the stop word list.

a. <u>Articles</u>

- "Al-Mudhakkar al-Mu'rabah al-Jam'a" المذكر المعرفة الجمع - Masculine Definite Article:
  الـ (Al-) - Used before most masculine singular nouns.
  Example: الكتاب (Al-kitāb) - "the book"
- "Al-Mu'annas al-Mu'rabah al-Jam'a" المؤنث المعرفة الجمع - Feminine Definite Article:
  الـ (Al-) - Used before most feminine singular nouns that start with a consonant.
  Example: البنت (Al-bint) - "the girl"
  المـ (Al-) - Used before most feminine singular nouns that start with a vowel or "hamza" (glottal stop).
  Example: المدينة (Al-madīnah) - "the city"
- "Al-Mu'rabah al-Jam'a li al-Mudhakkar wa al-Mu'annas" المعرفة الجمع للمذكر والمؤنث - Plural Definite Article (for masculine and feminine):
  الـ (Al-) - Used before most plural nouns.
  Example: الكتب (Al-kutub) - "the books"

b. Prepositions

- في (fī) - in, at, on
- على ('alā) - on, upon, over, about
- مع (ma'a) - with
- من (min) - from, of
- إلى (ilā) - to, towards
- عن ('an) - about, concerning, regarding
- ب (bi) - with, by, in, at
- ل (li) - for, to, in order to
- حتى (ḥattá) - until, up to, so that
- بين (bayna) - between, among
- عند (ʿind) - at, with, near
- فوق (fawqa) - above, over
- تحت (taḥta) - under, beneath
- خلف (khalf) - behind, after
- أمام (amām) - in front of, before
- جانب (jānib) - beside, next to
- بعد (ba'd) - after, behind, later
- بدون (bidūn) - without
- ما عدا (mā 'adā) - except, beside

The following is a list of personal pronouns in Arabic, including subject pronouns and object pronouns:

c. Subject Pronouns:

- أنا (anā) - I
- أنتَ (anta) - You (masculine singular)
- أنتِ (anti) - You (feminine singular)
- هو (huwa) - He
- هي (hiya) - She
- نحن (naḥnu) - We

- أنتم (antum) - You (masculine plural)
- أنتن (antunna) - You (feminine plural)
- هم (hum) - They (masculine)
- هن (hunna) - They (feminine)

d. <u>Object Pronouns:</u>

- مَني (manī) - Me
- مَنْكَ (manka) - You (masculine singular)
- مَنْكِ (manki) - You (feminine singular)
- مِنْهُ (minhu) - Him
- مِنْهَا (minhā) - Her
- مِنَّا (minnā) - Us
- مَنْكُمْ (mankum) - You (masculine plural)
- مَنْكُنَّ (mankunna) - You (feminine plural)
- مِنْهُمْ (minhum) - Them (masculine)
- مِنْهُنَّ (minhunna) - Them (feminine)

It is important to note that Arabic pronouns can have different forms depending on the grammatical case (nominative, accusative, genitive) and the position in a sentence. The forms provided here are for the nominative case, that is, the subject form.

2. Linguistic analysis: Linguists and language experts analyse the structure and grammar of the Arabic language to identify function words rather than content words. Function words, similar to conjunctions and auxiliary verbs, are often included in stop word lists.

<u>a. Conjunctions</u>:

- وَ (wa) - and
- أو (aw) - or
- لَكِنْ (lakin) - but
- إِذَا (idhā) - if, when
- لِأَنَّ (li'anna) - because
- بَعْدَ أَنْ (ba'da an) - after

- قَبْلَ أَنْ (qabla an) - before
- عِنْدَمَا (ʿindamā) - when
- حَتَّى (ḥattā) - until
- إِذْ (idh) - when, as

b. Auxiliary Verbs:

- يَكُونُ (yakūnu) - to be (present tense)
- كَانَ (kāna) - to be (past tense)
- سَيَكُونُ (sayakūnu) - to be (future tense)
- يَجِبُ (yajibu) - must, should
- يُمْكِنُ (yumkinu) - can, may
- يَحْبَسُ (yaḥbasu) - must, have to
- يَجِدُ (yajidu) - to find
- يُفَضِّلُ (yufaḍḍilu) - to prefer
- يُرَجِّحُ (yurajjihu) - to suggest, to indicate

3. Contextual relevance: Stop words should be chosen based on their lack of significant meaning in isolation and their limited contribution to understanding a text. Words crucial for conveying meaning or providing important context are typically excluded from the stop word list.

4. Corpus analysis: Analysing large collections of Arabic texts (corpora) helps identify common words that occur frequently across different domains and genres. Corpus analysis aids in identifying words that are likely to be stop words.

5. User requirements: The choice of stop words can also be influenced by the specific needs of users or applications. Certain words may be considered more or less important as stop words depending on the task.

In this context, it is important to note that there are different approaches and variations in the selection of stop word lists. Furthermore, there may not be a universally agreed-upon set of Arabic stop words. The choice of stop words can vary depending on the purpose, such as sentiment

analysis [299]–[302], topic classification [303]–[305] or document classification [306]–[308] and users' or researchers' domain and preferences. After examining the tokens extracted from the text, established lists such as the NLTK Arabic stop words and the "Stopwords Arabic Extended" list were reviewed.

In the quest to optimize the identification process of Arabic phishing emails, it became evident that the quality and comprehensiveness of Arabic stop word lists play a pivotal role. An in-depth study of the existing criteria used for curating Arabic stop word lists was undertaken with that in mind. The analysis extended to a detailed comparison of these criteria with other available resources in the domain. The exhaustive nature of this review allowed us to gain valuable insights into the gaps and potential improvements in the realm of Arabic stop words. It was this comprehensive understanding that led us to identify 106 distinctive terms that were absent or overlooked in previous compilations. Given their lack of core meaning in the context of the study, these terms were apt candidates for classification as stop words. What we now refer to as 'Salloum's list' was curated by amalgamating these terms. In the broader spectrum of ML, especially when it comes to text classification, reducing the dimensionality of the feature space is crucial. By doing so, the speed, accuracy, and overall performance of ML classifiers can be enhanced. The "Salloum's list" has been designed with this principle in mind. Its incorporation aims to trim unnecessary information, thereby potentially reducing the dimensionality of the feature space. In the specific application, which is the detection of phishing emails in Arabic, this reduction can be instrumental. By excluding these 106 terms, a more streamlined and focused analysis by ML classifiers is anticipated. As a next step, the primary objective is to quantitatively measure the list's influence on the efficacy of ML classifiers in accurately pinpointing Arabic phishing emails.

4.6 Feature extraction

Feature extraction is the process of transforming raw data into acceptable inputs (that is, features), which may be processed by an ML algorithm [309], [310]. In other words, the extracted features must reflect the primary textual material in a manner that most suits the requirements of the applied classifier algorithm. Minimal feature extraction is usually required, with the exception of DL neural networks, which can conduct feature extraction independently [311]. Furthermore, a weak classifier with relevant features is thought to outperform a robust classifier with low-quality

features. Bag-of-Words (BOW), DTM, TF-IDF, word embeddings and character-level convolutional networks are prominent feature extraction methods for TC. This study concurrently presents a phishing email detection model developed on the word level (DTM, TF-IDF and word embeddings) and character-level convolutional networks.

4.6.1 Word level

Word-level feature extraction is a critical component of phishing email detection. By analysing the textual content of an email, researchers can identify key features that distinguish between legitimate and malicious emails [312]. The use of word-level features enables the algorithm to focus on the specific language and phrasing used in the emails rather than solely relying on metadata or other contextual factors. This feature extraction method allows researchers to identify specific words and phrases commonly used in phishing emails, such as urgent calls to action or requests for sensitive information. By analysing these features, researchers can train ML algorithms to accurately detect and classify phishing emails, enabling organisations to protect themselves and their users from the potentially devastating effects of phishing attacks. The importance of word-level feature extraction in phishing email detection cannot be overstated, as it enables researchers to more effectively analyse and identify the specific characteristics of phishing emails, ultimately leading to more effective detection and prevention of these types of attacks. The present study uses three feature extraction methods based on word-level analysis – DTM, TF-IDF and word embeddings. These methods are widely used in NLP and text mining to extract meaningful information from textual data. DTM is a matrix representation of the frequency of terms in a document, whereas TF-IDF measures the relevance of a term in a document by comparing its frequency in that document with its frequency in the corpus as a whole. Word embeddings are a recent technique that involves representing words as high-dimensional vectors, capturing their semantic relationships and contextual meaning [313]. By utilising these three feature extraction methods based on word-level analysis, the present study aims to provide a comprehensive and comparative analysis of their effectiveness in detecting and classifying textual data, particularly in phishing email detection.

4.6.1.1 TF-IDF

The TF-IDF weight is employed in information retrieval to measure the value of a word to a document in a set of documents [314]. The relevance of a word increases in direct relation to the quantity of occurrence in the document (term frequency) and decreases in inverse relation to the word's document frequency in the set. IDF is a measure of the term's discriminating power. It calculates the frequency of a term across many documents. As a result, a word with a high term frequency in one document and a lower document frequency in the entire set of documents has a high TF-IDF weight. TF-IDF is a powerful technique that can provide valuable insights into the importance of terms in an email [11]. By analysing the frequency of each term in an email and comparing it to its occurrence in the entire corpus, TF-IDF assigns higher weights to terms specific to the email and less common across the corpus. This approach allows us to identify terms potentially indicative of phishing attempts or suspicious content [112]. However, it is important to note that TF-IDF is just one piece of the puzzle in phishing email detection. Combining it with other techniques, such as content-based analysis, header analysis, URL analysis and blacklisting, can enhance the effectiveness and accuracy of the detection system.

4.6.1.1.1 TF-IDF: Essential features for Arabic phishing email detection

When using TF-IDF for Arabic phishing email detection, several essential features can be derived from the TF-IDF representation of the emails. These features, in turn, can provide valuable insights into the characteristics of phishing emails in Arabic. The following section elucidates some salient attributes of this process.

4.6.1.1.1.1 Rare and important terms

TF-IDF assigns higher weights to terms that are rare in the corpus but frequently occur in a specific email. Identifying such terms can help capture specific language patterns and indicators of phishing attempts. In this section, an example of rare and important terms in the Arabic language, which can be identified using TF-IDF for phishing email detection, has been elucidated.

Consider the following phishing email example, as depicted in Figure 4.3:

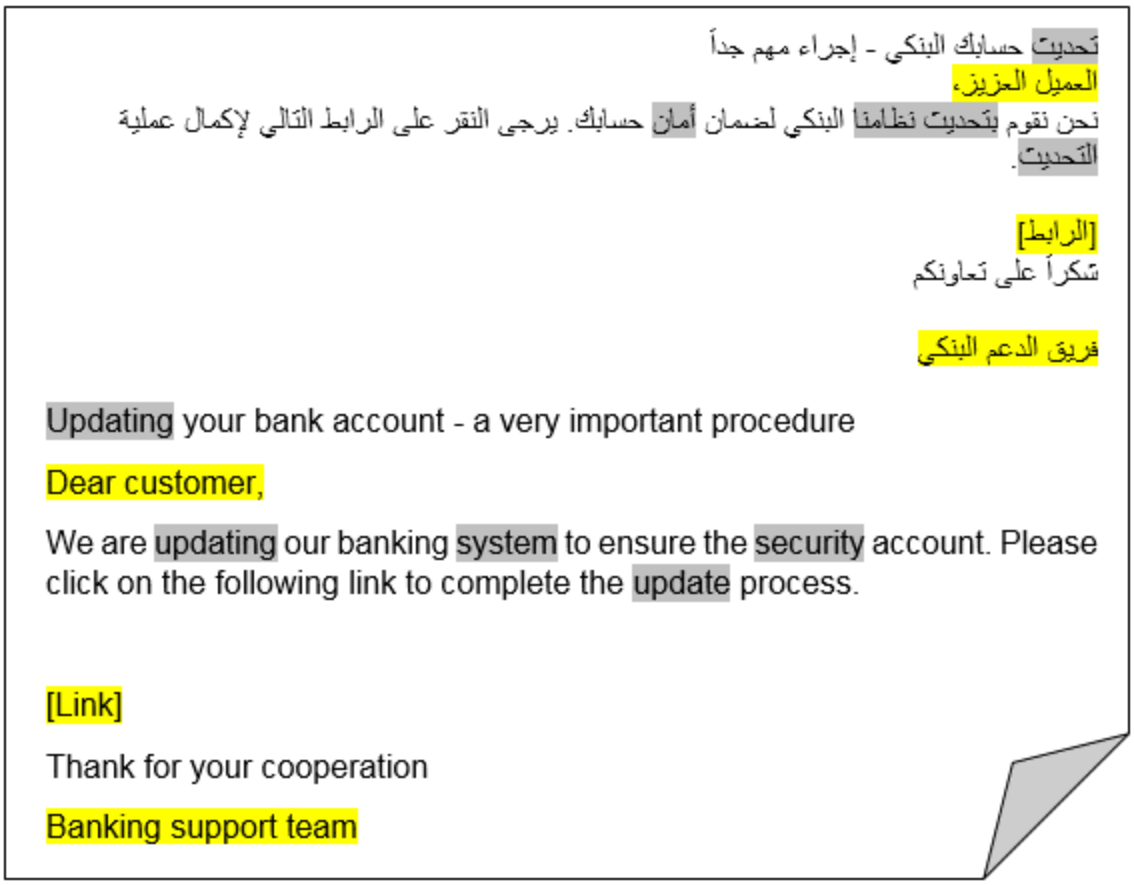


Figure 4.3 Phishing email example 1

Let us assume that a corpus of legitimate emails is available, and this email is flagged as a phishing attempt. The following terms might be found by applying TF-IDF and analyzing the rare and important terms:

A. Rare terms:

- تحديث (update)
- نظامنا (our system)
- أمان (security)

These terms are relatively less frequent in the corpus but appear frequently in the phishing email, thereby indicating their potential significance for phishing detection.

B. Important terms:

- العميل العزيز (dear customer)
- الرابط (link)
- فريق الدعم البنكي (bank support team)

These terms are important because they are commonly exploited in phishing emails to create a sense of familiarity, urgency and trustworthiness. Attackers often use personalised greetings, links and references to customer support teams to deceive recipients. By recognising the rare and important terms identified through TF-IDF analysis, a phishing email detection system can flag similar emails with similar language patterns and characteristics. This system assigns higher weights to these terms when analysing incoming emails, facilitating more accurate identification of potential phishing attempts.

4.6.1.1.1.2 Lexical variations

The Arabic language exhibits morphological variations and non-standard spellings, which attackers may utilise to deceive recipients. TF-IDF can help identify variations of common words and detect suspicious changes in spellings or word formations. This section provides an example of domain-specific terms in Arabic that can be identified using TF-IDF for phishing email detection.

Take into account an illustrative phishing email example that targets banking customers, as shown in Figure 4.4:

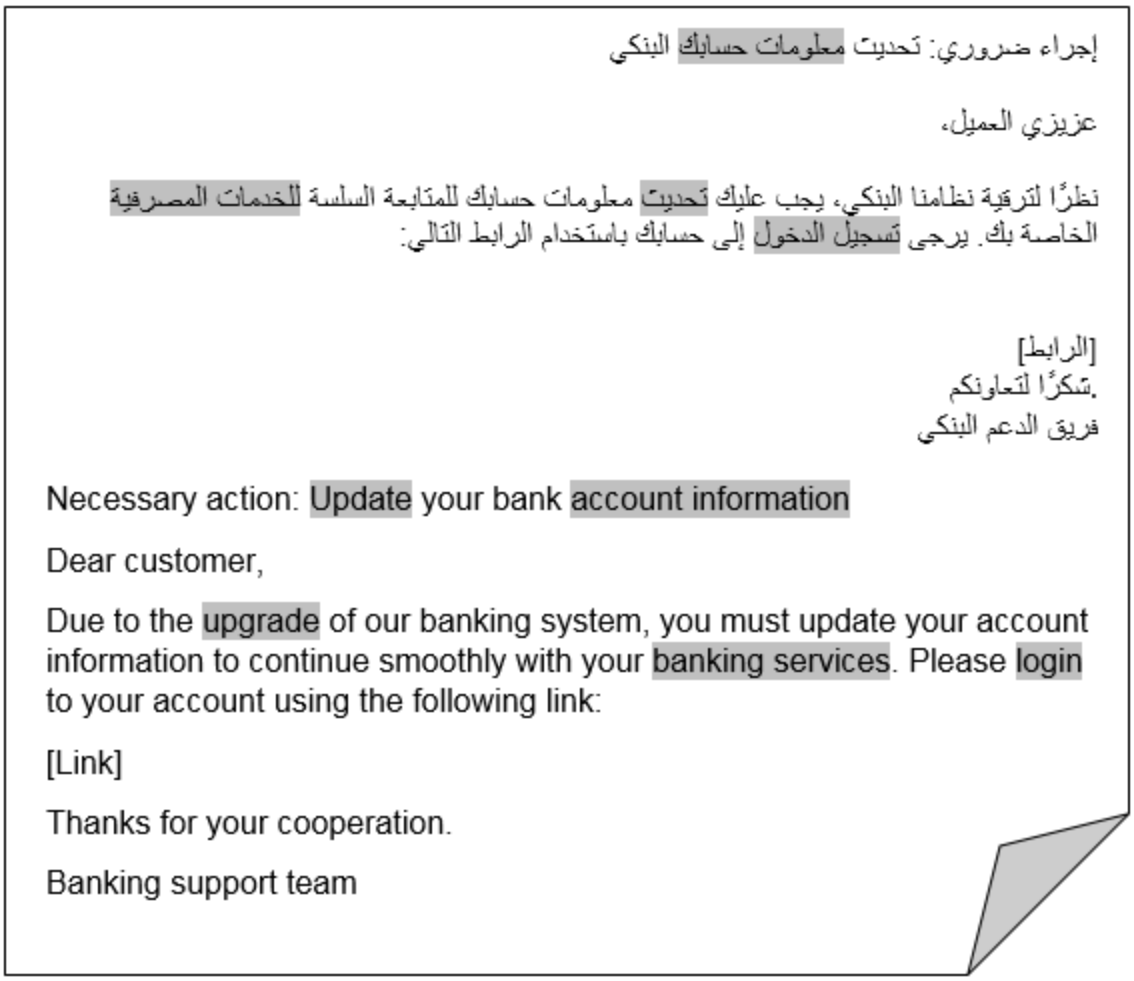


Figure 4.4 Phishing email example 2

In this example, the following domain-specific terms related to banking can be identified:

- معلومات حسابك (your account information)
- الخدمات المصرفية (banking services)
- تحديث (update)
- تسجيل الدخول (login)

These terms are specific to the banking domain and are commonly used in legitimate communications between banks and their customers. Attackers often exploit these domain-specific terms to create a false sense of urgency and persuade recipients to click on malicious links or provide sensitive information. By analysing the TF-IDF values of these domain-specific terms, a

phishing email detection system can identify similar emails that employ such terms. The system, in turn, assigns higher weights to these terms when evaluating incoming emails, facilitating better detection of phishing attempts targeting banking customers. In this context, it is important to note that attackers continually evolve their techniques so that domain-specific terms may change over time. Therefore, regular monitoring and updating of the detection system are essential to stay ahead of new phishing tactics.

4.6.1.1.1.3 Unusual linguistic patterns

Phishing emails may employ unusual linguistic patterns or syntactic structures that deviate from normal communication. TF-IDF can highlight uncommon terms or phrases in legitimate emails, thus indicating potential phishing attempts. Figure 4.5 illustrates an example of unusual linguistic patterns in Arabic, which can be identified using TF-IDF for phishing email detection.

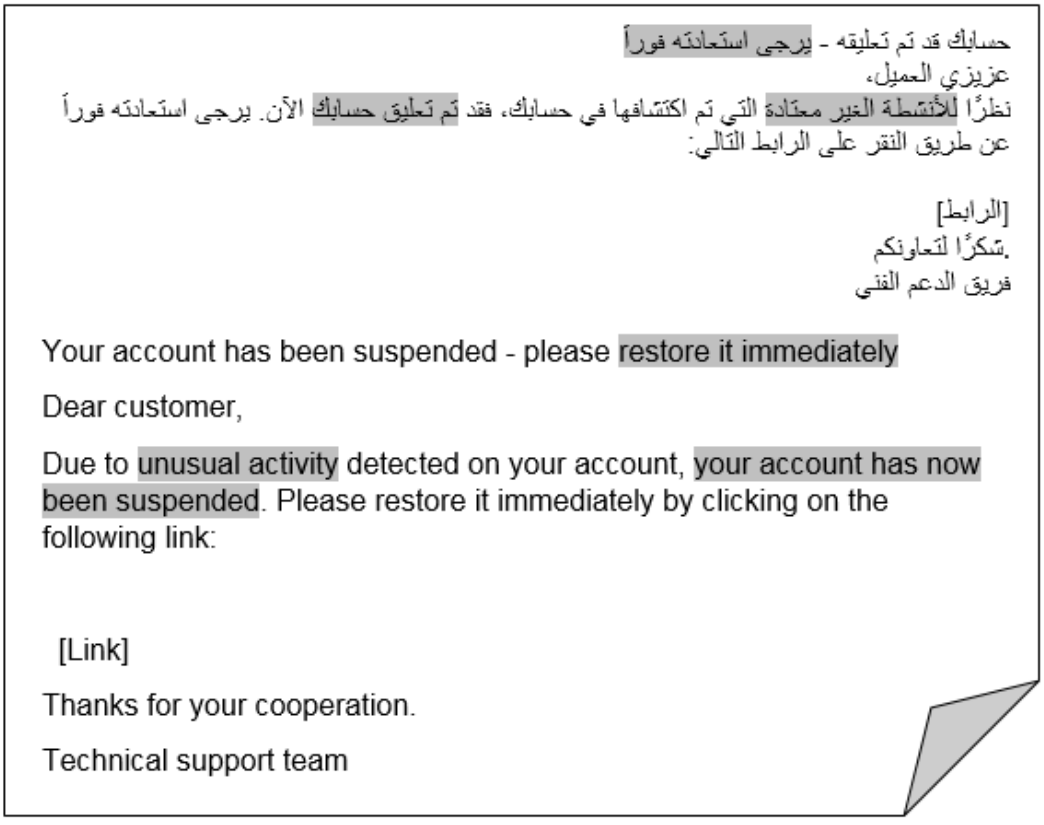


Figure 4.5 Phishing email example 3

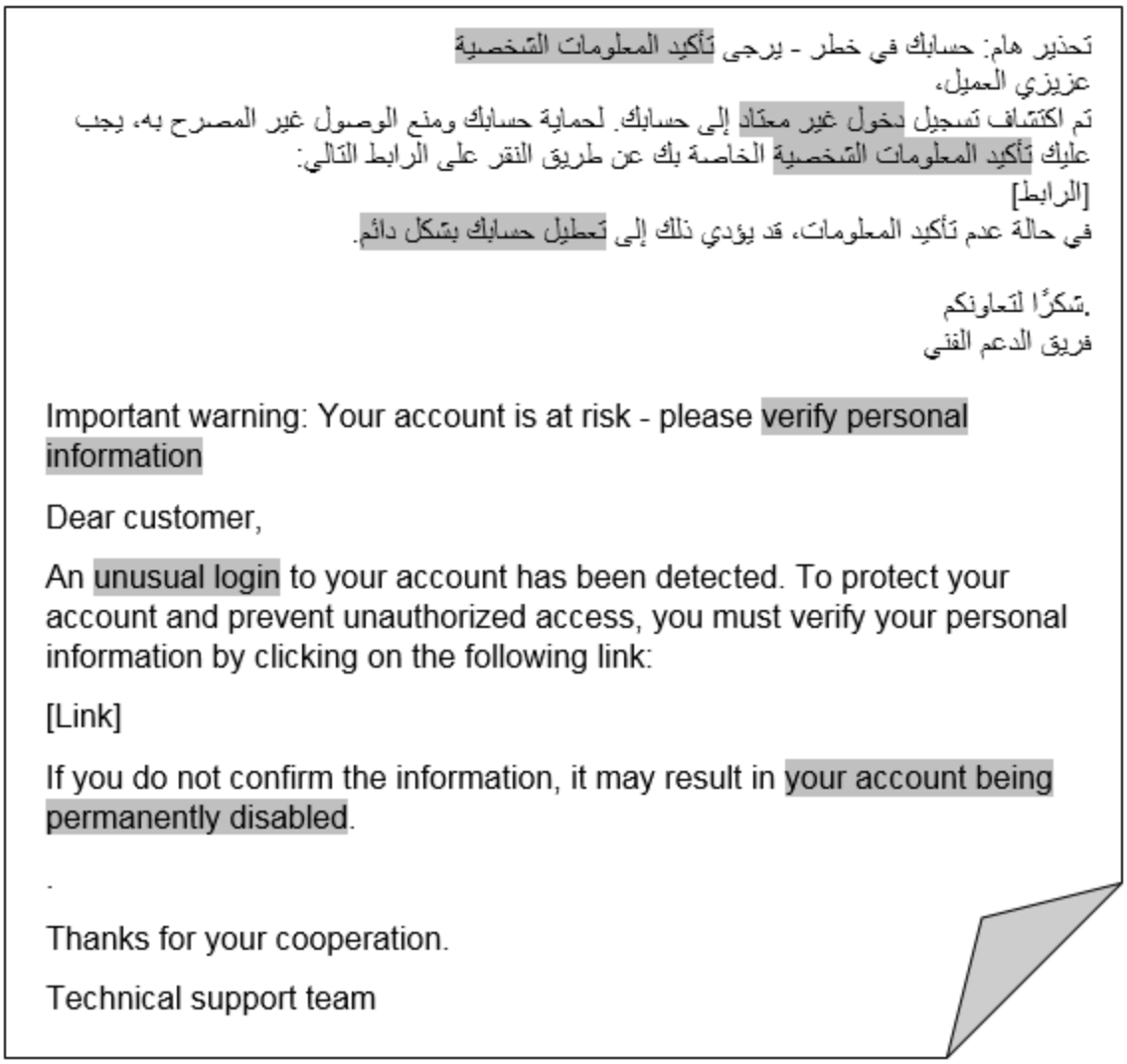
In this example, the following unusual linguistic patterns can be identified:

- استعادته فوراً (recover it immediately): The use of the imperative form and the urgency expressed through "فوراً" (immediately) create a sense of urgency and pressurise the recipient into taking immediate action.
- الأنشطة الغير معتادة (unusual activities): The phrase "الأنشطة الغير معتادة" (unusual activities) suggests that there have been suspicious or unauthorised activities in the account, which is a common tactic used in phishing emails to prompt the recipient to take action.
- تم تعليق حسابك (your account has been suspended): The use of the passive voice and the term "تعليق حسابك" (your account has been suspended) adds a sense of severity and urgency, aiming to create fear and prompt the recipient to click on the provided link.

Analysing the TF-IDF values of these unusual linguistic patterns, a phishing email detection system can flag similar emails exhibiting such language patterns. The system, in turn, assigns higher weights to these patterns, indicating a higher likelihood of the email being a phishing attempt. It is important to note that attackers continuously adapt their tactics, so the identification of unusual linguistic patterns should be regularly updated based on emerging trends in phishing attacks.

4.6.1.1.1.4 Social engineering cues

Phishing attacks often rely on social engineering techniques to manipulate recipients. TF-IDF can capture language patterns associated with persuasion, urgency, requests for personal information or threats commonly used in phishing emails. Figure 4.6 shows an example of social engineering cues in the Arabic language that can be identified using TF-IDF for phishing email detection.


تحذير هام: حسابك في خطر - يرجى تأكيد المعلومات الشخصية
عزيزي العميل،
تم اكتشاف تسجيل دخول غير معتاد إلى حسابك. لحماية حسابك ومنع الوصول غير المصرح به، يجب عليك تأكيد المعلومات الشخصية الخاصة بك عن طريق النقر على الرابط التالي:
[الرابط]
في حالة عدم تأكيد المعلومات، قد يؤدي ذلك إلى تعطيل حسابك بشكل دائم.

.شكرًا لتعاونكم
فريق الدعم الفني

Important warning: Your account is at risk - please verify personal information

Dear customer,

An unusual login to your account has been detected. To protect your account and prevent unauthorized access, you must verify your personal information by clicking on the following link:

[Link]

If you do not confirm the information, it may result in your account being permanently disabled.

.

Thanks for your cooperation.

Technical support team

Figure 4.6 Phishing email example 4

In this example, the following social engineering cues can be identified:

- تسجيل دخول غير معتاد (unusual login): The mention of an unusual login is designed to create a sense of concern and urgency in the recipient, prompting them to take immediate action.
- تأكيد المعلومات الشخصية (confirm/verify personal information): The request to confirm personal information is a classic social engineering tactic used in phishing attacks. Attackers exploit the recipient's fear of unauthorised access to their account and trick them into providing sensitive information.

- تعطيل حسابك بشكل دائم (permanent account suspension): The threat of permanent account suspension aims to create a sense of fear and urgency in the recipient, compelling them to click on the provided link and take the desired action.

Analysing the TF-IDF values of these social engineering cues, a phishing email detection system can identify similar emails exhibiting such language patterns. The system, in turn, assigns higher weights to these cues, indicating a higher likelihood of the email being a phishing attempt. As exhibited in the example provided in the previous section, it is important to note that attackers constantly refine their social engineering techniques, so the identification of social engineering cues should be regularly updated to stay ahead of evolving phishing tactics.

4.6.1.1.1.5 Cross-domain terms

Some phishing emails target specific industries or sectors. By examining the TF-IDF values of terms across different domains, one can identify terms that are significantly more frequent or important in phishing emails targeting a specific sector. Figure 4.7 displays an instance of cross-domain terms in Arabic, which are identifiable using TF-IDF for detecting phishing emails.

Figure 4.7 Phishing email example 5

In this example, cross-domain terms that can be indicative of phishing attempts targeting email services can be identified:

- تحديث معلومات حسابك (update your account information): This term refers to updating account information and can be relevant to various online services, including email services.
- خدمة البريد الإلكتروني (email service): This term specifically refers to email services, and attackers may exploit it to target users of various email platforms.

Analysing the TF-IDF values of these cross-domain terms, a phishing email detection system can identify similar emails that employ such terms across different domains. The system assigns higher weights to these terms when evaluating incoming emails, indicating a higher likelihood of the email being a phishing attempt targeting email services. It is important to note that attackers

continuously adapt their techniques, so the identification of cross-domain terms should be regularly updated based on emerging trends in phishing attacks and the targeted domains.

4.6.1.1.1.7 Contextual keywords

Take the context of certain terms indicative of phishing attempts into consideration. For example, terms such as "account verification", "password reset" or "urgent action required" can be strong indicators when combined with other features and context. Figure 4.8 illustrates how contextual keywords in Arabic can be pinpointed using TF-IDF for phishing email detection.

Figure 4.8 Phishing email example 6

In this example, the following contextual keywords can be identified:

- تحديث الرقم السري (password update): The phrase "تحديث الرقم السري" (password update) indicates a request to update the password. Attackers often exploit this context to deceive recipients into revealing their account credentials.
- اختراق (breach): The mention of multiple breach attempts adds a sense of urgency and concern, aiming to prompt the recipient to take immediate action.
- الموقع الرسمي للبنك (official bank website): The inclusion of this phrase emphasises the need to visit the official bank website, creating an illusion of legitimacy. Attackers may use deceptive links to redirect users to phishing websites resembling the bank's official site.

Analysing the TF-IDF values of these contextual keywords, a phishing email detection system can identify similar emails that employ such language patterns in a specific context. The system assigns higher weights to these keywords when evaluating incoming emails, indicating a higher likelihood of the email being a phishing attempt. It is crucial to note that attackers continuously adapt their strategies, so the identification of contextual keywords should be regularly updated based on emerging trends and evolving phishing techniques.

4.6.1.1.2 Feature extraction for Arabic phishing emails using TF-IDF

The TF-IDF weights were employed as clustering features, as described in [315]. The parameters and terminology outlined in this section are used to construct these weights. Let us assume we extract features from a data set **E** made up of |**E**| emails. Let N (*w; m*) be the quantity *w*, which appears in *m* for a word *w* and an email m. Assume you are looking at a set T = {*t1…tk*} of terms *t1...., tk*. TF (*w*, *m*) denotes the repetends of a word ***w € T*** in an email *m* and is described as the quantity ***w*** appears in *m*, normalised over the number of repetends of all words in ***m***:

$$\text{TF}(w, m) = \frac{N(w,m)}{\sum_{i=1}^{k} N(ti,m)} \tag{4.2}$$

DF(*w*) stands for the document frequency of the word *w*, which is described as the proportion of emails in a data set in which the word *w* appears at a minimum once. The inverse document

frequency is employed to determine the importance of every term. The IDF (*w*) symbol is associated with it, and the following formula determines it:

$$\text{IDF}(w) = \log\left(\frac{|E|}{DF(w)}\right) \quad (4.3)$$

The TF-IDF weight of *w* in *m*, or TF-IDF of a word *w* in email *m*, is specified as follows:

$$\text{TF-IDF}(w, m) = \text{TF}(w, m) \times \text{IDF}(w, m) \quad (4.4)$$

A list of words with the maximum TF-IDF values across the entire dataset of emails was compiled. The TF-IDF values of these words in each email were then calculated. These weights and other features were compiled into a vector. The TF-IDF values were calculated using Gensim, a Python and NumPy package for vector space modeling of text documents.

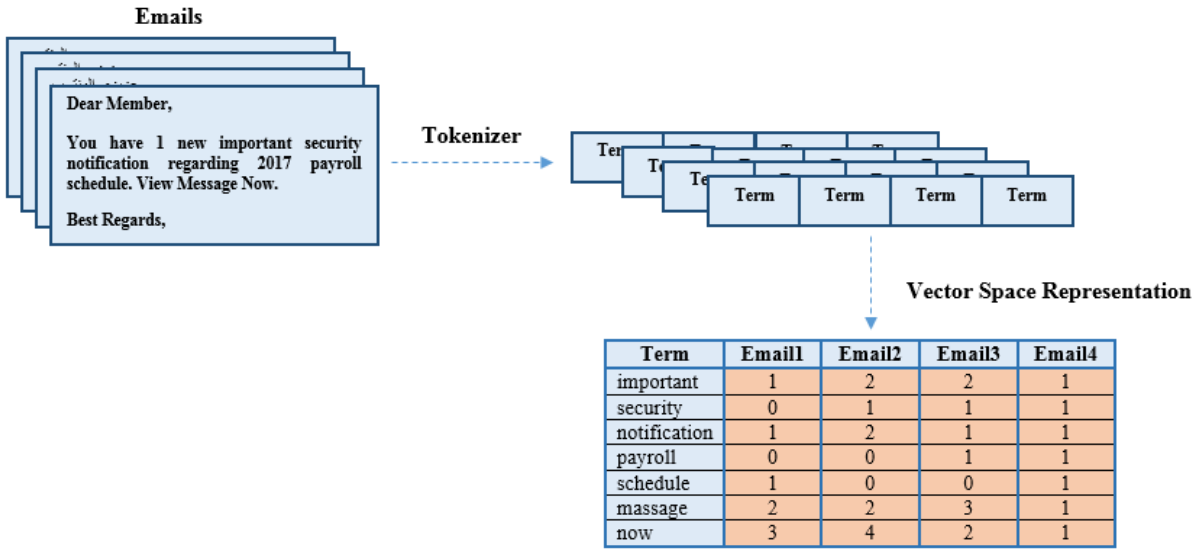
4.6.1.2 DTM

DTM is a mathematical matrix that delineates the frequency of terms that transpire in a set of documents, forming a systematic representation of the textual information [316]. The rows in the table represent the documents in the collection, whereas the columns represent the words. Each cell in the table contains the frequency of the corresponding word in the corresponding document [316]. DTM is a commonly used feature extraction method in NLP and text mining, as it enables researchers to analyse and compare the frequency of different words in different documents. DTM is a useful tool for identifying patterns and trends in large collections of textual data, and it is often used in ML algorithms to classify and categorise text data based on the frequency of different words. DTM is a simple and efficient method for representing textual data in a numerical format, making it easier to analyse and process large volumes of textual data [317].

A dataset of 1000 emails, encompassing both legitimate and phishing emails, can be used to create a DTM in which each row corresponds to an email in the dataset, and each column corresponds to a unique word that is found in any of the emails within the dataset. The cells in the table represent the frequency of each word in each email (see Figure 4.9). For example, if the word "password"

appears ten times in the first email in the dataset, and five times in the second email, then the corresponding cells in the first and second rows of the "password" column would be 10 and 5, respectively.

The frequency of words commonly associated with phishing emails, such as "urgent," "verify," or "password," can be calculated using this DTM. These words are often used in phishing emails to create a sense of urgency or convince recipients to provide sensitive information. By analysing the DTM and identifying words commonly associated with phishing emails, ML algorithms can be trained to accurately classify new emails as legitimate or phishing based on their content using this approach. In this context, DTM is an effective way to extract meaningful features from textual data and improve the accuracy of phishing email detection algorithms.



Term	Email1	Email2	Email3	Email4
important	1	2	2	1
security	0	1	1	1
notification	1	2	1	1
payroll	0	0	1	1
schedule	1	0	0	1
massage	2	2	3	1
now	3	4	2	1

Figure 4.9 DTM representation

When using a DTM for Arabic phishing email detection, several essential features can be derived from the DTM representation of the emails. These features can provide valuable insights into the characteristics of phishing emails in Arabic. In the following section, a few examples have been outlined.

4.6.1.2.1 Term Frequencies

Calculate the frequency of each term in the DTM. This feature provides information about the occurrence of specific words or phrases in phishing emails and can help identify commonly used deceptive language. Figure 4.10 presents a sample of term frequencies for detecting Arabic phishing emails.


تحذير هام: حسابك في خطر - يرجى تأكيد المعلومات الشخصية

عزيزي العميل،
تم اكتشاف تسجيل دخول غير معتاد إلى حسابك. لحماية حسابك ومنع الوصول غير المصرح به، يجب عليك تأكيد المعلومات الشخصية الخاصة بك عن طريق النقر على الرابط التالي:

[الرابط]

في حالة عدم تأكيد المعلومات، قد يؤدي ذلك إلى تعطيل حسابك بشكل دائم

شكرًا لتعاونكم
فريق الدعم الفني

Important warning: Your account is at risk - please verify personal information

Dear customer,

An unusual login to your account has been detected. To protect your account and prevent unauthorized access, you must verify your personal information by clicking on the following link:

[Link]

If you do not confirm the information, it may result in your account being permanently disabled

Thanks for your cooperation

Technical support team Bank security team

Figure 4.10 Phishing email example 7

In this example, the term frequencies of the words in the email can be analyzed. Let us consider some key terms and their frequencies:

Term Frequency: تحذير (Warning): 1, هام (Important): 1, حسابك (Your account): 2, يرجى (Please): 1, تأكيد (Confirm): 2, المعلومات (Information): 2, الشخصية (Personal): 1, الخاصة (Your): 1, النقر (Click): 1, الرابط (Link): 1, تعطيل (Disable): 1, دائم (Permanent): 1, شكرًا (Thank you): 1, تعاونكم (Your cooperation): 1, فريق (Team): 1, الدعم (Support): 1, الفني (Technical): 1.

By analysing the term frequencies, it can be observed that some terms occur more frequently than others. For example, "حسابك" (Your account) and "تأكيد" (Confirm) appear twice, indicating their importance in the email content. On the other hand, terms such as "تحذير" (Warning) and "هام" (Important) appear only once, emphasising a sense of urgency.

Phishing detection systems can leverage term frequencies to identify patterns or clusters of frequently occurring terms indicative of phishing emails. By comparing the term frequencies of incoming emails with known phishing email patterns, these systems can assess the likelihood of an email being a phishing attempt and take appropriate actions to protect users.

4.6.1.2.2 TF-IDF scores

The feature of assigning TF-IDF scores to the terms in the DTM highlights important terms specific to phishing emails and have higher discriminatory power. Terms with high TF-IDF scores appear frequently in phishing emails but less frequently in legitimate emails.

Figure 4.11 depicts a sample of TF-IDF scores for detecting phishing emails in Arabic.

Figure 4.11 Phishing email example 8

In this example, the TF-IDF scores for the words in the email can be calculated. Let us consider some key terms and their TF-IDF scores:

Term TF-IDF Score: استعادة (Account Recovery): 0, حسابك (Your Account): 0.346, تحديث (Update): 0.231, معلومات (Information): 0.231, الدخول (Log in): 0.231, يرجى (Please): 0.115, نشاط (Activity): 0, غير معتاد (Unusual): 0, الوصول (Access): 0, غير المصرح به (Unauthorised): 0, الخاصة (Your): 0.231, على الفور (Immediately): 0, الرابط (Link): 0.231, الإزعاج (Inconvenience): 0, ضروري (Necessary): 0, سلامة (Safety): 0, تعاونك (Your Cooperation): 0, فريق (Team): 0, الأمان (Security): 0, الفني (Technical): 0.

By calculating the TF-IDF scores, the importance of certain terms in the context of the emails can be observed. For example, "حسابك" (Your Account) has a higher TF-IDF score of 0.346, indicating

its significance in distinguishing phishing emails. On the other hand, terms such as "استعادة" (Account Recovery) and "نشاط" (Activity) have TF-IDF scores of 0, suggesting they are less informative or less discriminative in this context.

Phishing detection systems can utilise TF-IDF scores to identify significant terms specific to phishing emails in Arabic. By comparing the TF-IDF scores of terms in incoming emails with those of known phishing email templates, these systems can determine the likelihood of an email being a phishing attempt and take appropriate measures to safeguard users.

4.6.1.2.3 Rare terms

Identify rare terms in the DTM that infrequently occur across the corpus of emails. These terms might indicate specific linguistic patterns or domain-specific language used in phishing attempts. Figure 4.12 illustrates a sample of uncommon terms used in detecting phishing emails in Arabic.

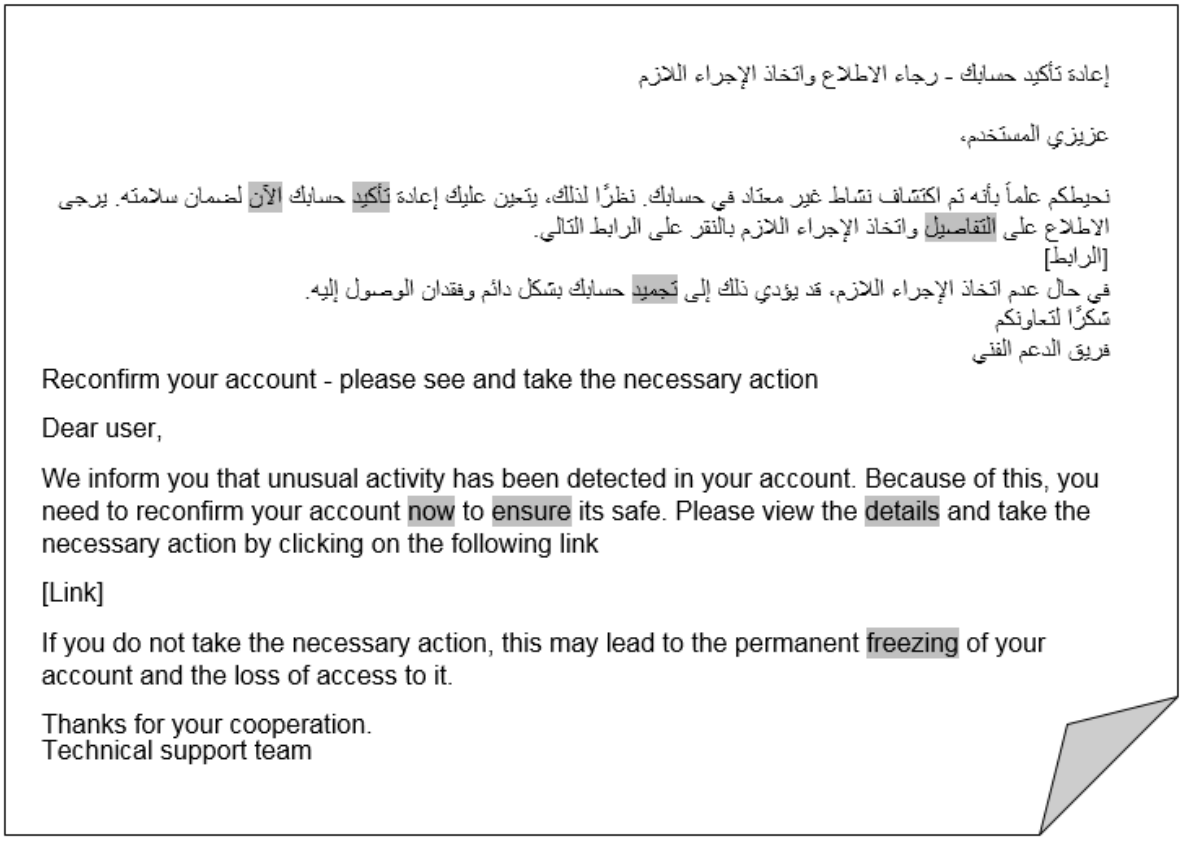


Figure 4.12 Phishing email example 9

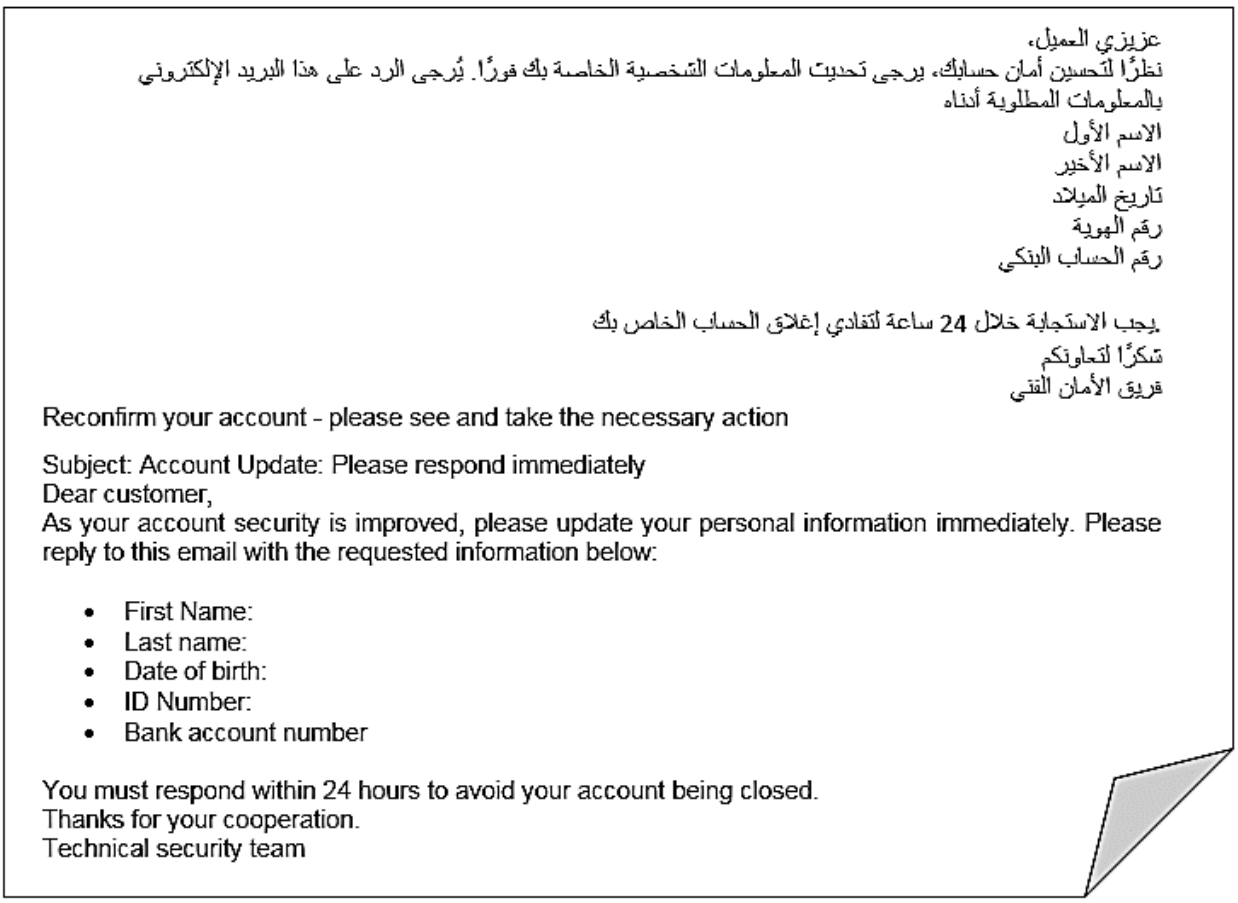
In this example, let us identify some rare terms that occur infrequently across the corpus of emails:

- تجميد (Freeze)
- ضمان (Ensure)
- الآن (Now)
- التفاصيل (Details)

These terms occur less frequently compared to more common terms such as "حسابك" (Your Account), "نشاط" (Activity) and "رابط" (Link) found in phishing emails. Rare terms can indicate specific linguistic patterns or domain-specific language used in phishing attempts. Phishing detection systems can consider these rare terms to identify unusual language usage or specific vocabulary associated with phishing emails. By comparing rare terms in incoming emails with known patterns of phishing emails, these systems can assess the likelihood of an email being a phishing attempt and take appropriate actions to protect users.

4.6.1.2.4 High-variance terms

Terms with high variance in the DTM must be identified. This is because these terms have varying frequencies across different emails and can capture distinctive language patterns or vocabulary associated with phishing emails. Figure 4.13 displays a sample of terms with high variance in detecting Arabic phishing emails.


عزيزي العميل،
نظرًا لتحسين أمان حسابك، يرجى تحديث المعلومات الشخصية الخاصة بك فورًا. يُرجى الرد على هذا البريد الإلكتروني بالمعلومات المطلوبة أدناه
الاسم الأول
الاسم الأخير
تاريخ الميلاد
رقم الهوية
رقم الحساب البنكي

يجب الاستجابة خلال 24 ساعة لتفادي إغلاق الحساب الخاص بك
شكرًا لتعاونكم
فريق الأمان الفني

Reconfirm your account - please see and take the necessary action

Subject: Account Update: Please respond immediately
Dear customer,
As your account security is improved, please update your personal information immediately. Please reply to this email with the requested information below:

- First Name:
- Last name:
- Date of birth:
- ID Number:
- Bank account number

You must respond within 24 hours to avoid your account being closed.
Thanks for your cooperation.
Technical security team

Figure 4.13 Phishing email example 10

In this example, let us identify some high-variance terms that have varying frequencies across different emails:

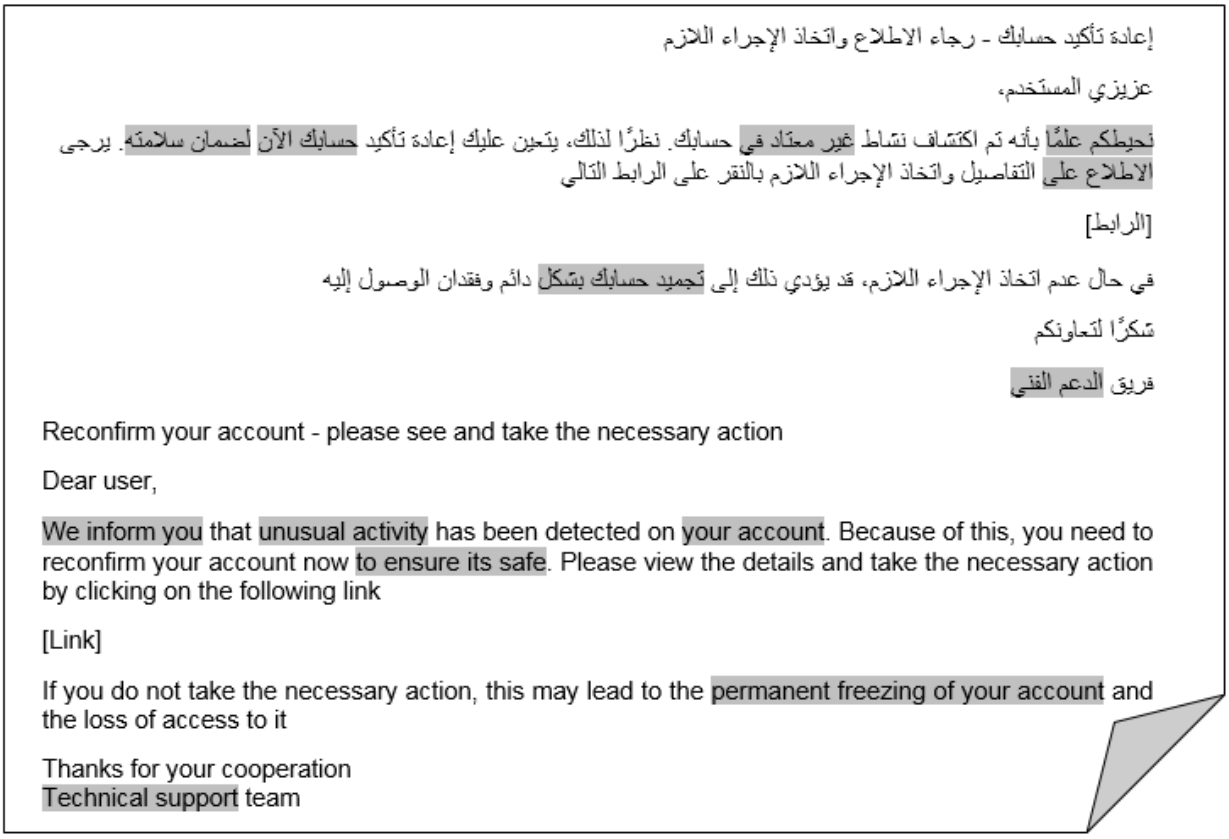
- الاسم الأول (First Name)
- الاسم الأخير (Last Name)
- تاريخ الميلاد (Date of Birth)
- رقم الهوية (ID Number)
- رقم الحساب البنكي (Bank Account Number)

These terms are likely to have high variance because they are specific to individual users and their personal information. Each phishing email may target different personal details, resulting in

varying frequencies of these terms across different emails. Phishing detection systems can consider these high-variance terms to identify emails that request sensitive personal information. By comparing the presence and frequency of these terms in incoming emails with known patterns of phishing emails, these systems can assess the likelihood of an email being a phishing attempt and take appropriate actions to protect users from disclosing their personal information.

4.6.1.2.5 N-grams

N-grams (sequences of N words) in the DTM capture contextual information and preserve the order of words. This feature can identify specific phrases or patterns commonly used in phishing emails in the Arabic language. Figure 4.14 presents a sample of N-grams for detecting phishing emails in Arabic.


إعادة تأكيد حسابك - رجاء الاطلاع واتخاذ الإجراء اللازم

عزيزي المستخدم،

نحيطكم علمًا بأنه تم اكتشاف نشاط غير معتاد في حسابك. نظرًا لذلك، يتعين عليك إعادة تأكيد حسابك الآن لضمان سلامته. يرجى الاطلاع على التفاصيل واتخاذ الإجراء اللازم بالنقر على الرابط التالي

[الرابط]

في حال عدم اتخاذ الإجراء اللازم، قد يؤدي ذلك إلى تجميد حسابك بشكل دائم وفقدان الوصول إليه

شكرًا لتعاونكم

فريق الدعم الفني

Reconfirm your account - please see and take the necessary action

Dear user,

We inform you that unusual activity has been detected on your account. Because of this, you need to reconfirm your account now to ensure its safe. Please view the details and take the necessary action by clicking on the following link

[Link]

If you do not take the necessary action, this may lead to the permanent freezing of your account and the loss of access to it

Thanks for your cooperation
Technical support team

Figure 4.14 Phishing email example 11

In this example, let us consider trigrams (three grams) to capture contextual information and preserve the order of words. Here are some trigrams from the email:

- نحيطكم علمًا (We inform you)
- غير معتاد في (Unusual activity in)
- حسابك الآن (your account now)
- لضمان سلامته (to ensure its safety)
- الاطلاع على (to review)
- تجميد حسابك بشكل (freeze your account permanently)
- الدعم الفني (technical support)

By taking trigrams into consideration, combinations of three words that provide more context and meaning than individual words can be captured. These trigrams can identify specific phrases or patterns commonly used in phishing emails in the Arabic language. Phishing detection systems can utilise N-grams, such as trigrams, to identify suspicious language patterns or common phrases used in phishing emails. By comparing the presence and frequency of N-grams in incoming emails with known patterns of phishing emails, these systems can assess the likelihood of an email being a phishing attempt and take appropriate actions to protect users.

By extracting these essential features from the DTM, ML or statistical techniques can be applied to train models for Arabic phishing email detection. These models, in turn, can classify incoming emails as either legitimate or phishing based on the presence or absence of these features.

4.6.1.3 Word embeddings

Conventional features, including TF-IDF, are less efficient in NLP implementations than word embeddings [318]. As a result, by transforming words into real-valued vectors, word embeddings were proposed for text representation. Vectorisation is performed after training neural networks on a text corpus. As discrete atomic symbols, words necessitate a continuous space projection in which semantically similar words are expressed by identical and homogenous vectors. Word embeddings retain the semantic meaning and syntax of words in documents dependent on their

content. Therefore, word embeddings were used with pre-trained neural network models in numerous NLP applications, including MT, speech recognition and TC [47].

Mikolov et al. [319] presented a prediction-based model, word2vec, for learning dense vector representations of distinct words from a huge unlabelled corpus. It is a shallow neural network model that learns how to map words to vector space points. Word2vec provides the following two models: Continuous Bag of Words (CBOW) and Skip-Gram (SG) and sophisticated optimisation approaches such as hierarchical Softmax and negative sampling. CBOW learns vector representations by anticipating the centre word provided by all the context words, whereas SG learns vector representations by predicting every context word individually depending on the centre word, as shown in Figure 4.15. The embedding dimension, namely the dimension of the word vectors and the size of the context frame, which symbolises the number of words that should be selected initially and following the centre word as context for training the word vectors, are the two pivotal parameters for training CBOW or SG embeddings.

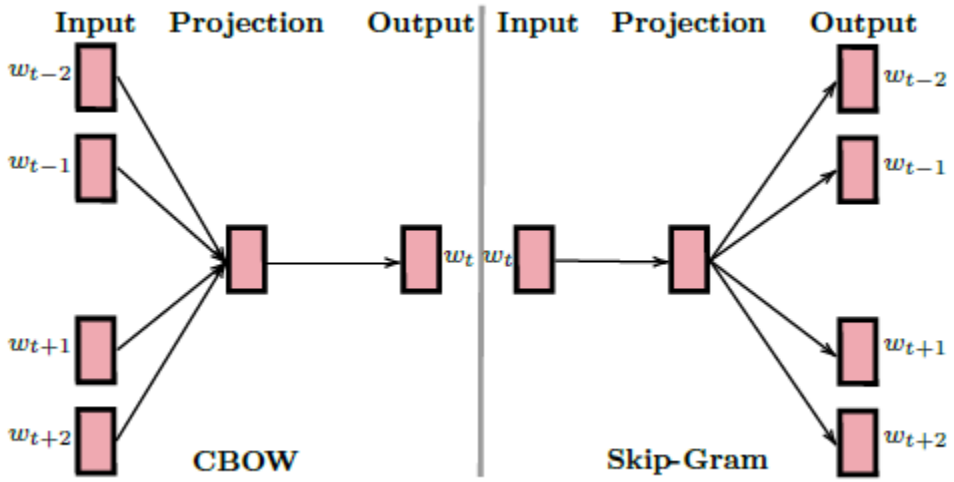


Figure 4.15 CBOW and SG training model illustrations [319]

The CBOW model uses a frame of nearby source-context words to predict the present target word. By predicting the source context words from the target words, the SG model, on the other hand, attributes more weight to the adjacent context words. Herein, word vectors, which have been pre-trained on datasets from a variety of concepts and were developed using unsupervised learning on a huge text corpus, are accessible. Words are considered atomic entities in embedding models such as CBOW and SG, which allocate dense vector representations to them [320]. Nevertheless, they cannot manage the Out Of Vocabulary (OOV) predicament because they neglect underlying sub-word data (sequences of neighbouring characters), which are critical in languages with a wide and diverse vocabulary.

4.6.1.3.1 TC model based on FastText

In 2016, Bojanowski et al. [321] presented FastText – a word2vec advancement. Every word in the vocabulary is regarded as a bag of character n-grams in this model, with the character embeddings formed being combined to construct the word's vector representation. As a corollary, the FastText model can generate vectors of OOV words and express morphological and lexical similarity of words. The advantage of this technique is that it can discover vector representations for words not directly available in the dictionary, as the technique preserves the word vectors as n-grams of characters.

FastText, a toolkit developed by the Facebook research team, helps people learn word representations and TC more efficiently [321]. The FastText embedding model algorithm's primary contribution is acknowledging the underlying pattern of words parallel to learning word representations, which is especially useful for morphologically rich languages such as Arabic [322].

Several challenges need to be addressed in the context of Arabic phishing email detection. Herein, one of the main problems is the unique linguistic characteristics of the Arabic language [323], including its rich morphology [324], unique script [325] and dialectal variations [326]. These linguistic complexities make it challenging to develop effective detection techniques that accurately identify phishing emails in Arabic. The lack of comprehensive datasets and research on Arabic phishing further exacerbates the problem. While phishing detection techniques have been

developed for other languages, their effectiveness in Arabic is limited due to language-specific challenges.

The FastText technique suitably addresses the problem of Arabic phishing email detection due to its ability to handle subword-level representations [327] and morphological variations [328]. FastText embeddings capture semantic and syntactic information by representing words as bags of character n-grams [329], [330], which is particularly useful in languages with complex morphology, such as Arabic [331]. This approach enables FastText to capture context-aware word representations, making it well-suited for capturing the nuances of Arabic email text. Moreover, FastText has shown promising results in various NLP tasks, including TC [332], [333] and sentiment analysis [334]–[336], which further supports its effectiveness in phishing email detection.

To address the problem of Arabic phishing email detection, FastText can be used in a two-step process. First, pre-trained FastText embeddings [337] specifically designed for the Arabic language can encode the semantic and syntactic information of Arabic words in phishing emails. These embeddings capture the contextual meaning of words and handle morphological variations, enabling the model to understand the linguistic nuances of Arabic emails. Second, DL architectures, such as CNNs and RNNs [338]–[340], can be employed to leverage the FastText embeddings and learn to detect phishing indicators in the Arabic email text.

4.6.1.3.1.1 FastText embeddings: Essential features for Arabic phishing email detection

FastText offers a range of powerful features that contribute to its effectiveness in various NLP tasks, including phishing email detection. In this section, some key features and capabilities of FastText that make it a valuable tool for the proposed approach are discussed.

4.6.1.3.1.1.1 Subword-level representations

FastText can generate subword-level representations by considering word composition based on character n-grams [341]. This is particularly advantageous in languages with rich morphology, such as Arabic, where words can have multiple forms derived from the same root [342]. By

encoding subword information, FastText captures the semantic and syntactic relationships between words more effectively than traditional word-level representations [343]. This enables the model to understand the meaning of unseen or OOV words, enhancing the detection of phishing indicators expressed in different morphological variations.

FastText embeddings fix subword-level representations by considering character n-grams as subword units during training. This allows FastText to capture morphological information and generate vector representations for complete words and their subword units. Let us take the Arabic word "الاحتيا لية" (al-ihtiyaaliya), which means "fraud" in English, as an example. In FastText, this word would be broken down into character n-grams such as "ال" ,"يا " ,"تيا" ,"حتي" ,"احت" and the boundary symbols "<" and ">". During training, FastText learns the embeddings for the complete word "الاحتيا لية" and its subword units. This allows FastText to capture the morphological variations and relationships between subword units. For instance, if we encounter a related word such as "احتياجات" (ihtiyaajaat), which means "needs" in English, the subword units "احت" and "تيا" are shared between the two words. As a result, the embeddings for these shared subword units will have similar representations, capturing their morphological similarity (see Table 4.2).

FastText embeddings excel at handling subword-level representations because they encode the information of complete words and their constituent subword units. This makes them effective in capturing morphological variations, especially in languages such as Arabic with rich morphological structures. By leveraging subword-level representations, FastText embeddings can benefit various NLP tasks in Arabic, including word similarity, sentiment analysis and named entity recognition (NER), by capturing the morphological nuances and relationships between words and their subword units.

4.6.1.3.1.1.2 Morphological variations handling

The rich morphology of the Arabic language poses a significant challenge in phishing email detection [344]. FastText embeddings are known for their ability to handle morphological variations effectively. This is achieved through the use of subword information in the training process. In FastText, words are broken down into character n-grams (subword units) of variable length. For example, the word "running" can be decomposed into the character n-grams "run",

"uni", "nning" and the special boundary symbols "<" and ">". By considering these subword units, FastText captures morphological information and generates word representations sensitive to morphological variations. During training, the word embeddings are learned for complete words and the constituent subword units. This approach allows FastText to capture similarities and relationships between words with shared subword units, even with different morphological forms. As a result, morphologically related words tend to have similar vector representations in FastText embeddings. For example, the embeddings of "run", "running" and "ran" will be closer to each other compared to unrelated words.

This property of FastText embeddings enables them to handle morphological variations effectively, even for words not explicitly present in the training data. By leveraging the subword information, the embeddings can generalise well to unseen words and capture their morphological similarities to known words. This capability makes FastText embeddings particularly useful in tasks where morphological variations play a significant role, such as language modelling, part-of-speech tagging and sentiment analysis.

Here is an example of how FastText embeddings can handle morphological variations in Arabic. Let us take the word "تحديثات" (tahdeeyat) in Arabic, which means "updates" in English. With FastText embeddings, the handling of morphological variations of this word can be explored. FastText considers subword units, such as character n-grams, during training. For "تحديثات", the subword units could include "ات" "ثا," "يت," "دي," "حد," "تح," and the boundary symbols "<" and ">". The FastText model learns vector representations for the complete word and these subword units (see Table 4.2).

Now, let us consider a morphological variation of "تحديثات" such as "تحديث" (tahdeeth), which means "update" in English. Although "تحديث" has a different form and lacks the "ات" suffix, FastText can still generate meaningful embeddings for this variation. As the subword units "دي" "تح," and "ث" are shared between the two words, their embeddings will have similar representations, capturing their morphological relationship. FastText embeddings in Arabic can effectively handle morphological variations by leveraging the shared subword units. This ability allows the model to generalise well to unseen word forms and capture the underlying morphological similarities.

4.6.1.3.1.1.3 Misspellings handling

FastText embeddings can help address misspellings by capturing the similarity between words with similar subword units [345]. Let us consider the following examples in Arabic:

- **Misspelling**: "الرابت" instead of "الرابط" (al-rabet instead of al-rabt) - where the letter "ط" (ta) is mistakenly replaced with "ت" (ta) at the end of the word. FastText's subword-level representations can help overcome this misspelling. As "الرابط" and "الرابت" share the same subword units "ب" ",را" ",ال" and "ط" or "ت," their embeddings are likely to be similar. This similarity can be leveraged to identify and correct misspellings.
- **Misspelling word example 1**: "بق" instead of "بنك" (baq instead of bank) - where the letter "ن" (nun) is omitted in the word. Again, FastText embeddings can be useful in handling this misspelling. The subword units "ب" and "نك" in "بنك" and "بق" are shared, which means their embeddings exhibit similarity. By comparing the embeddings of these words, the model can potentially recognise the misspelling and suggest the correct form.
- **Misspelling word example 2**: "يرجي" instead of "يرجى" (yurji instead of yurja) - where the letter "ى" (alef maqsura) is mistakenly replaced with "ي" (ya). FastText embeddings can assist in handling this type of misspelling as well. As "يرجي" and "يرجى" share the subword units "ي" and "رجى," their embeddings are likely to be similar. The model can identify the misspelling and suggest the correct form by comparing these embeddings.

The ability of FastText to capture subword-level representations and their similarities allows it to handle misspellings effectively. By leveraging these representations, NLP models can detect and correct misspelt words by identifying similarities with correctly spelt words.

4.6.1.3.1.1.4 OOV word handling

One notable advantage of FastText for feature extraction is its capability to handle OOV words [346]. This feature is particularly crucial in the context of phishing email detection, especially in Arabic language settings. Arabic, known for its rich and complex vocabulary, often poses challenges in dealing with unfamiliar or rare words. However, FastText's ability to represent words as subword n-grams enables it to effectively capture the semantic and syntactic information even

for OOV words. By leveraging subword representations, FastText enhances the accuracy and robustness of phishing email detection models in Arabic, ensuring a more comprehensive and reliable defence against phishing attacks. Here are a few more examples of OOV words in Arabic and how FastText embeddings handle them:

- **OOV word example 1**: "سلامتك" (salamatik) - meaning "your health" in English. If "سلامتك" is an OOV word, FastText will treat it as an unseen word. In turn, the model will assign an OOV representation as it is not a part of the training data or the embedding vocabulary.
- **OOV word example 2**: "تجربتي" (tajribati) - meaning "my experience" in English. If "تجربتي" is an OOV word, FastText will handle it as an unseen word. The model will assign an OOV representation to it, as it is absent in the training data or the embedding vocabulary.
- **OOV word example 3**: "مستقبلنا" (mustaqbalna) - meaning "our future" in English. If "مستقبلنا" is an OOV word, FastText will treat it as unseen. The model will assign an OOV representation to it, as it is not a part of the training data or the embedding vocabulary.

If the words are OOV, FastText will assign them an OOV representation in all these cases. This representation is typically a special token or vector that signifies the absence of pre-learned embeddings in these words. It is important to note that the coverage of the FastText embeddings depends on the model and the training data used. OOV words are handled as unseen words and represented using a distinct OOV representation, which allows the model to handle them appropriately in downstream NLP tasks. These features collectively make FastText a valuable tool for the proposed approach, allowing us to effectively capture the linguistic intricacies and variations present in Arabic phishing emails. FastText enhances the phishing email detection model's accuracy and robustness in Arabic by leveraging the subword-level representations, morphological variation handling, language-specific embeddings and efficient training and inference.

Table 4.2 Key features of FastText for effective phishing email detection in Arabic

No.	English Text	Arabic Text	Issue
1	Subject: Update for Your Application I hope this email finds you well. I would like to share with you some exciting new updates regarding your application and its latest developments. We have gathered user feedback and worked diligently to enhance your experience. We are delighted to inform you that we have implemented several noteworthy updates to improve the application's performance. We have addressed numerous reported issues and problems identified in our recent analysis. This will contribute to making your experience smoother and more comfortable. Furthermore, we have added a fantastic new feature that allows you to customize the user interface according to your personal preferences. You can now choose the style and colors you prefer and adapt the application to suit your individual needs. We always value your feedback and suggestions. Therefore, if you have any inquiries or comments about the current updates or any new ideas to enhance your experience, please don't hesitate to share them with us. We welcome your input and care about your satisfaction. To update your application, click on the following link:	الموضوع: تحديث تطبيقك أتمنى أن تكون بخير. أود أن أشاركك بعض التحديثات الجديدة والمثيرة حول تحديثك ومستجداته. لقد قمنا بجمع ملاحظات المستخدمين وعملنا بجد لتحسين تجربتك. نحن فرحون بأن نخبرك أننا أضفنا العديد من التحديثات المميزة لتحسين أداء التطبيق. قمنا بإصلاح العديد من الأخطاء والمشاكل التي أبلغت عنها والتي تم تحديدها في تحليلنا الأخير. وهذا سيساعد في جعل تجربتك أكثر سلاسة ومريحة. لقد أضفنا أيضًا ميزة جديدة رائعة تمكنك من تخصيص واجهة المستخدم وفقًا لتفضيلاتك الشخصية. يمكنك الآن اختيار النمط والألوان التي تفضلها وتكييف التطبيق بما يناسب احتياجاتك الفردية. نحن نستمع دائمًا إلى ملاحظاتك واقتراحاتك. لذا، إذا كان لديك أي استفسارات أو تعليقات حول التحديثات الحالية أو أي فكرة جديدة لتحسين تجربتك، فلا تتردد في مشاركتها معنا. نحن نرحب برأيك ونهتم برضاك لتحديث تطبيقك، أضغط على الرابط التالي .	**Morphological variations** "تحديث" (update) "تحديثات" (updates) "التحديثات" (updates) "تحديثك" (your update) "لتحديث" (To update)
2	Invitation to Participate in the Awareness Conference on Combating E-mail Scams We are pleased to extend an invitation to you to participate in the E-mail Scam Conference, a distinguished event aimed at raising awareness and combating the growing threat of e-mail scams and phishing. As a recognized expert in this field, your contributions and expertise will greatly contribute to the progress of the conference. We believe that your valuable knowledge and experience will shed light on the latest trends,	دعوة للمشاركة في مؤتمر التوعية للتصدي للرسائل الاحتيالية يسعدنا تمديد الدعوة لكم للمشاركة في مؤتمر الرسائل الاحتيا لية، وهو حدث مميز يهدف إلى زيادة الوعي ومكافحة التهديد المتزايد للرسائل الاحتيا لية والتصيد الإلكتروني. بصفتكم خبيرًا معترفًا في هذا المجال، فإن إسهاماتكم وخبراتكم ستسهم بشكل كبير في سير المؤتمر. نعتقد أن معرفتكم القيمة وخبرتكم ستسلط الضوء على أحدث الاتجاهات والتقنيات والإجراءات الوقائية لحماية الأفراد والمؤسسات من المخاطر المرتبطة بالرسائل الاحتيا لية. سيتضمن المؤتمر متحدثين متميزين، وورش عمل تفاعلية، وجلسات نقاشية مشوقة.	**Subword-level representations** احتيالي fraudulent "احتيا" and "لية" **Misspellings** بنق bank (incorrect spelling) يرجي kindly (incorrect spelling) الرابت link (incorrect spelling)

	techniques, and preventive measures to protect individuals and organizations from the risks associated with e-mail scams. The conference will feature distinguished speakers, interactive workshops, and engaging panel discussions. To participate in the conference, kindly click on the following link to update your information and submit the registration fee through our account with Dubai Islamic Bank.	للمشاركة في المؤتمر يرجي الضغط على الرابت التالي لتحديث بياناتك و تسديد رسوم الاشتراك عبر حسابنا في بنق دبي الاسلامي.	
3	Dear, I hope this email finds you well. I would like to share some exciting updates regarding your health, my experience, and our future. We have collected user feedback and worked diligently to improve your experience and ensure our future well-being. We are proud to inform you that we have added new and exciting features to enhance your health and elevate your experience. Our team has been working on developing our products and services to better meet your needs. We highly value your personal experience and consider it a vital aspect. Therefore, if you have any questions or comments regarding your health, experience, or our future, please don't hesitate to share them with us. Your opinion matters to us, and we care about your satisfaction. Lastly, we thank you for your continuous support. We look forward to staying in touch and aim to provide you with high-quality service that meets your expectations. Best regards,	العزيزة/العزيز أتمنى أن تكون/تكوني بخير. أود أن أشاركك بعض التحديثات الجديدة والمثيرة حول سلامتك، تجربتي، ومستقبلنا. لقد قمنا بجمع ملاحظات المستخدمين وعملنا بجد لتحسين تجربتك/تجربتي وضمان رفاهيتك/رفاهيتنا المستقبلية. نحن فخورون بأن نعلمك/نعلمكي بأننا أضفنا ميزات جديدة مثيرة لتحسين سلامتك/سلامتكي وتعزيز تجربتك/تجربتي. لقد قام فريقنا بالعمل على تطوير منتجاتنا وخدماتنا لضمان تلبية احتياجاتك/احتياجاتكي بشكل أفضل. نحن نولي اهتمامًا كبيرًا لتجربتك/تجربتي الشخصية ونعتبرها أمرًا مهمًا جدًا. ولذا، إذا كان لديك أي استفسارات أو تعليقات بشأن سلامتك، تجربتك، أو مستقبلنا، فلا تتردد/تترديدي في مشاركتها معنا. نحن نرحب برأيك/رأيكي ونهتم برضاك/رضاكي. وفي الختام، نشكرك/نشكركي على دعمك/دعمكي المستمر لنا. نتطلع إلى تواصلك/تواصلكي الدائم ونأمل أن نستمر في تقديم خدمة عالية الجودة تلبي توقعاتك/توقعاتكي. مع خالص التحية،	**<u>Out-of-vocabulary words</u>** "سلامتك" (your health) "تجربتي" (my experience) "مستقبلنا" (our future) **<u>Morphological variations</u>** "تجربتك" (my experience) "تجربتي" (my experience) "لتجربتك" (for my experience) "رفاهيتك" (well-being) "رفاهيتنا" (well-being) "سلامتك" (your health) "سلامتكي" (your health) "احتياجاتك" (your needs) "احتياجاتكي" (your needs) "رأيكي " (your opinion) "برأيك" (your opinion) "برضاك" (your satisfaction) "رضاكي" (your satisfaction) "نشكرك" (we thank you) "نشكركي" (we thank you) "دعمك" (your continuous support) "دعمكي" (your continuous support) "تواصلك" (staying in touch) "تواصلكي" (staying in touch) "توقعاتك" (your expectations) "توقعاتكي" (your expectations)

4.6.1.3.1.2 The embedding matrix

Algorithm 4-1 depicts the processes of the embedding matrix, which analyses the dataset to identify the relevant array of features. FastText tokenizer is used to convert source documents to a FastText-based embedding matrix. Sentences are tokenised, padded, trimmed and converted into a sparse matrix where each row represents the FastText representation of a word.

ALGORITHM 4-1: Embedding matrix algorithm

```
1   Begin embedding matrix algorithm
2   Set a MAX_NB_WORDS value to denote max number of words; MAX_NB_WORDS = 100000
3       Create a tokenizer with MAX_NB_WORDS
4       Fit tokenizer on dataset (Splits sentences on spaces and treats each split value like a word).
5       Convert the dataset to sequences (replace each word with the index number of each token).
6       Calculate max_seq_len. The value is computed by using percentile. If a percentile value of 100 is used,
          the string with the largest number of words is used. A percentile value of 90 takes the number of words
          that's greater than the number of words in 90% of the sentences in the dataset.
7       Pad sequences with 0 to so that the padded sequence has length equal to max_seq_len (English)
8       Get word and index pairs from tokenizer
9       Use the smaller value nb_words between MAX_NB_WORDS and the total number of words from the
          tokenizer.
10      Create a matrix embedding_matrix of size nb_words x 300 to store embeddings for the dataset. Where
        300 is the size of fast text word embedding.
11      Load fast text model
12      Iterate over each word in word_indexes
13          If the index of the word is greater than nb_words value, skip the word
14          Get embedding vector for the word from fasttext model
15          If word vector is found, set the embedding_matrix index i of the word with index i to the  vector of
            the word embedding
16          Save embedding matrix and residual data to disk
17      Return embedding matrix along with prepared dataset, embedding matrix is passed into the model, the
          prepared dataset is used to train the model
18  End embedding matrix algorithm
```

4.6.2 Character-level Convolutional Networks (CharEmbedding)

The character embedding technique is used in phishing email detection to capture and represent the inherent structure and sequential information in the text at a character level [347]. Unlike traditional word embeddings, which focus on individual words, character embeddings consider the composition of characters within words [348]. In phishing email detection, character embeddings can uncover subtle patterns and anomalies indicative of malicious intent [347]. By encoding each character as a vector representation, the embedding captures the relationship between characters, thereby enabling the model to recognise similarities and differences in spelling, syntax and

grammar [349]. Character embeddings offer several advantages in phishing email detection. They can handle OOV words effectively [350], as they are based on the characters rather than the predefined vocabulary.

Additionally, they can capture morphological variations [351] and misspellings [352] commonly employed by attackers to deceive recipients. Phishing email detection systems can better discern between legitimate and malicious content by incorporating character embeddings into ML models. These embeddings contribute to the effectiveness of the detection process by providing a rich representation of the text's underlying structure and aiding in identifying suspicious patterns and linguistic cues commonly found in phishing emails.

4.6.2.1 CharEmbedding: Essential features for Arabic phishing email detection

Character-level embeddings offer valuable features that enhance the analysis and understanding of the text in various languages, including Arabic. These embeddings capture the fine-grained details of individual characters, enabling models to leverage the features outlined in the following sections.

4.6.2.1.1 Visual distinctions

Character-level embeddings encode the visual differences between characters, including their shapes, strokes and diacritical marks [353]. This allows models to differentiate visually similar characters, such as homographs [354], and detect potential phishing attempts or other deceptive tactics. Here is an example that highlights visual distinctions in the Arabic language using character-level embeddings: Consider the characters "ت" (ta), "ث" (tha) and "ب" (ba) in Arabic. These characters exhibit the following visual differences that character-level embeddings can capture:

- "ت" (ta): This character has two downward strokes and a horizontal line connecting them, forming a distinctive shape.
- "ث" (tha): This character also has two downward strokes connected by a curved line on the top. This curvature is visually different from the straight line in "ت".

- "ب" (ba): This character features a single downward stroke without the horizontal or curved lines present in "ت" and "ث".

Character-level embeddings represent each character as a distinct vector, capturing their visual distinctions. Subsequently, ML models can leverage these embeddings to differentiate between the characters and understand their unique properties. Visual distinctions are crucial in phishing email detection [355]. Attackers may attempt to deceive users by using visually similar characters to create URLs or email addresses that mimic legitimate ones. Character-level embeddings enable models to detect these subtle visual differences and identify potential phishing attempts based on the characters' similarity. Character-level embeddings provide a valuable feature for capturing the distinct shapes and visual cues that can aid in text analysis, language understanding and the identification of deceptive practices by considering the visual aspects of Arabic characters.

4.6.2.1.2 Contextual information

By considering the positional information of characters within a word or sentence, character-level embeddings capture the contextual cues that determine word boundaries and identify linguistic patterns. This is particularly useful in Arabic, where the absence of explicit spaces between words can pose challenges for word-level analysis. Here is an example that demonstrates the importance of contextual information in the Arabic language using character-level embeddings: Consider the word "البيت" (pronounced "al-bayt"), which means "the house" in Arabic. Character-level embeddings can capture the contextual information within this word:

- "ا" (alif): This character, when appearing at the beginning of a word, represents the definite article "the" (الـ). It provides essential contextual information about the noun that follows.
- "ل" (lam): This character, when combined with "ا", forms the definite article "al" (ـالـ). It connects to the following character and indicates that the noun is definite.
- "ب" (ba): This character represents the root letter of the word, which means "house" in Arabic. It carries the core meaning of the word.
- "ي" (ya): This character represents a vowel sound and affects the word's pronunciation. It provides phonetic information and contributes to contextual understanding.

- "ت" (ta): This character represents a feminine ending, indicating that the noun "بيت" is in the feminine gender. It provides grammatical context and affects the agreement with other elements in the sentence.

By considering the contextual information encoded in character-level embeddings, ML models can grasp the relationships between characters, understand each character's role within a word and capture the syntactic and semantic properties of the text. This contextual understanding is essential for completing accurate language processing, sentiment analysis, MT and other natural language understanding tasks in Arabic. In the context of phishing email detection, contextual information can also be crucial for identifying suspicious language patterns or deceptive practices. Models can leverage character-level embeddings to analyse the contextual cues in email content and detect phishing attempts that manipulate or misuse the context to deceive users. By capturing the contextual information present in Arabic text, character-level embeddings enhance the ability of models to accurately interpret and process the language, contributing to more effective language understanding and improved phishing email detection.

4.6.2.1.3 Morphological properties

Arabic is known for its rich morphology, with words derived from root letters through various prefixes, suffixes and vowel modifications. Character-level embeddings encode morphological variations, allowing models to understand Arabic words' internal structure and inflectional changes. Here is an example that illustrates the morphological properties in the Arabic language using character-level embeddings: Consider the root word "كتب" (pronounced "kataba"), which means "to write" in Arabic. The morphological properties of this root word can be captured through character-level embeddings:

- Derived noun: From the root "كتب," we can derive the noun "كتاب" (pronounced "kitāb"), which means "book" in Arabic. The character-level embeddings encode the relationship between the root letters "ت" ",ك," and "ب," enabling models to understand the connection between the verb and the derived noun.
- Verb conjugation: Arabic verbs undergo varied conjugations based on tense, person and gender. For example, the past tense of "كتب" in the first-person singular form is "كتبت"

(pronounced "katabtu"), meaning "I wrote". Character-level embeddings capture the morphological changes in the verb form, reflecting the addition of the suffix "-ت" for the first-person singular.

- Active participle: The active participle form of "كتب" is "كاتب" (pronounced "kātib"), which means "writer" in Arabic. Character-level embeddings represent the change from the root "كتب" to the active participle form "كاتب", incorporating the altered vowel patterns and additional letters.

Models can learn Arabic word morphological variations and patterns by leveraging character-level embeddings. This understanding allows the models to generalise across different forms derived from the same root, identify grammatical properties and make accurate predictions for various morphological contexts. Morphological properties play a significant role in Arabic and impact tasks such as information retrieval, sentiment analysis and MT. In phishing email detection, character-level embeddings that capture morphological details can aid in identifying suspicious language patterns used in phishing attempts, as attackers often manipulate morphological structures to deceive users. By incorporating character-level embeddings, models can effectively handle the morphological complexity of Arabic and improve their ability to understand and process the language, contributing to enhanced text analysis, language understanding and phishing email detection capabilities.

4.6.2.1.4 Subword representation

In languages with complex scripts, such as Arabic, character-level embeddings provide a subword representation that captures the compositionality of words. This facilitates the analysis of words with shared roots or affixes, enabling models to generalise across different word forms. Here is an example that demonstrates subword representation in the Arabic language using character-level embeddings: Consider the word "تعلمت" (pronounced "taʿallamtu"), which means "I learned" in Arabic. This word exhibits subword representation through character-level embeddings:

- Root letters: The root letters of the word are "ع" (ʿayn), "ل" (lam) and "م" (meem), which form the core meaning of the word. These root letters remain consistent across different derived forms and provide a foundation for word understanding.

- Affixes and vowels: The characters "ت" (ta) and "ت" (ta) at the beginning and end of the word represent prefixes and suffixes, respectively. These affixes indicate tense and person, modifying the meaning of the root word. Additionally, the vowels represented by diacritical marks provide phonetic information and contribute to the subword representation.

By representing Arabic words at the subword level, character-level embeddings capture the compositionality of the language. Instead of treating the entire word as a single unit, these embeddings consider the individual characters and their positional information within the word. This allows for a more fine-grained analysis and understanding of Arabic text. Subword representation is particularly beneficial in tasks such as MT, where models can generalise across different word forms derived from the same root. It also facilitates text analysis, sentiment analysis and NER by capturing the shared subword components across related words. In the context of phishing email detection, character-level embeddings that encode subword representation can aid in identifying suspicious patterns and similarities between words. Attackers may attempt to deceive users by using variations of familiar words or replacing certain subword components to create deceptive content. By leveraging subword representation, models can detect these manipulations and raise alerts for potential phishing attempts. Subword representation through character-level embeddings enhances the analysis and understanding of Arabic text, facilitating various language processing tasks and contributing to effective phishing email detection.

4.6.2.1.5 Misspellings and noise

Character-level embeddings are resilient to misspellings, as they can capture the similarity between characters with minor differences. They can also handle noisy or OCR-generated text by independently encoding individual characters rather than relying on word-level representations that could be affected by errors. Here is an example that showcases the impact of misspellings and noise in the Arabic language using character-level embeddings: Consider the word "مرحباً" (pronounced "marhaban"), which means "hello" in Arabic. Misspellings or noise can affect the accurate representation of this word as follows:

a) **Misspellings**: A common misspelling of "مرحباً" is "مرحب" (pronounced "marhab"), where the final "اً" is omitted. This omission can occur due to typographical errors or lack of attention to diacritical marks. Character-level embeddings can capture the similarity between the correct and misspelt versions, as they encode the fine-grained details of individual characters.
b) **Noise**: Noise refers to unintended characters or errors introduced during text input or transmission. For example, if the word "مرحباً" is affected by noise and appears as "مرح بان" (pronounced "marh ban"), with a space inserted between "ح" and "بان", character-level embeddings can still capture the contextual information and recognise the intended word despite the noise.

Character-level embeddings are robust to misspellings and noise because they encode each character independently. This enables models to handle variations and deviations from correct spelling and tolerate unexpected characters or errors in the text. In the context of phishing email detection, misspellings and noise can be used by attackers to evade detection or create deceptive content. They may intentionally introduce misspelt words or inject noise to mimic legitimate language usage. By leveraging character-level embeddings that account for misspellings and noise, models can identify suspicious patterns, flag potential phishing attempts and minimise the risk of falling victim to such attacks. By capturing the variations and noise commonly encountered in the Arabic language, character-level embeddings enhance the robustness and accuracy of models in text analysis, language understanding and phishing email detection tasks.

4.6.2.1.6 Capturing fine-grained details

Here is an example that demonstrates capturing fine-grained details in the Arabic language using character-level embeddings: Consider the word "سَعِيد" (pronounced "saʿīd"), which means "happy" in Arabic. This word contains several fine-grained details that character-level embeddings can capture:

a) **Diacritical marks**: The character "س" (seen) is accompanied by a diacritical mark, known as a "sukun" (ْ), indicating the absence of a vowel sound. This diacritical mark provides

phonetic information and distinguishes the pronunciation of "سَ" from other letter variations.

b) **Vowel variations**: The character "ع" ('ayn) is represented with a diacritical mark called a "fatḥa" (◌َ), which denotes the short vowel "a" sound. This vowel mark affects the pronunciation and meaning of the word, distinguishing it from other variations of the root.

c) **Letter morphology**: The character "ي" (ya) is written differently when it appears at the end of a word. In this case, it takes the form of "◌ِي" with a diacritical mark called a "kasra" (◌ِ). This subtle change in form captures the fine-grained morphological details that character-level embeddings can encode.

Character-level embeddings consider these fine-grained details by representing each character and its associated diacritical marks as distinct units. By capturing these visual and phonetic variations, character-level embeddings enable ML models to understand and differentiate between similar-looking characters with different phonetic values and meanings. This level of detail is particularly important in Arabic phishing email detection, as attackers may attempt to manipulate similar-looking characters or exploit diacritical marks to create deceptive content. By leveraging character-level embeddings, models can identify suspicious patterns that mimic legitimate language usage, helping to detect phishing attempts and protect users.

4.6.2.1.7 Morphological complexity handling

Here is an example that showcases the morphological complexity in the Arabic language: Consider the root word "كتب" (pronounced "kataba"), which means "to write" in Arabic. The morphological complexity in Arabic allows for various forms of this root word based on tense, gender, number and other linguistic features.

Examples of different morphological forms derived from the root word "كتب" include the following:

a) Past tense, masculine singular: كتبت (katabtu) - "I wrote"

b) Present tense, masculine singular: يكتب (yaktubu) - "He writes"

c) Future tense, feminine singular: ستكتب (sataktabu) - "She will write"

d) Present tense, dual form: تكتبان (taktuban) - "You (dual) write"
e) Past tense, plural: كتبنا (katabna) - "We wrote"
f) Present tense, feminine plural: يكتبن (yaktubna) - "They (feminine) write"

As demonstrated, the root word "كتب" undergoes various morphological changes, including vowel modifications, additions of prefixes or suffixes and changes in the word's internal structure. This morphological complexity is a fundamental aspect of the Arabic language. Character-level embeddings can effectively capture these morphological variations by encoding the individual characters and their positions within the word. This enables ML models or algorithms to understand the underlying structure of Arabic words and identify patterns specific to different morphological forms. By leveraging character-level embeddings and considering the morphological complexity of the Arabic language, models can improve their ability to accurately detect suspicious or deceptive language patterns in phishing emails or perform other text analysis tasks in Arabic.

4.6.2.1.8 Lack of word boundaries handling

Here is an example that demonstrates the lack of word boundaries in the Arabic language: Consider the phrase "أنا أحب الشوكولاتة" (pronounced "Ana uhibb al-shawkulata"), which translates to "I love chocolate" in English. The Arabic script has no explicit spaces between words, making it challenging to identify word boundaries solely based on the written text.

The phrase "أنا أحب الشوكولاتة" can be visually represented without spaces as follows: "أناأحبالشوكولاتة". Each word comprises individual Arabic characters connected to form a continuous sequence. Therefore, it is difficult to determine where one word ends and the next one begins without proper segmentation or analysis. This lack of clear word boundaries challenges the processing of Arabic text using traditional word-level techniques. Character-level embeddings help overcome this challenge by encoding each character and its positional information. The embeddings capture the context of characters in the phrase, allowing the model to infer and understand the boundaries between words. By leveraging character-level embeddings, ML models or algorithms can effectively learn to identify and segment words in Arabic text, aiding in tasks such as natural language processing, sentiment analysis or phishing email detection.

4.6.2.1.9 Homographic attacks handling

Here is an example of a homographic attack in the Arabic language:

Consider the word "بنك" (pronounced "bank"), which means "bank" in Arabic. Attackers can exploit the presence of visually similar characters to create a phishing email or webpage that appears legitimate. They might also use the character "٢" (Arabic-Indic digit 2) instead of the regular Arabic numeral "2", resulting in the word "بن٢" (pronounced "ban2"). In this case, the character "٢" visually resembles the Arabic letter "ن" (pronounced "n"). The attacker might design a phishing email or webpage to deceive users into thinking it is a legitimate banking communication. By replacing the "ن" with "٢", the attacker can create a visually similar word, exploiting the potential confusion and tricking users into providing sensitive information. These homographic attacks aim to exploit the visual similarity of characters while creating deceptive content that mimics legitimate messages or websites. By leveraging such techniques, attackers manipulate users into divulging their personal information, login credentials or financial details. Character-level embeddings can help distinguish between visually similar characters, such as "ن" and "٢", as they capture the subtle visual and contextual differences. This allows the phishing detection system to identify suspicious homographic attacks and raise alerts to protect users from falling victim to such phishing attempts.

Table 4.3 presents a collection of illustrative examples showcasing essential features for detecting Arabic phishing emails using character-level embeddings. These features, derived from the intricate composition of characters within words, play a crucial role in unveiling subtle patterns and identifying suspicious linguistic cues employed by attackers. By harnessing the power of character embeddings, the detection system gains the ability to effectively handle subword representation and lack of word boundaries, capture morphological variations and detect misspellings commonly utilised in phishing attempts. The comprehensive nature of these character-level features enhances the accuracy and robustness of the phishing email detection process, thereby fortifying the system's defence against phishing threats in Arabic.

Table 4.3 Key features of CharEmbedding for effective phishing email detection in Arabic

No.	English Text	Arabic Text	Issue
1	Welcome and Happiness - Pay Now to Enjoy Exclusive Services Hello, I hope this message finds you in good health and high spirits. I would like to welcome you and wish you a happy and exceptional experience with us. We are your dedicated team at Secure Payment Services. We would like to remind you that it is time to make the payment due in order to enjoy the fantastic services we provide. Please complete the payment through the enclosed link in this message. You will be directed to a secure page to easily and safely finalize the payment. If you have any questions or issues regarding the payment, feel free to contact our support team. We are committed to providing the best services to our valued customers. Just pay now to fully benefit from the range of excellent services we offer. Thank you for your trust, and we look forward to serving you again. Best regards, Secure Payment Services Team	ترحيب وسعادة - ادفع الآن للاستمتاع بالخدمات المميزة مرحب، أتمنى أن تجدك هذه الرسالة في أتم الصحة والعافية. أود أن مرح بان بك وأتمنى لك تجربة سَعِيدة ومميزة معنا. نحن فريقك المخلص في شركة خدمات الدفع الآمنة. نود أن نذكرك بأنه قد حان الوقت لادفع المبلغ المستحق للاستفادة من الخدمات الرائعة التي نقدمها. يرجى إتمام الدفع عن طريق الرابط المرفق في هذه الرسالة. سيتم توجيهك إلى صفحة آمنة لإكمال عملية الدفع بكل سهولة وأمان. إذا كان لديك أي استفسارات أو مشاكل تتعلق بالدفع، فلا تتردد في الاتصال بفريق الدعم الخاص بنا نحن ملتزمون بتقديم أفضل الخدمات لعملائنا الكرام. فقط قم بادفع الآن للاستفادة الكاملة من مجموعة الخدمات المتميزة التي نوفرها. شكراً لثقتكم بنا، ونتطلع إلى خدمتكم مرة أخرى. مع أطيب التحيات، فريق خدمات الدفع الآمنة.	**Misspellings and noise** "مرحب" (Hello) "مرح بان" (Hello) **Fine-grained details** " سَعِيدة " (happy)
2	Welcome to Our Bank - We Are Delighted to Serve You Hello, We hope this message finds you in good health and well-being. We would like to welcome you to our bank and express our happiness to serve you. At our bank, we care about meeting your financial needs and achieving your future goals. We have written this message to inform you about the services we offer and the opportunities that may be available to you. We are here to assist you in achieving your financial aspirations. We will provide you with innovative and flexible banking solutions tailored to your individual needs.	مرحبًا بك في بنكنا - نحن سعداء بخدمتك السلام عليكم، نتمنى أن تكون في أتم الصحة والعافية. نود أن نرحب بك في ب٢ك١٢ ونعبّر عن سعادتنا بخدمتك. في بنكنا، نهتم بتلبية احتياجاتك المالية وتحقيق أهدافك المستقبلية. لقد كتبنا هذه الرسالة لإعلامك بالخدمات التي نقدمها والفرص التي قد تكون متاحة لك. نحن هنا لمساعدتك في تحقيق طموحاتك المالية. سنقدم لك حلولًا مصرفية مبتكرة ومرنة تلبي احتياجاتك الفردية. تفضل بزيارة أحد فروعنا أو قم بزيارة موقعنا الإلكتروني لمزيد من المعلومات. سوف تجد فريقنا من الموظفين المتفانين والمؤهلين دائمًا جاهزًا لمساعدتك. نحن هنا للإجابة على أسئلتك ومساعدتك في اتخاذ القرارات المالية الصائبة.	**Homographic attacks** " بنكنا " "ب٢ك١٢" **Morphological complexity** "بنكنا" (our bank) "كتبنا" (written)

	Please visit one of our branches or explore our website for more information. You will find our team of dedicated and qualified staff always ready to assist you. We are here to answer your questions and help you make informed financial decisions. We are delighted to serve you and look forward to building a long and fruitful relationship with you. Rest assured that we will do our utmost to meet your banking needs in the best possible way. Thank you for choosing our bank, and we wish you a happy and successful experience with us. Best regards, Our Bank Team	نحن سعداء جدًا بخدمتك ونتطلع إلى بناء علاقة طويلة ومثمرة معك. كن مطمئنًا أننا سنبذل قصارى جهدنا لتلبية احتياجاتك المصرفية بأفضل طريقة ممكنة. نشكرك على اختيارك بنكنا ونتمنى لك تجربة سعيدة وناجحة معنا مع أطيب التحيات، فريق بنكنا.	
3	Request for Personal Information Hello, I hope this email finds you well. We kindly request you to provide some personal information to complete the necessary procedures. Please provide us with the following information: • Full Name: • Date of Birth: • Current Address: • Phone Number: • Email Address: We assure you that all the information you provide will be treated with utmost confidentiality and used only for the mentioned purposes. We will protect your personal data in accordance with our privacy policy. Thank you for your cooperation and understanding. If you have any questions or inquiries, please feel free to contact us. Best regards,	طلب معلوماث شخصية مرحبًا، آمل أن تكون في صحة جيدة. نود أن نطلب منك تقديم بعضالمعلوماتالشخصية لإكمال الإجراءات الضرورية. يرجى تزويدنا بالمعلومات التالية: • الاسم الكامل: • تاريخ الميلاد: • العنوان الحالي: • رقم الهاتف: • البريد الإلكتروني: نحن نضمن أن جميع المعلوماث التي تقدمها ستعامل بسرية تامة وستستخدم فقط للأغراض المذكورة أعلاه. سنقوم بحماية بياناتك الشخصية وفقًا لسياسة الخصوصية لدينا شكرًا لتعاونك وتفهمك. إذا كان لديك أي أسئلة أو استفسارات، فلا تتردد في التواصل معنا. مع خالص التحية،	**<u>Lack of word boundaries</u>** "بعضالمعلوماتالشخصية" provide some personal information) **<u>Visual distinctions</u>** "معلوماث" (Information)

4.6.2.2 Character embedding extractor

Method 2 mentioned in [171] was used and enhanced to present and embed the text with the increased character vocabulary. In this context, the character-level analysis of text forms a counter for the vocabulary query. Character-level embedding is applied instead of word-level embedding due to the significance of words commonly found in the text and the availability of greater information at the character level. Attackers sometimes make slight changes in the text of emails by altering insignificant characters to confuse the recipients. They may, for instance, alter "google.com" as "g00gle.com", replacing "oo" with "00". Character-level embedding also assists in detecting interpretative content, which, in turn, improves the speed and efficiency of detecting malware text. An m-sized alphabet of the input language is used for embedding the text, following which each character is embedded through one-hot coding (Figure 4.16). The sequence of characters is modified as an m-sized sequence of characters with a fixed length L. Characters longer than L are truncated, and characters not lying in the alphabet list, such as blanks, are then embedded in all-zero vector form. The operation character level embedding comprises “tokenizer, character to index, padding, one-hot encoding and embedding weight matrix”.

The main functions of a tokeniser are the processing of texts at the character level and the integration of the unknown token (UNK) in vocabulary. The tokeniser has all the required information about certain texts at the end of the training data fitting. The alphabet list has 122 characters, 26 lowercase English letters, 26 upper-case English letters, 27 Arabic letters, 10 numerals and 33 sundry characters (see Table 4.4). Any text can be processed through the character index presented in Figure 4.16 after the vocabulary set in Table 4.4 is revised. The text lies within different lengths, but the neural network only handles fixed-length vector text. Therefore, CNN analyses only batch data with text within the fixed length.

The suggested model utilises a CNN. Figure 4.16 comprises the following layers: (1) an embedding layer, (2) convolutional layers, (3) fully connected layers, and (4) an output layer. The top layer of CNN is the embedding layer for NLP queries. The embedding layer changes the one-hot matrix to an embedding vector. The embedding layer also produces vectors that allow identifying relations among characters, thereby enhancing performance.

Furthermore, two convolutional layers are applied right after the embedding layer. The convolutional filters (the filter number, filter size or kernel and pooling size) present in every convolutional layer collect useful features and detach the insignificant ones. The rectified linear unit (ReLU) activation function after the convolutional layer compresses the output from the convolutional layers.

More precisely, raw input in the text is provided in Table 4.20. In addition to 122 characters, it contains a UNK to indicate the vocabulary's uncommon characters. Subsequently, the text is truncated or subjected to zero-filling and converted to a fixed-size sequence. Ninety-five characters are then denoted by a one-hot vector implying 122 dimensions for every character. The convolution and max-pooling layers extract and minimise the text features obtained from the embedding matrix. Two fully connected layers produce the final output. Thereafter, the layers are given the pooling outcome to create an output corresponding to the number of classes. The review has been provided in Figure 4.16.

Table 4.4 Character vocabulary index (adapted from [241])

‘a	‘b	‘c	‘d	‘e	‘f	‘g’	‘h	‘i	‘j	‘k’	‘l’	‘m	‘n’	‘o’	‘p	‘q’	‘r	‘s	‘t	‘u	‘v	‘w	‘x	‘y	‘z’
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
‘A’	‘B’	‘C’	‘D’	‘E’	‘F’	‘G’	‘H’	‘I’	‘J’	‘K’	‘L’	‘M’	‘N’	‘O’	‘P’	‘Q’	‘R’	‘S’	‘T’	‘U’	‘V’	‘W’	‘X’	‘Y’	‘Z’
27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52
‘أ’	‘ب’	‘ت’	‘ث’	‘ج’	‘ح’	‘خ’	‘د’	‘ذ’	‘ر’	‘ز’	‘س’	‘ش’	‘ص’	‘ض’	‘ط’	‘ظ’	‘ع’	‘غ’	‘ف’	‘ق’	‘ك’	‘ل’	‘م’	‘ن’	‘ه’
53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78
‘و	‘ي’	‘0’	‘1’	‘2’	‘3’	‘4’	‘5’	‘6’	‘7’	‘8’	‘9’	,	;	.	!	?	:	"	“	/	\\	|	_	@	#
79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104
$	%	^	&	*	~	‘	+	-	=	<	>	(	)	[	]	{	}	UNK							
105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123							

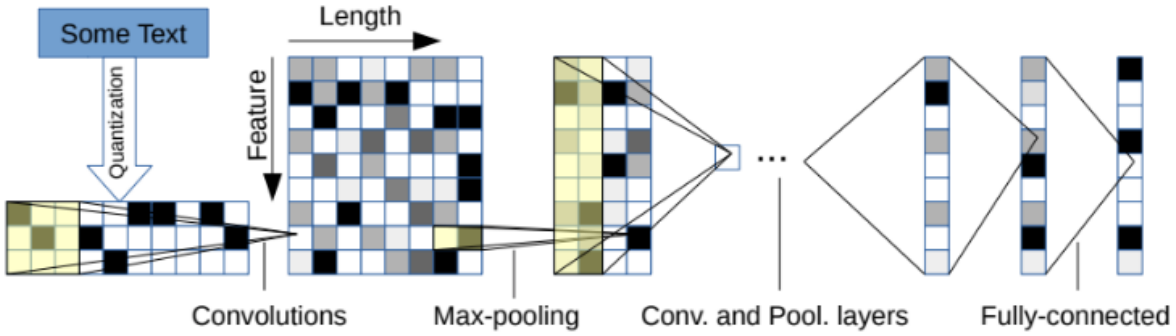


Figure 4.16 Model architecture [171]

The findings are passed on to the character embedding extractor algorithm after the pre-processing phase, extracting the features indicated in the preceding part. This algorithm is employed to identify how many actual features are subjected to the classification process. The selected features will be determined by the training dataset, which is utilised to determine how many features are required to classify the emails in the training dataset. When the training dataset is altered, the utilised features will differ. The algorithm will select a collection of significant features in distinguishing email types based on the dataset employed to develop the detection model.

Algorithm 4-2 depicts the processes of the character embedding extractor algorithm. This algorithm will analyse the training dataset to obtain the relevant set of features by building a character embedding from the data.

ALGORITHM 4-2: Character Embedding Extractor

1 Create Object for extractor and fit to docs, the fit function is responsible for creating char embedding from the data.
2 Set the vectorizer vocab to a predefined list of characters, characters_list
3 Set max sentence length (+1 for stop token)
4 If condition, runs if characters_list was not provided.
5 loop over all sentences:
6 loop over all characters in the sentence
7 check if the character is in the characters_list
8 if not, then append a character to characters_list
9 if condition, runs if the max_sentence_length was specified
10 Get the lengths of all sentences
11 Calculate what max_sentence_length is supposed to be. The value is computed by using percentile. If a percentile value of 100 is used, the largest string length is used. A percentile value of 90 takes the length that's greater than the length of 90% of the sentences in the dataset. 1 is added to the length for the end token
12 Create the stop token end_token, a vector of size of characters_list, with the last value set to 1 to indicate a stop token
13 Call the predict function to convert the dataset to character embeddings.
14 Iterate over the dataset
15 Convert each sentence into a one hot encoded vector by calling getOneHotVector function
16 Return the one hot encoded dataset
17 Run the getOneHotVector function
18 trim the sentence to be as long as max_sentence_length
19 Create empty list oneHotSentence to store character embeddings for the sentence
20 Iterate over each character C_n in the trimmed sentence
21 Create a zeroes vector Ca_{ne} the size of characters_list
22 if condition, runs if C_n is in characters_list
23 set the i^{th} element of CV_n to 1, where i is the index of the character Cn in characters_list
24 Add the created character vector CV_n to oneHotSentence list
25 Append the end_token to the end of the oneHotSentence list
26 Loop until the length of oneHotSentence is less than max_sentence_length, for each iteration on line 34, append the end_token to the end of the oneHotSentence list increasing its length by 1
27 Return the one hot encoded sentence oneHotSentence

4.7 Results of analysing emails for keyword occurrences

Phishing emails are fraudulent emails designed to trick the recipients into divulging personal information, such as passwords, credit card numbers or social security numbers. One way to identify potential phishing emails is by analysing the presence of certain keywords in the emails. Legitimate emails from trusted sources may contain specific keywords that indicate their authenticity, whereas phishing emails often use deceptive language and may contain keywords that attempt to trick the recipient into taking action. By being aware of the keywords commonly used in legitimate and phishing emails, you may be able to better protect yourself from potential threats and maintain the security of your personal information.

4.7.1 Keyword occurrences in original IWSPA-AP 2018 and APEC corpus

Table 4.5 displays the frequency of 26 English keywords in the text based on the evaluation of phishing and legitimate emails in the IWSPA-AP 2018 and APEC corpus and creates word features for the content. Table 4.6 shows the frequency of appearance of 26 Arabic keywords for every packet of assessed legitimate and phishing emails in the translated IWSPA-AP 2018 and APEC corpus. URLs and email addresses were deleted from all examined emails before evaluation for occurrences of specified keywords to prevent the inclusion of any of the specified keywords that could be present in these URLs and/or email addresses. This feature is comparable to what [116], [122], [253], [356] have suggested. Certain stemmed words have different meanings (for example, *secure* and *inconvenient* were utilised).

The existence of *"login دخول"*, *"click انقر"* and *"update حدث"* can be observed in the link text of a link. The language of the links in phishing emails frequently includes words such as *"login دخول"*, *"click انقر"* and *"update حدث"* This feature checks and records all the text of every link in an email depending on the presence or absence of the words *"login دخول"*, *"click انقر"* and *"update حدث"* in the link text. [107], [132], [133], [137], [140], [231] and [352] employed an identical feature. Lastly, a favorable word list, which consists of words suggesting phishing, is employed. It is tracked whether every word in the set appears in the email as a phishing feature. The following are the nine word stems in the set: *"account حساب"*, *"update حدث"*, *"confirm تأكيد"*, *"verify تحقق "*, *"secure أمن"*, *"notify أشعار "*, *"log دخول "*, *"click انقر"* and *"inconvenient مزعج "*.

Table 4.5 English keywords frequency for phishing and legitimate emails

Keyword	IWSPA-AP 2018		APEC	
	Legitimate emails	Phishing emails	Legitimate emails	Phishing emails
email	207	1178	43	113
account	0	705	0	162
please	145	408	32	98
click	67	376	19	57
message	71	293	71	514
service	38	262	38	262
information	49	258	13	39
update	66	199	20	35
dear	43	198	10	61
verify	0	189	0	23
access	0	159	19	141
log	0	150	0	127
secure	34	142	17	30
bank	0	114	0	66
online	0	110	0	30
address	40	105	10	33
password	0	80	0	15
important	53	73	53	73
attach	0	71	0	71
request	22	67	22	67
limited	4	65	4	24
complete	16	64	16	65
validate	0	44	0	71
credit	0	31	0	31
inconvenient	0	31	0	0
suspend	0	27	0	27
Total	**855**	**5399**	**387**	**2235**

Table 4.6 Arabic keywords frequency for phishing and legitimate emails

Keyword	Translated-IWSPA-AP 2018		APEC	
	Legitimate emails	Phishing emails	Legitimate emails	Phishing emails
بريد	213	1181	54	129
الكتروني	248	714	49	134
خاص	206	695	50	139
حساب	1	694	1	91
يرجى	140	385	0	76
انقر	59	324	16	37
معلومات	47	264	10	36
رابط	189	224	66	153
صندوق	0	221	0	24
تحقق	0	199	0	46
عزيزي	37	193	9	41
تحديث	59	192	19	34

تسجيل	0	186	0	105
مرور	0	168	0	15
دخول	0	155	0	25
ارسال	99	125	73	87
وصول	5	119	6	107
ترقيه	0	108	0	38
اجراء	1	66	1	8
ازعاج	0	39	0	28
دفع	0	39	0	32
سري	0	36	0	16
تجاوز	6	30	10	27
فيروس	0	21	0	10
تعطيل	0	20	0	9
احتيال	0	19	0	13
المجموع	**1310**	**6417**	**364**	**1460**

Table 4.7 Comparison between English and Arabic words and the difference

Keyword (English)	Legitimate emails	Phishing emails	Keyword (Arabic)	Legitimate emails	Phishing emails	Differences legitimate emails	Differences phishing emails
email	250	1291	بريد	267	1310	17	19
account	0	867	حساب	2	785	2	82
please	177	506	يرجى	140	461	37	45
click	86	433	انقر	75	361	11	72
information	62	297	معلومات	57	300	5	3
update	86	234	تحديث	78	226	8	8
dear	53	259	عزيزي	46	234	7	25
verify	0	212	تحقق	0	245	0	33
access	19	300	مرور	0	183	19	117
log	0	277	دخول	0	180	0	97
inconvenient	0	31	ازعاج	0	67	0	36
suspend	0	54	تعطيل	0	29	0	25

The findings of the tests in Tables 4.5 and 4.6 show a considerable variation in the frequency of phishing keywords against legitimate keywords in the two categories of examined emails. Furthermore, the quantity of phishing keywords in the dataset exceeds the number of legitimate keywords. This shows that phishers cannot resist using phishing keywords when launching phishing attacks. The findings, on the other hand, demonstrate that phishing keywords are generally a non in the examined legitimate emails (that is, *"account حساب", "bankبنك " , "verify " تحقق ,"password كلمة مرور ", "attachملحق", "noticeتنويه " , "credit ائتمان " ,"inconvenientمزعج "* and *"suspend معلق"*). Furthermore, due to the reasons mentioned earlier in Chapter 3 regarding the

problems associated with translation from English to Arabic, a difference in the number of words translated from English to Arabic is noticed (see Table 4.7). However, it seems that the difference is slight, and this indicates the quality of the translation in translating email content from English to Arabic.

4.8 Parallel corpus after text cleaning

Table 4.8 displays statistics after the text has been cleaned of numbers, special characters and stop words for phishing and legitimate emails. In addition, Table 4.9 shows the percentage of words left in the parallel corpus after text cleaning. The subsequent chapter will use the remaining words as features in the email classification process.

Table 4.8 Number of words left in the parallel corpus after text cleaning

Language	**Type**		**Total**
	Legitimate emails	**Phishing emails**	
English Words **(IWSPA-AP 2018)**	63008	24714	**87722**
Arabic Words **(HT)**	62561	28683	**91244**
Arabic Words **(Real cases)**	13538	7497	**21035**
English Words **(HT)**	15965	6441	**22406**
Total	**155072**	**67335**	**222407**

Table 4.9 Percentage of words left in the parallel corpus after text cleaning

Language	**Type**	
	Legitimate emails	**Phishing emails**
English Words **(IWSPA-AP 2018)**	56%	56%
Arabic Words **(HT)**	58%	66%
Arabic Words **(Real cases)**	57%	67%
English Words **(HT)**	57%	51%

4.9 Summary

In summary, the architecture of the phishing email detection model has been strategically designed to harness word-level and character-level textual analysis. After the initial data pre-processing, during which the email content is cleaned and standardized, the feature extraction phase employs NLP techniques such as TF-IDF, DTM and FastText embeddings for word-level understanding alongside a character-level CNN for fine-grained, subtle pattern detection. These methods transform the raw email data into a structured and meaningful format for the next phase. Having set the stage with efficient pre-processing and comprehensive feature extraction, the transition is now made into the classification stage, where the processed data will be utilised by state-of-the-art ML/DL classifiers to discern between legitimate and phishing emails.

Chapter Five

Discussion of the Results / Model Architectures

5.1 Overview

This chapter comprehensively evaluates the EAPD model for Arabic and English languages. The architectures of traditional and DL classifiers are elucidated. Hypothesis testing is conducted concerning the efficacy of English and Arabic phishing email detectors. A nuanced analysis delineates the variance in accuracy between phishing detection models utilising the conventional Arabic stop words set compared to the extended set. Furthermore, the advantages and drawbacks of various feature extraction methodologies are critically assessed. In this chapter's conclusion, a comparative review juxtaposes prior studies with distinct contributions.

5.2 Experimental and evaluation

The EAPD model has been analysed by contrasting it with traditional classification techniques and CNN classification models to cross-check performance and accuracy. The measurement of classifiers with respect to quality involves the employment of precision, accuracy, recall and f1-measures. The confusion matrix in Table 2.6 can be examined to understand these measures.

The traditional models used for comparison included MLP, KNN, DT, LR, SVM, RF, NB and XGBoost, whereas the CNNs used were associated with TC based on techniques such as CNN and RNN. Experiments were performed using traditional and DL methods to present equal parallel contrast between models. Furthermore, the models were selected based on their comparable and competitive outputs. The results will be correctly recorded without any bias in choosing models.

5.3 Experimental setup

All experiments in this thesis were conducted on a PC Lenovo "LEGION 5" (15IMH05H GAMING Core™ i7-10750H 2.6 GHz 1TB +512 GB SSD 16GB 15.6" (1920x1080) 144 Hz BT WIN10). The EAPD model was implemented using the scikit-learn, TensorFlow and Keras

libraries. An 80/20 training and testing split was employed for all methods. The dataset was initially imported into the Jupyter Notebook in the Anaconda environment and then categorised using text features across ten algorithms. The top 10 high-precision algorithms were further assessed based on their performance with the IWSPA-AP 2018, the translated IWSPA-AP 2018 and APEC corpora using metrics such as accuracy, precision, recall and f-measure. A key aspect of the methodology involved varying the '**random_state**' parameters during data partitioning. This approach aimed to ensure reproducibility [358], assess model variance, detect potential overfitting [359] and aid in model selection [360]. A consistent '**random_state**' parameter ensures that the results reflect the model's inherent qualities rather than data split differences. The model's resilience and generalization capacity were gauged by exploring different train-test splits. Any performance metric fluctuation across diverse '**random_state**' parameters could hint at the model's vulnerability to certain data set-ups or overfitting tendencies. The methodology, specifically iterating through '**random_state**' values from 41 to 51, was designed to provide a comprehensive view of model performance across various data configurations. Technically, the approach involved an iterative process cycling through these '**random_state**' values. In each cycle, the data was partitioned, ML models were trained, predictions were made and results were evaluated, storing accuracy for further analysis. Such rigorous testing across ten different configurations allowed deeper insights into the model's consistency and adaptability.

5.4 Traditional classifier architecture

Phishing email detection has witnessed the application of various traditional classifier architectures to discern genuine emails from malicious ones. MLP employs a feed-forward artificial neural network, optimising its weights through backpropagation. KNN classifies emails based on the majority label of its 'k' closest training examples. On the other hand, DT uses a tree-like model, making decisions based on attribute values. LR predicts the probability of an email being a phishing email, distinguishing it using a logistic function. SVM segregates emails in a high-dimensional space using hyperplanes. RF employs an ensemble of DT, providing a consensus decision.

Furthermore, NB classifies emails based on the application of Bayes' theorem with a 'naive' feature independence assumption. Lastly, XGBoost, an optimised gradient boosting library, builds an

additive model in a forward stage-wise manner. These architectures offer unique advantages and capabilities in the ongoing battle against phishing threats.

5.4.1 MLP

MLP is a feed-forward artificial neural network that includes several layers, mostly three, of neurons. Each of these neurons is referred to as a processing unit that can be activated by applying the activation function. This MLP is a supervised ML procedure where the network is trained with the help of a labelled training data set. Using a trained MLP, it is possible to map the input data set (email features in this case) to the output data set (email class). The MLP classifier for the present system contains the following parameters:

*hidden_layer_sizes=(100,), activation='relu', *, solver='adam', alpha=0.0001, batch_size='auto ', learning_rate='constant', learning_rate_init=0.001, power_t=0.5, max_iter=200, shuffle=Tru e, random_state=None, tol=0.0001, verbose=False, warm_start=False, momentum=0.9, nestero vs_momentum=True, early_stopping=False, validation_fraction=0.1, beta_1=0.9, beta_2=0.99 9, epsilon=1e08, n_iter_no_change=10, max_fun=15000.*

Figure 5.1 represents an MLP neural network with one hidden layer containing 100 neurons designed to classify phishing emails. The network has the following two main parts: the input layer, which receives the features of the input data, and the output layer, which produces the network's final classification output. The features of the input data, which are the characteristics of the email being classified as either phishing or legitimate, are received by the input layer. In this case, the specific features used are not listed but could include things such as the presence of certain keywords or phrases in the email or other characteristics that might be relevant for classifying phishing emails. The input layer is connected to a hidden layer consisting of 100 neurons. Each neuron in the hidden layer receives input from all of the neurons in the input layer and performs a weighted sum of these inputs, followed by the application of an activation function (in this case, the **ReLU** function). The output layer consists of a single neuron that produces the final classification output. This neuron receives input from each of the 100 neurons in the hidden

layer and performs another weighted sum followed by an activation function (in this case, the sigmoid function is used, as binary classification is being performed.).

The MLP uses the input features to predict whether the email is a phishing email by passing the input through the network and producing an output in the range from 0 to 1, where values closer to 1 indicate a higher probability of the email being a phishing email and values closer to 0 indicate a higher probability of it being a legitimate email.

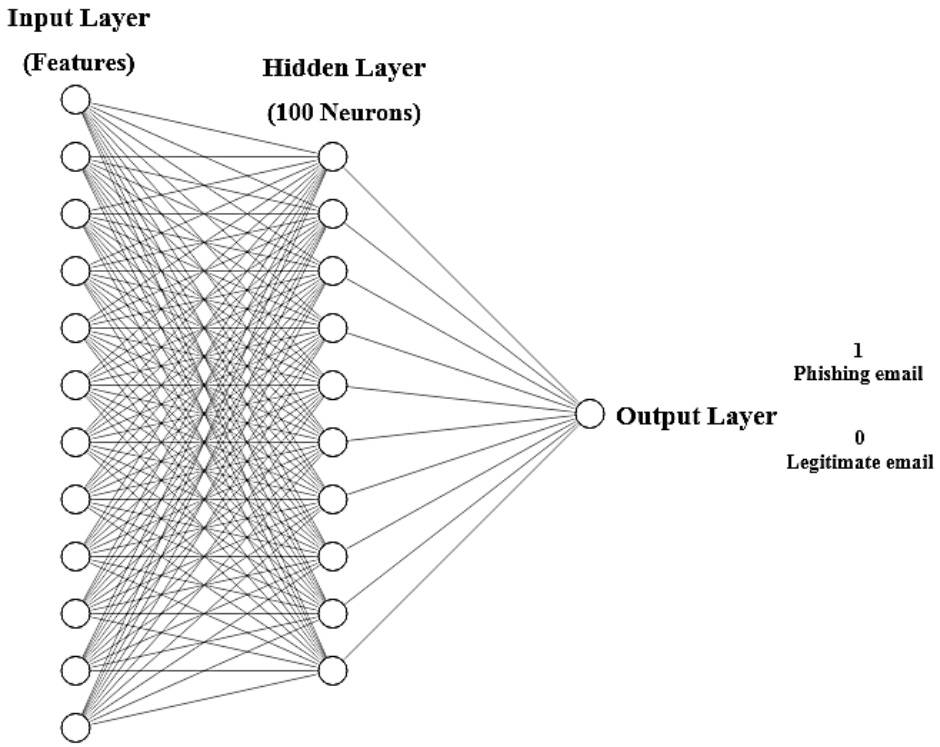


Figure 5.1 MLP classifier

5.4.2 KNN

The KNN algorithm is a powerful and intuitive ML approach used to detect phishing emails. It operates by measuring the similarity between emails based on their features, such as the frequency of specific words or phrases. In KNN, each email is represented as a data point in a multi-dimensional space, and the algorithm identifies the KNN to a given email based on a distance metric. By analysing the labels of these neighbours, KNN classifies the email as either legitimate or phishing. This algorithm is particularly effective in detecting phishing emails as it leverages the collective intelligence of similar emails to make accurate predictions. However, the performance of KNN can be influenced by the choice of k, the distance metric and the quality of the feature representation. Therefore, careful parameter selection and pre-processing are crucial to ensure reliable and efficient phishing email detection using KNN. Presently, academics prefer the KNN classifier because it is easy, polished and straightforward [361], [362]. If new sample data *x* occurs, KNN will use some distance measure to find the k neighbours closest to the unlabelled data starting from the training space. The following parameters were used in the study: You

*class sklearn.neighbors.KNeighborsClassifier(n_neighbors=5, *, weights='uniform', algorithm= 'auto', leaf_size=30, p=2, metric='minkowski', metric_params=None, n_jobs=None).*

Figure 5.2 provides evidence that the KNN algorithm utilises the local features of neighbouring data points to classify the target.

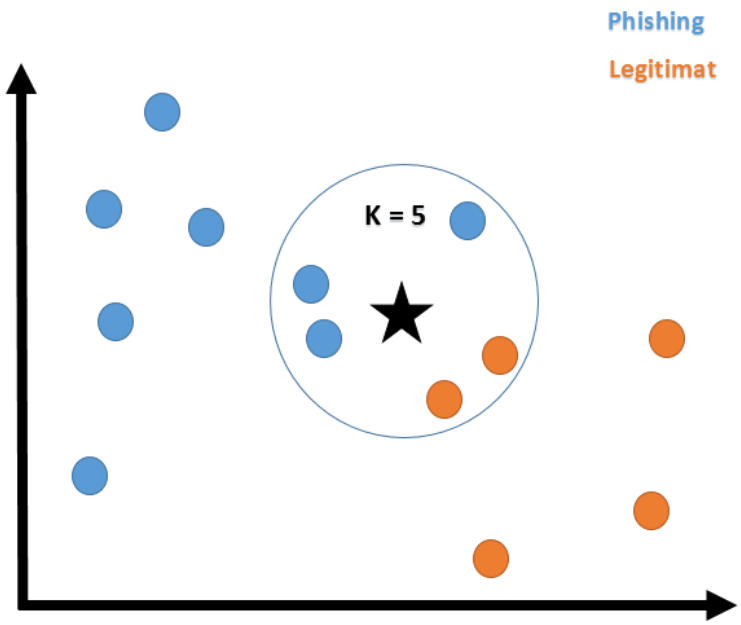


Figure 5.2 A KNN classifier

5.4.3 DT

With respect to classification and regression, DTs are a non-parametric supervised learning approach. The objective is to learn simple decision rules from data features to develop a model that predicts the significance of a target variable. Herein, a tree is equivalent to a piecewise constant. A DT classifier is a class that can classify a dataset into multiple classes. '**DecisionTreeClassifier**', similar to other classifiers, requires the following two arrays as input: a sparse or dense array *x* of shape (*n* samples, *n* features) containing the training samples and an array γ of integer values of shape (*n* samples) containing the class labels for the training samples. The following parameters were used in the ongoing study:

class sklearn.tree.DecisionTreeClassifier(, criterion='gini', splitter='best', max_depth=None,min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_decrease=0.0, class_weight=None, ccp_alpha=0.0).*

5.4.4 LR

LR is a popular and widely used ML algorithm for binary classification tasks, including phishing email detection. It is particularly effective in cases where the outcome variable is categorical, such as classifying emails as legitimate or phishing. The algorithm uses the logistic function to model the relationship between the input features (for example, email characteristics) and the binary outcome. By fitting an LR model to a labelled dataset of emails, it estimates the probabilities of an email belonging to either class. These probabilities can be further thresholded to make predictions. LR is advantageous for phishing email detection as it provides interpretable results, allowing us to understand the contribution of different features in classifying emails.

It can also handle numerical and categorical features and is relatively computationally efficient. However, the performance of LR depends on the quality and relevance of the selected features and the appropriate thresholding of probabilities. Therefore, careful feature engineering and model evaluation are crucial for effective phishing email detection using LR. The LR topology layout for the present model contains the following parameters:

*class sklearn.linear_model.LogisticRegression(penalty='l2', *, dual=False, tol=0.0001,C=1.0, fit_intercept=True, intercept_scaling=1, class_weight=None, random_state=None, solver='lbfgs', max_iter=100, multi_class='auto', verbose=0, warm_start=False, n_jobs=None, l1_ratio=None).*

5.4.5 SVM

SVM is used for pattern recognition and classification problems, as it is deemed straightforward and adequate for the computation of ML algorithms. Unlike other classifiers, the classification performance is relatively efficient due to the minimal training data. Once trained, the SVM classifier can classify new emails as either phishing or legitimate with high accuracy. SVM is particularly effective at identifying complex, non-linear patterns in the data, making it a powerful tool for detecting phishing emails that use sophisticated tactics to deceive users. The basic architecture of SVM involves mapping the input data to a high-dimensional feature space, where a hyperplane separates the data into different classes. The hyperplane is chosen to maximise the

margin between the closest data points of different classes, improving the classifier's generalisation performance (see Figure 5.3). As a result, the textual data was categorised by employing SVM in the ongoing study with the following parameters:

class sklearn.svm.SVC(, C=1.0, kernel='rbf', degree=3, gamma='scale', coef0=0.0, shrinking= True, probability=False, tol=0.001, cache_size=200, class_weight=None, verbose=False, max_ iter=- 1, decision_function_shape='ovr', break_ties=False, random_state=None).*

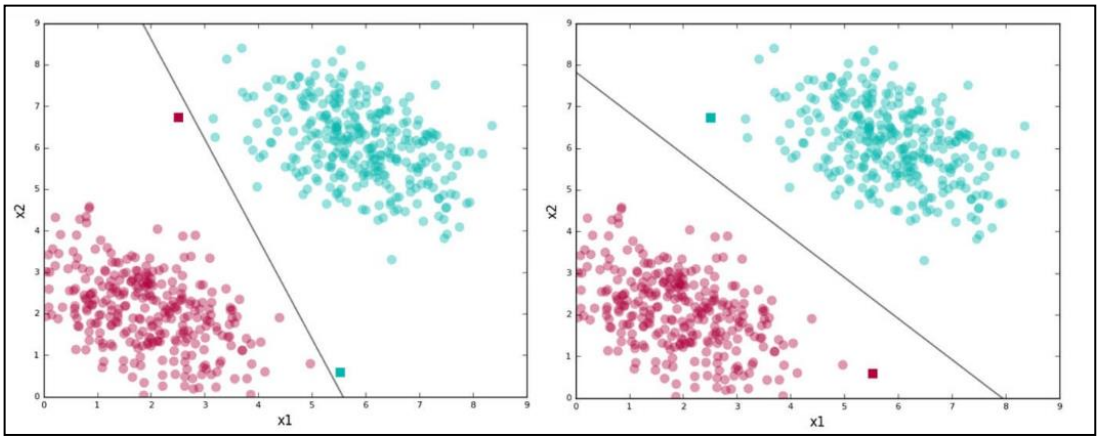


Figure 5.3 SVM classifier

5.4.6 RF

RF is a powerful ML algorithm commonly used for phishing email detection. It belongs to the ensemble learning family, combining multiple DTs to make predictions. This algorithm selects a random subset of features for each tree, ensuring diversity in the models. During the training process, the trees collectively vote on the class label of an email, and the majority vote determines the final prediction. RF is highly effective in handling high-dimensional datasets and dealing with feature interactions, crucial in phishing email detection. It is also resistant to overfitting and performs well even with noisy or incomplete data. The algorithm's ability to handle unbalanced classes is particularly valuable in phishing email detection, where legitimate emails typically outnumber phishing emails. RF's robustness, accuracy and flexibility make it a popular choice for this task, and it provides interpretable insights into feature importance. However, tuning the

hyperparameters, such as the number of trees and the maximum depth, is important to optimise the model's performance. An RF is a meta-estimator that employs averaging to increase predictive accuracy and minimise overfitting by adopting a set of DT classifiers on different sub-samples of the dataset. If bootstrap = True (default), then the max sample's parameter specifies the sub-sample size. Alternatively, the entire dataset is utilised to create every tree. The following parameters were used in the study:

*class sklearn.ensemble.RandomForestClassifier(n_estimators=100, *, criterion='gini', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features='auto', max_leaf_nodes=None, min_impurity_decrease=0.0, bootstrap=True, oob_score=False, n_jobs=None, random_state=None, verbose=0, warm_start=False, class_weight=None, ccp_alpha=0.0, max_samples=None.*

5.4.7 NB

NB is an ML algorithm commonly used in various applications, including TC tasks such as phishing email detection. It is based on the probabilistic principle of Bayes' theorem and assumes that features are independent, hence the name "naive". Despite this simplifying assumption, NB can achieve good performance in many cases. In the context of phishing email detection, NB estimates the probability of an email belonging to a particular class (legitimate or phishing), given its features. It calculates the conditional probabilities for each feature based on the training data and combines them using Bayes' theorem to make predictions. Specifically, it calculates the likelihood of observing the features given the class and multiplies it by the prior probability of the class. The predicted class is the one with the highest probability. NB is computationally efficient, even with large datasets, and it performs well when the independence assumption holds reasonably well. It is particularly effective when dealing with high-dimensional feature spaces, making it suitable for TC tasks. However, NB may struggle with correlated features, as it assumes independence, and its performance can be affected by the quality and relevance of the selected features. In the present model, the following parameters are present for the NB topology:

class sklearn.naive_bayes.GaussianNB(, priors=None, var_smoothing=1e-09).*

5.4.8 XGBoost

XGBoost is a powerful ML algorithm widely used in various domains, including phishing email detection. It belongs to the family of gradient boosting algorithms, which iteratively combine weak prediction models to create a strong ensemble model. In XGBoost, DTs are the weak learners for building the ensemble. The algorithm operates by fitting the initial DT to the data and then iteratively adding new DT, each attempting to correct the errors made by the previous trees. During each iteration, XGBoost emphasises the misclassified data points, enabling the subsequent trees to focus on those areas. The final prediction is obtained by aggregating the predictions of all the DTs in the ensemble. XGBoost offers several advantages for phishing email detection. It can handle high-dimensional datasets with a large number of features and effectively capture complex feature interactions. It also incorporates regularisation techniques to prevent overfitting, making it robust against noise and outliers. Additionally, XGBoost provides built-in mechanisms for handling missing data and supports parallel processing, leading to efficient and scalable model training.

Using gradient-boosted DT, the module sklearn.ensemble offers approaches for classification and regression. The Gradient Boosting Classifier and Gradient Boosting Regressor are provided below, along with their parameters and application. These estimators' most essential parameters are n_estimators and learning_rate. In the current research, the following parameters have been applied:

class sklearn.ensemble.GradientBoostingClassifier(, loss='deviance', learning_rate=0.1, n_estimators=100, subsample=1.0, criterion='friedman_mse', min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_depth=3, min_impurity_decrease=0.0, init=None, random_state=None, max_features=None, verbose=0, max_leaf_nodes=None, warm_start=False, validation_fraction=0.1, n_iter_no_change=None, tol=0.0001, ccp_alpha=0.0).*

5.5 DL classifier architecture

DL is a powerful technique that can be used for detecting phishing emails. It is particularly useful for this task because it can learn to identify subtle patterns in the data that may be difficult for traditional ML algorithms to detect. One common approach for detecting phishing emails using DL is using a CNN to analyse the email text. CNN can learn to extract important features from the email text, such as the presence of certain keywords or phrases, and use this information to predict whether the email is a phishing attempt. Another approach is to use an RNN, such as LSTM or GRU, to analyse the email text. This can be useful because RNNs are designed to handle sequential data, such as text, and can learn to identify patterns and dependencies across the entire email. Another approach is to use a combination of CNN and RNN to analyse the text. This will help to simultaneously extract the features and context of the text.

With respect to phishing email detection objectives, CNN and RNN models are used in isolation and conjunction, totaling six models. Several academics [46], [48], [280], [363]–[366] have used a hybrid of RNN and CNN in their studies. Subsequently, they have unequivocally remarked that this combination is far more efficient than ML models.

5.5.1 CNN

CNN is a type of DL model that is commonly used for image processing and computer vision tasks. In recent years, CNNs have also been used for TC, including detecting phishing emails. To detect phishing emails using CNNs, the email is first pre-processed, and the text is converted into a numerical representation using word technique embeddings or BOW. The CNN model is then trained on a dataset of known phishing emails and legitimate emails to learn the patterns and features that distinguish them from one another. During training, the CNN learns to automatically extract relevant features from the text and identify the patterns common in phishing emails. Once the model has been trained, it can predict whether or not a new email is a phishing email. CNNs effectively detect phishing emails, achieving high accuracy rates even on imbalanced datasets. However, as with any ML model, the performance depends on the training data quality and the chosen hyperparameters. Regular updates and monitoring are also necessary to ensure the model

continues to be effective against evolving phishing tactics. The architecture of a CNN typically consists of several layers, including the following:

- **Input Layer**: This is the first layer of the CNN, where the input image, text or video is fed into the network.
- **Convolutional Layers**: These layers detect features in the input image or video. They consist of a set of filters applied to the input image or video to extract features at different scales. Each convolutional layer typically has multiple filters, and the output of each filter is called a feature map.
- **Pooling Layers**: These layers reduce the spatial dimension of the feature maps. They typically use a max or average pooling operation to reduce the size of the feature maps while maintaining the important features.
- **Fully Connected Layers**: These layers are used to make the final decision about the input image or video. They consist of a set of neurons connected to all the neurons in the previous layer. The output of the fully connected layer is typically a set of probabilities, which represent the likelihood of the input image or video belonging to each class.
- **Output Layer**: This is the final layer of the CNN, where the network output is produced. It typically consists of a single neuron for each class, and the output is the class with the highest probability.

The basic structure of the CNN TC model is illustrated in Figure 5.4.

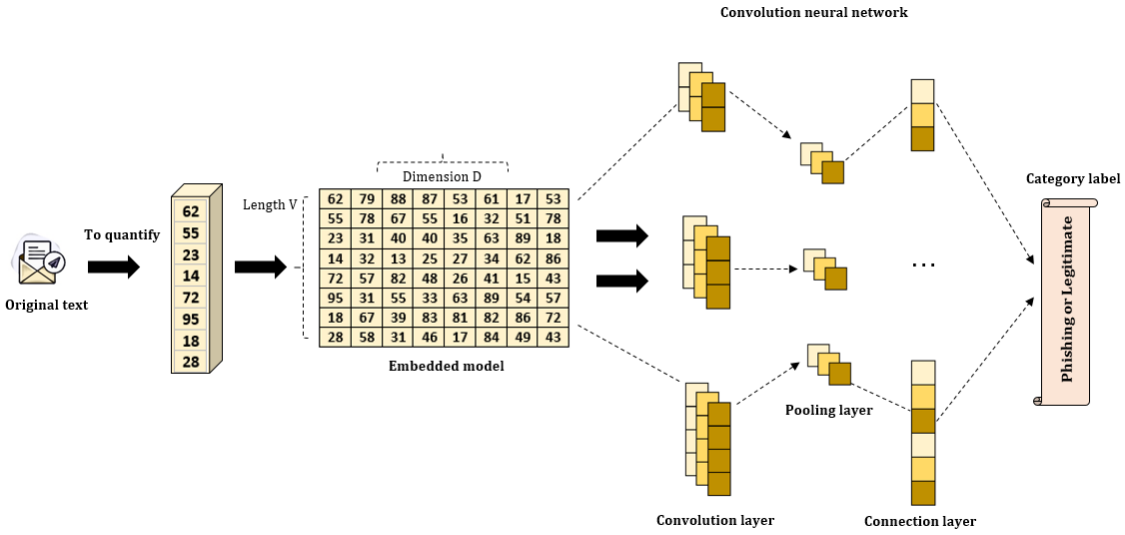


Figure 5.4 The basic architecture of the CNN text classification (adapted from [367])

Some additional layers, such as '**Dropout**', could be added to improve the performance of the CNN. Figure 5.5 visually represents the CNN model structure, composed of a series of interconnected layers. Each layer serves a distinct purpose in the model's function, and their sequential arrangement is crucial to achieving optimal outcomes.

- The first layer is a **2D convolutional layer** with 32 size filters (5, 5) and a '**ReLU**' activation function. This layer also expects the input shape to be (shape [1], shape [2], 1), where the shape is a tuple containing the dimensions of the input data.
- The second layer is a **max pooling layer** with a pool size of (2, 2), which is used to down-sample the spatial dimensions of the input data.
- The third layer is another **2D convolutional layer** with 64 filters of size (5, 5) and a **ReLU** activation function.
- The fourth layer is a dropout layer with a dropout rate of 0.3. This helps to prevent overfitting by randomly dropping out some neurons during training.

- The fifth layer is another **max pooling layer** with a pool size of (2, 2), which is used to further down-sample the spatial dimensions of the input data.
- The sixth layer is a **flattened layer** that flattens the output from the previous pooling layer into a 1D array.
- The sixth layer is a **dense** (**fully connected**) **layer** with 32 neurons and a '**ReLU**' activation function.
- The seventh layer is a **dense** (**fully connected**) layer with 16 neurons and a '**ReLU**' activation function.
- The final layer is a **dense** (**fully connected**) layer with one neuron and a **sigmoid** activation function. This is the output layer of the model, which produces a probability value between 0 and 1, representing the probability that the input text belongs to a particular class. The model is then compiled with a '**binary cross-entropy loss**' function, Adam optimizer, and accuracy metric.

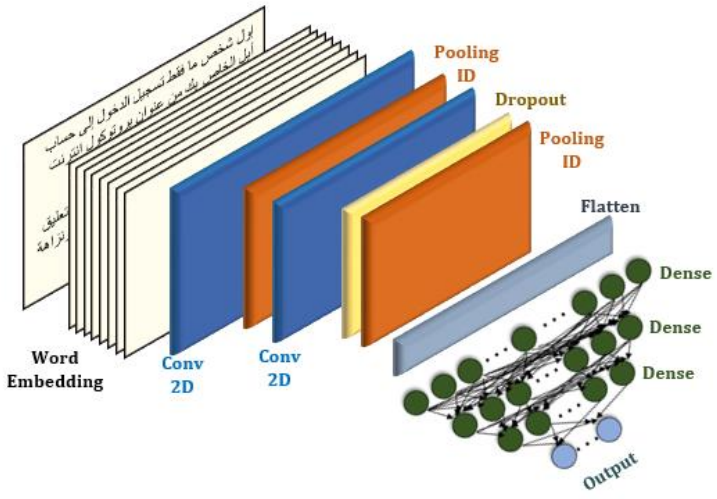


Figure 5.5 Architecture of the CNN model

5.5.2 RNNs (LSTM)

RNN is another DL model type commonly used for sequential data processing tasks such as NLP. RNNs are particularly suited for TC tasks that involve analysing the context and meaning of sequences of words, rendering them a promising approach for phishing email detection. To detect phishing emails using RNNs, the email text is first pre-processed and converted into a sequence of tokens or embeddings. The RNN model is then trained on a dataset of labelled phishing and legitimate emails to learn the patterns and features that distinguish one from the other. During training, the RNN model uses a recurrent layer to maintain a memory of previous tokens in the sequence, enabling it to capture long-term dependencies and contextual information important for distinguishing between phishing and legitimate emails. The model learns to extract features from the text and use them to predict whether an email is phishing or not.

One advantage of using RNNs for phishing email detection is that they can capture the temporal relationships and context in an email, allowing them to identify subtle patterns and indicators of phishing that may be difficult to detect using other approaches. However, as with any ML model, the performance depends on the training data quality and the chosen hyperparameters. Regular updates and monitoring ensure the model stays effective against evolving phishing tactics. One popular variant of RNN is the LSTM network, which is designed to resolve the problem of vanishing gradients.

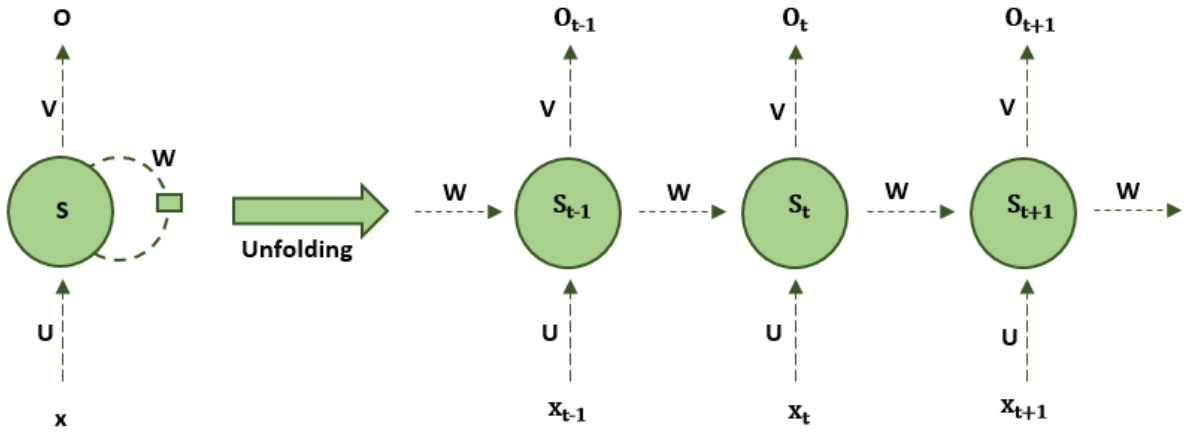


Figure 5.6 A RNN (adapted from [368])

The basic concept of RNN and unfolding during the computation in its forward computation is shown in Figure 5.6 [369]. The traditional neural network at each layer uses different parameters. Contrary to the traditional deep neural network, the same parameters ('U', 'V' and 'W' in Figure 5.6) are shared by the RNN across all steps. The following are the hidden state formulas and variables:

$$S_t = f(Ux_t + Ws_{t-1}) \tag{5.1}$$

- x_t *: The input at time step t*
- S_t *: The hidden state at time step t*
- 0_t *: The output at time step t*
- *U, V, W: The parameter matrices*

The suggested LSTM network structure for comparing the classification accuracy with CNNs and sequential CNNs is shown in Figure 5.7. An LSTM layer is directly linked with each time step, and three LSTM layers are stacked one after the other. The architecture of an LSTM cell consists of the following three gates: the input gate, forget gate and output gate, each of which is responsible for different aspects of the LSTM's memory and control flow.

- **Input Gate**: This gate controls whether or not new information should be added to the cell state. It takes the current input and the previous hidden state as input and passes them through a sigmoid activation function. The output of this gate is then multiplied with a candidate activation vector '**tanh**' to create a new memory vector that will be added to the cell state.
- **Forget Gate**: This gate controls whether or not the current cell state should forget certain information from the past. It takes the current input and the previous hidden state as input and passes them through a sigmoid activation function. The output of this gate is then multiplied by the previous cell state to decide which information to forget.
- **Output Gate**: This gate controls the amount of information output from the LSTM cell. It takes the current input and the previous hidden state as input and passes them through a sigmoid activation function. The output of this gate is then multiplied with the updated cell

state, which has been passed through a *tanh* activation function. The output of this multiplication is the current hidden state, which is then passed to the next time step.

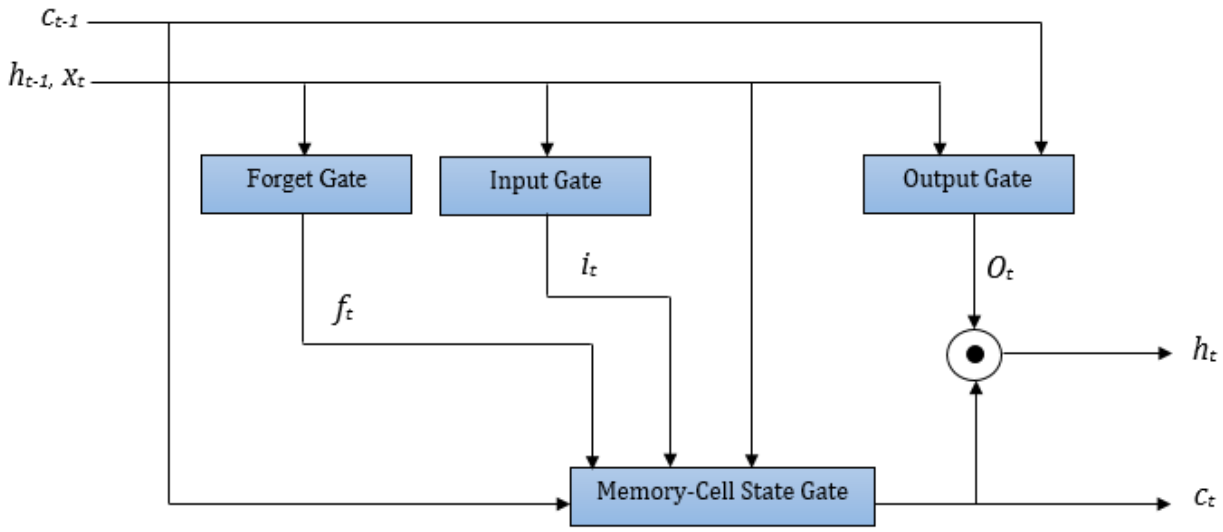


Figure 5.7 Architecture of LSTM cell (adapted from [370])

The architecture of an LSTM network typically consists of several layers, including the following:

- **Input Layer**: This is the first layer of the LSTM network, where the input data is fed into the network.
- **LSTM Layer(s)**: These layers are the heart of the LSTM network. Each LSTM layer contains memory cells, input gates, forget gates and output gates. These gates control the flow of information in and out of the memory cells, allowing the LSTM to remember information for an extended period. LSTM layers can be stacked to increase the capacity of the network.
- **Fully Connected Layer(s)**: These layers are used to make the final decision about the input data. They consist of a set of neurons connected to all the neurons in the previous layer. The output of the fully connected layer is typically a set of probabilities, which represent the likelihood of the input data belonging to each class.

- **Output Layer**: This is the LSTM network's final layer, where the network's output is produced. It typically consists of a single neuron for each class, and the output is the class with the highest probability.

As with CNNs, the architecture of LSTM can vary depending on the problem, and the number and type of layers can be adjusted to improve performance. Additionally, it is possible to use other variants of RNNs, such as GRU.

A function was defined that takes the following three arguments: '**nb_words**' '**max_seq_len**' and '**embedding_matrix**'. The function creates a sequential model using the Keras library. The function first adds an embedding layer to the model. The embedding layer is used to learn dense representations of words in a low-dimensional space (also known as word embeddings). The first argument to the embedding layer is the number of words in the vocabulary, the second is the dimension of the embeddings and the third is the maximum length of the input sequences. The embedding layer also takes an optional weights argument, set to the '**embedding_matrix**' passed to the function. This is used to initialise the embedding layer with pre-trained embeddings. The trainable argument is set to '**False**', meaning the pre-trained embeddings will not be updated during training.

The model then includes a dropout layer with a dropout rate of 0.3. This prevents overfitting by randomly dropping out some neurons during training. Subsequently, a dense layer with 32 neurons and '**ReLU**' activation is added. The model includes three LSTM layers. The first LSTM layer has 128 units and '**return_sequences = True**', which return the full sequences of successive outputs for each element in the input sequence. The second LSTM layer is similar but has 64 units, and the last LSTM layer has 32 units. The model then includes another dense layer with 32 neurons and '**ReLU**' activation followed by a dropout layer with a 0.3 dropout rate. Thereafter, a dense layer with 16 neurons and '**ReLU**' activation is added. The final dense layer has one neuron and a sigmoid activation function. Finally, the model is compiled with the '**binary_crossentropy**' loss function, Adam optimizer and an accuracy evaluation metric. The function returns the created model.

5.5.3 Bidirectional LSTM (BI-LSTM)

A BI-LSTM is a type of RNN that processes the input sequence in two directions—forward and backward. This allows the network to learn the context from the past and the future of the current time step. This can be useful in tasks such as NLP, where the meaning of a word can depend on the words that come before and after it. In a BI-LSTM, two separate LSTM networks are trained, one on the input sequence in the forward direction and another on the input sequence in the backward direction. The output of networks is then concatenated or averaged to produce the final output (see Figure 5.8). This can be useful for capturing patterns in sequences that may be missed by a single LSTM network trained in one direction alone.

BI-LSTM can be used to detect phishing emails. This task aims to classify an email as legitimate or phishing based on its contents. The input to the LSTM network can be the text of the email, which is then pre-processed to produce a sequence of word vectors. BI-LSTM can then be used to learn the context of the words in the email before and after the current word. This can be useful for capturing patterns in the language used in phishing emails, such as certain keywords, grammar and tone, which may not be obvious when looking at a single word or phrase in isolation. The output of the BI-LSTM can be fed into a fully connected layer followed by a sigmoid function to produce a probability score between 0 and 1, indicating the likelihood that the input email is a phishing email. This can be used to decide whether to flag the email as suspicious.

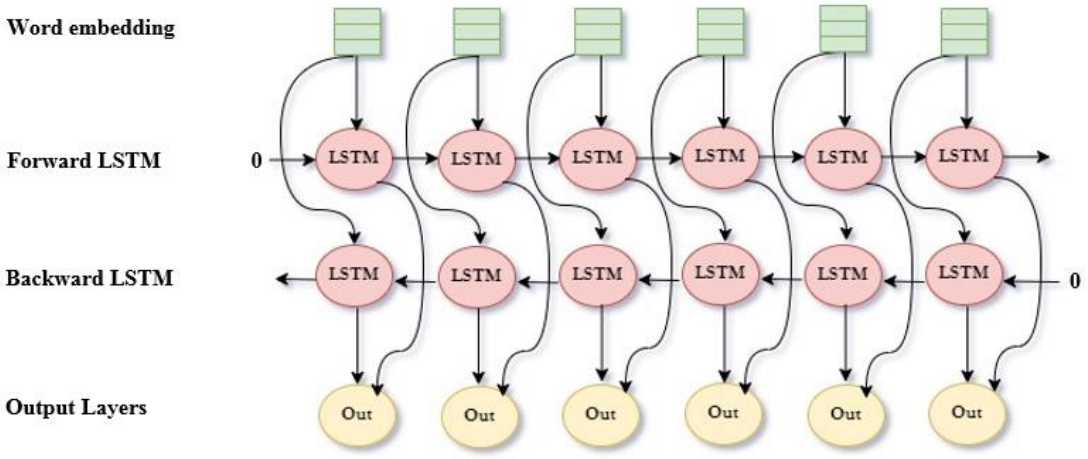


Figure 5.8 Architecture of BI-LSTM (adapted from [371])

The architecture of a BI-LSTM typically consists of several layers, including the following:

- **Input Layer**: This is the first layer of the BI-LSTM, where the input data is fed into the network.
- **Forward LSTM Layer(s)**: These layers process the input data in the forward direction. They contain memory cells, input gates, forget gates and output gates.
- **Reverse LSTM Layer(s)**: These layers process the input data in the reverse direction. They contain memory cells, input gates, forget gates and output gates.
- **Concatenation Layer:** This layer concatenates the output of the forward LSTM layers and reverse LSTM layers.
- **Fully Connected Layer(s)**: These layers are used to make the final decision about the input data. They consist of a set of neurons connected to all the neurons in the previous layer. The output of the fully connected layer typically comprises a set of probabilities, which represent the likelihood of the input data belonging to each class.
- **Output Layer**: This is the final layer of the BI-LSTM, where the network output is produced. It typically consists of a single neuron for each class, and the output is the class with the highest probability.

A function was defined that takes the following three arguments: **'nb_words**' **'max_seq_len**' and **'embedding_matrix'**. The function creates a sequential model using the Keras library. It starts by adding an embedding layer to the model. The embedding layer converts the input data (integer sequences) into fixed-sized dense vectors (embeddings). The layer takes in the following parameters:

- **nb_words**: The vocabulary size, that is, the maximum integer value + 1.
- **300**: The dimensionality of the embedding space.
- **input_length**: The length of the input sequences.
- **weights**: The embedding matrix (trained on the given dataset).

- **trainable**: A Boolean that indicates whether the embeddings should be updated during training. In this case, it is set to False, which means that the embeddings will not be updated during training.

After the embedding layer, the model contains a dropout layer with a dropout rate of 0.3, which is used to prevent overfitting. It also has a dense layer with 32 neurons and '**ReLU**' activation. Furthermore, the model contains three LSTM layers with 128, 64 and 32 units, respectively. It is wrapped by a bidirectional wrapper, making the LSTM process the data forward and backward. This allows the LSTM to learn contextual information from past and future states. After the LSTM layers, there is another dense layer with 32 neurons and ReLU activation. This is followed by a dropout layer with a dropout rate of 0.3 and a dense layer with 16 neurons and ReLU activation. Finally, the model has a final dense layer with one neuron and sigmoid activation. This output layer is used to make binary predictions. The model is then compiled with the '**binary_crossentropy**' loss function, Adam optimizer and an accuracy metric.

5.5.4 CNN-BI-LSTM

A CNN combined with a BI-LSTM network can be used to detect phishing emails. This task aims to classify an email as legitimate or phishing based on its contents. The input to the network can comprise the email text, which is first pre-processed to produce a sequence of word vectors. CNN can extract features from the input word vectors, whereas BI-LSTM can be used to learn the temporal dependencies in the sequence of features. Moreover, CNN can be used to learn features such as n-grams, which can be useful for capturing patterns in the language used in phishing emails. BI-LSTM can then be used to learn the relationships between the features over time, which can be useful for capturing patterns in the email structure and language. The network output can be fed into a fully connected layer followed by a sigmoid function to produce a probability score between 0 and 1, indicating the likelihood that the input email is a phishing email. This approach can be used to decide whether to flag the email as suspicious.

The architecture of a CNN-BI-LSTM typically consists of several layers, including the following:

- **Input Layer**: This is the first layer of the network, where the input image, text or video is fed into the network.
- **CNN Layer(s)**: These layers detect input image or video features. They consist of a set of filters applied to the input image or video to extract features at different scales. Each CNN layer typically has multiple filters, and the output of each filter is called a feature map.
- **Pooling Layer(s)**: These layers reduce the spatial dimension of the feature maps. They typically use a max or average pooling operation to reduce the size of the feature maps while maintaining the important features.
- **Fully Connected Layer(s)**: These layers are used to make the final decision about the input image or video. They consist of a set of neurons connected to all the neurons in the previous layer. The output of the fully connected layer is typically a set of probabilities, which represent the likelihood of the input image or video belonging to each class.
- **LSTM Layer(s)**: These layers are the heart of the BI-LSTM network. Each LSTM layer contains memory cells, input gates, forget gates and output gates. These gates control the flow of information in and out of the memory cells, allowing the LSTM to remember information for an extended period. LSTM layers can be stacked to increase the network capacity.
- **Output Layer**: This is the final layer of the CNN-BI-LSTM, where the network output is produced. It typically consists of a single neuron for each class, and the output is the class with the highest probability.

The architecture of the CNN-BI-LSTM network is depicted in Figure 5.9, which includes the input layer, feature extraction, sequence learning and output layer.

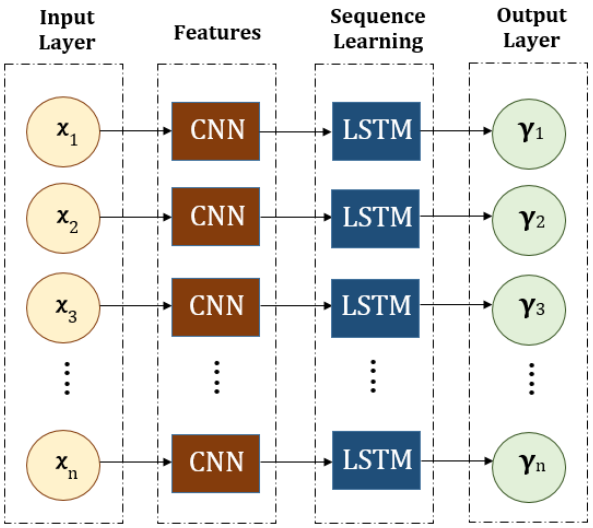


Figure 5.9 Basic architecture of the CNN-LSTM network

A function was created to define a model architecture with convolutional and BI-LSTM layers. The model utilizes the following three parameters: '**nb_words**', '**max_seq_len**' and '**embedding_matrix**'. The input to the model is passed through an embedding layer, which maps the input words to their corresponding embeddings. Subsequently, the embeddings are passed through a dropout layer to prevent overfitting.

The model then includes a sequence of 1D convolutional layers (Conv1D) followed by max-pooling layers (MaxPooling1D). These layers are used to extract features from the input data. The output of the Conv1D and MaxPooling1D layers are then passed through a dropout layer to prevent overfitting. Subsequently, it includes a dense layer with 32 units and '**ReLU**' activation followed by a BI-LSTM layer with 64 units and '**return_sequences=True**'. Another BI-LSTM layer with 32 units follows this LSTM layer. The output of this layer is then passed through a dense layer with 32 units and ' **ReLU**' activation. The output of this layer is passed through another dropout layer to prevent overfitting. The final layers of the model include a dense layer with 16 units and ' **ReLU**' activation and a final dense layer with 1 unit and '**sigmoid**' activation. This final layer makes binary predictions (0 or 1). The model is compiled with 'binary_crossentropy' as the loss function, '**adam**' as the optimizer and '**accuracy**' as the evaluation metric.

The FastText embedding matrix can train a model using a combination of CNN and BI-LSTM layers. CNN layers are used for learning local features in the input text, and LSTM layers are used for learning sequential information in the input text. The model is then trained using the '**binary cross-entropy**' loss function, the '**adam**' optimizer and the '**accuracy**' metric. When the model is trained, the data is split into train and test sets. The model is trained on the training set, and its performance is evaluated on the test set by making predictions and comparing them to the true labels.

5.5.5 GRU

GRU is another type of RNN architecture that is similar to LSTM but has a simpler architecture. It is designed to address the vanishing gradient problem in traditional RNNs and is more computationally efficient than LSTM. The GRU cell has the following two gates: the reset and update gates (see Figure 5.10).

- **Reset Gate**: This gate controls how much of the previous hidden state is used in the current timestep calculation. It takes the previous hidden state and the current input as input and passes them through a sigmoid activation function. The output of this gate is then multiplied by the previous hidden state to decide which information to reset.
- **Update Gate**: This gate controls how much new information from the current input should be used in the current timestep calculation. It takes the previous hidden state and the current input as input and passes them through a sigmoid activation function. The output of this gate is then multiplied with the candidate activation vector (*tanh*) to decide which information to update.

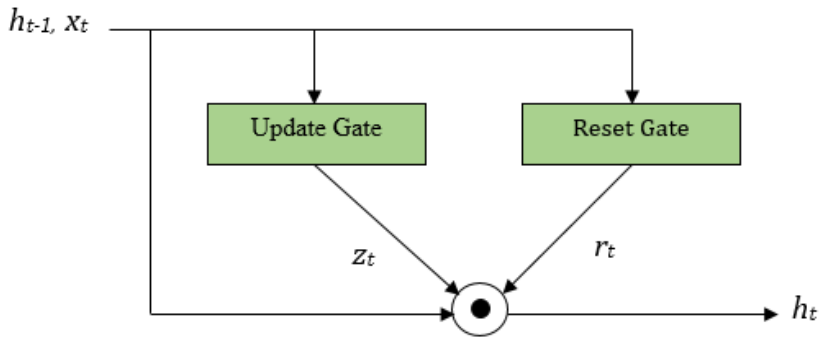


Figure 5.10 Architecture of GRU cell (adapted from [370])

GRU can be used to detect phishing emails. This task aims to classify an email as legitimate or phishing based on its contents. The input to the GRU network can comprise the email text, which is then pre-processed to produce a sequence of word vectors. The GRU can then be used to learn the context of the words in the email by learning the relationships between the words over time. This approach can be useful for capturing patterns in the language used in phishing emails, such as certain keywords, grammar and tone, which may not be obvious when looking at a single word or phrase in isolation. The output of the GRU can be fed into a fully connected layer followed by a sigmoid function to produce a probability score between 0 and 1, indicating the likelihood that the input email is a phishing email. This can be used to decide whether to flag the email as suspicious. GRUs are similar to LSTMs but with fewer parameters; therefore, they are easier to train and faster to run. They can perform well in tasks such as NLP and speech recognition.

The architecture of a GRU typically consists of several layers, including the following:

- **Input Layer**: This is the first layer of the GRU, where the input data is fed into the network.
- **GRU Layer(s)**: These layers comprise the heart of the GRU network. Each GRU layer contains update and reset gates, which control the flow of information in and out of the memory cells. These gates allow the network to selectively forget or update the previous hidden state based on the current input. GRU layers can be stacked to increase the capacity of the network.

- **Fully Connected Layer(s)**: These layers are used to make the final decision about the input data. They consist of a set of neurons connected to all the neurons in the previous layer. The output of the fully connected layer is typically a set of probabilities, which represent the likelihood of the input data belonging to each class.
- **Output Layer**: This is the final layer of the GRU, where the network output is produced. It typically consists of a single neuron for each class, and the output is the class with the highest probability.

In this section, a new function has been created to define a neural network architecture using the Keras library. The architecture includes an embedding layer, which converts the input text to a numerical representation that the network can process. The embedding layer is initialised with the '**embedding_matrix**' parameter, passed on as an argument to the function. The architecture includes several other layers, such as dropout, dense and GRU. The dropout layers prevent overfitting by randomly dropping a certain percentage of the inputs. The dense layers are fully connected layers that perform a dot product between the input and the weights and add a bias term. The GRU layers comprise RNN layers that can capture long-term dependencies in sequential data. The final layers of the architecture comprise a dense layer with 1 output and a '**sigmoid**' activation function, which is used to output a probability of the input being legitimate or phishing email. The model is then compiled with a '**binary cross-entropy loss**' function, the Adam optimizer and an accuracy metric. Three GRU layers make up the suggested GRU model. There are 128 filters in these three GRU layers. A dense layer with a '**Sigmoid**' activation function, 16 nodes, and a fully connected layer was included. Lastly, a dropout layer was included to counteract overfitting, with the dropout ratio being set to 30 percent.

5.5.6 BI-GRU

BI-GRU can also be used to detect phishing emails. A BI-GRU is similar to a regular GRU, but it processes the input sequence in forward and backward directions, allowing it to capture context from past and future words in an email. In this task, the input to the bidirectional GRU network can be the text of the email, which is then pre-processed to produce a sequence of word vectors. BI-GRU can then be used to learn the context of the words in the email by learning the relationships between the words over time in forward and backward directions. This can be useful for capturing

patterns in the language used in phishing emails, such as certain keywords, grammar and tone, which may not be obvious when looking at a single word or phrase in isolation.

The output of the bidirectional GRU can be fed into a fully connected layer followed by a '**sigmoid**' function to produce a probability score between 0 and 1, indicating the likelihood that the input email is a phishing email. This, in turn, can be used to decide whether to flag the email as suspicious. BI-GRUs can be particularly useful in tasks such as NLP, speech recognition and TC, as they can capture context from past and future words in the input sequence. The architecture of a BI-GRU typically consists of several layers, including the following:

- **Input Layer**: This is the first layer of the BI-GRU, where the input data is fed into the network.
- **Forward GRU Layer(s)**: These layers process the input data in the forward direction. They contain update gates, reset gates and memory cells.
- **Reverse GRU Layer(s)**: These layers process the input data in the reverse direction. They contain update gates, reset gates and memory cells.
- **Concatenation Layer**: This layer concatenates the output of the forward GRU layers and reverse GRU layers.
- **Fully Connected Layer(s)**: These layers are used to make the final decision about the input data. They consist of a set of neurons connected to all the neurons in the previous layer. The output of the fully connected layer is typically a set of probabilities, which represent the likelihood of the input data belonging to each class.
- **Output Layer**: This is the final layer of the BI-GRU, where the network output is produced. It typically consists of a single neuron for each class, and the output is the class with the highest probability.

The BI-GRU neural network is a type of neural network that consists of a GRU neural network with an additional three-layer structure. The additional structure allows the resulting stage to obtain all relevant data at every point in the provided input. The basic principle of the BI-GRU neural network is to aggregate the results of each of the three networks in the resulting stage after

passing the input signal by a forward neural network and a backward neural network. The three-layer BI-GRU neural network is depicted in a time series expansion manner in Figure 5.11.

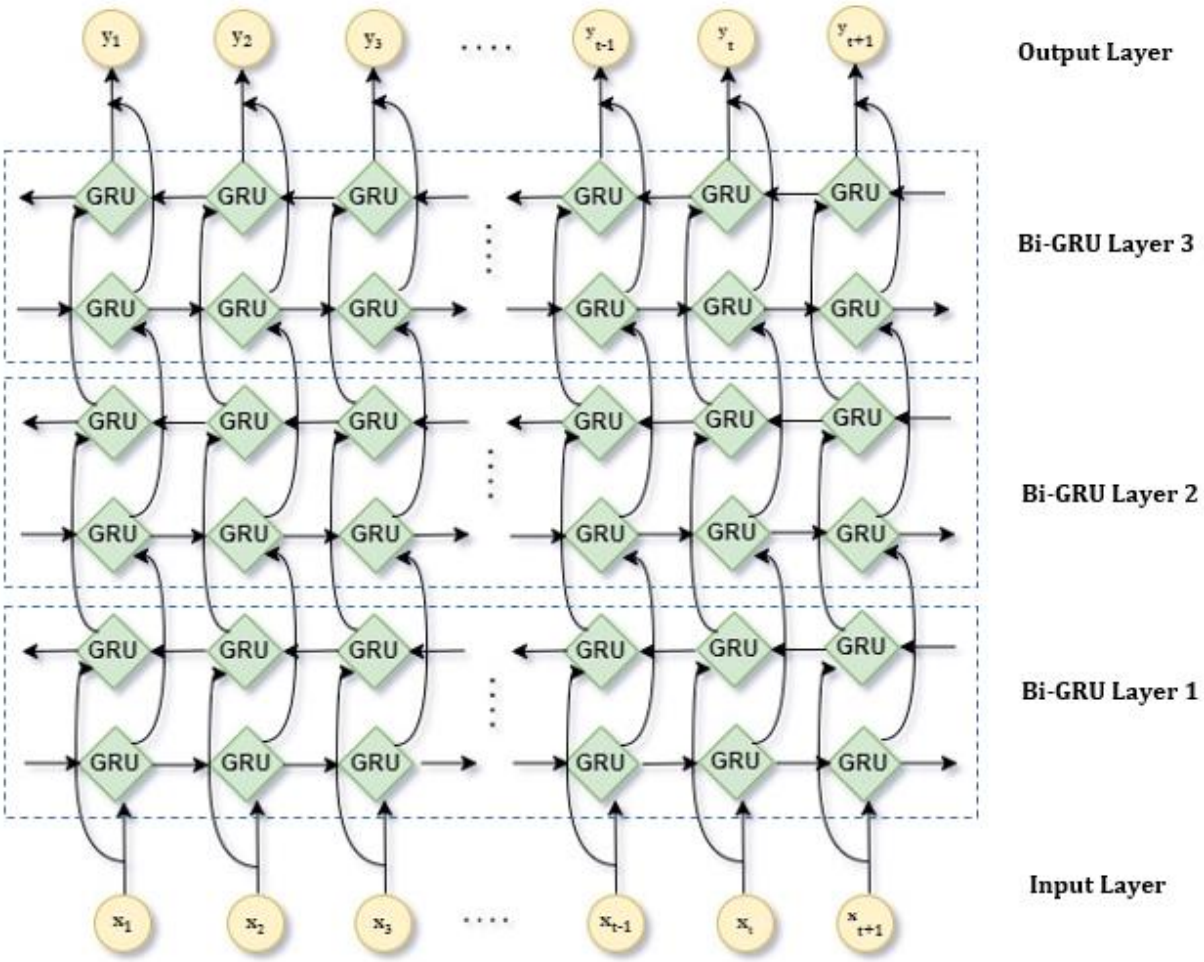


Figure 5.11 BI-GRU neural network unit structure (adapted from [372])

In the BI-GRU neural network, each layer has a forward layer that computes the hidden layer's output from the forward to the backward direction each time, whereas the backward layer computes the hidden layer's output from the backward to the forward direction each time. The output layer combines and standardises the forward and backward layer outputs each time [372].

$$\overrightarrow{h_t^1} = f\left(w_{\underset{xh^1}{\rightarrow}} x_t + w_{\underset{h^1h^1}{\rightarrow\rightarrow}} \overrightarrow{h_{t-1}^1} + b_{\rightarrow_{h^1}}\right) \quad (5.2)$$

$$\overleftarrow{h_t^1} = f\left(w_{xh^1\leftarrow} x_t + w_{h^1h^1\leftarrow\leftarrow} \overleftarrow{h_{t+1}^1} + b_{\leftarrow_{h^1}}\right) \quad (5.3)$$

$$\overrightarrow{h_t^2} = f\left(w_{\underset{h^1h^2}{\rightarrow\rightarrow}} \overrightarrow{h_t^1} + w_{\underset{h^2h^2}{\rightarrow\rightarrow}} \overrightarrow{h_{t-1}^2} + b_{\rightarrow_{h^2}}\right) \quad (5.4)$$

$$\overleftarrow{h_t^2} = f\left(w_{h^1h^2\leftarrow\leftarrow} \overleftarrow{h_t^1} + w_{h^2h^2\leftarrow\leftarrow} \overleftarrow{h_{t+1}^2} + b_{\leftarrow_{h^2}}\right) \quad (5.5)$$

$$y_t = g\left(w_{\underset{h^2y}{\rightarrow}} \overrightarrow{h_t^2} + {}_{h^2y}\overleftarrow{h_t^2} + b_y\right) \quad (5.6)$$

At time $\boldsymbol{t}$, the output vectors of the forward layer's concealed plane of the first and second stages of the BI-GRU neural network are represented by $\overrightarrow{\boldsymbol{h_t^1}} \boldsymbol{\epsilon}\, \boldsymbol{R^H}$ and $\overrightarrow{\boldsymbol{h_t^2}} \boldsymbol{\epsilon}\, \boldsymbol{R^H}$, respectively. Herein, $\boldsymbol{H}$ represents the number of divisions in a GRU cell. The initial and secondary stages of the BI-GRU neural network's resulting vectors for the concealed and backward layers at the moment $\boldsymbol{t}$ are represented by $\overleftarrow{\boldsymbol{h_t^1}} \boldsymbol{\epsilon}\, \boldsymbol{R^H}$ and $\overleftarrow{\boldsymbol{h_t^2}} \epsilon\ \boldsymbol{R^H}$, respectively, $\mathbf{t}$, $\boldsymbol{y_t} \boldsymbol{\epsilon}\, \boldsymbol{R^T}$ is a count of tags, $\boldsymbol{x_t}$ represents the neural network input at moment $\mathbf{t}$, $\boldsymbol{f(\cdot)}$ shows neural network processing of GRU, $\boldsymbol{g(\cdot)}$ represents the activation function and $\boldsymbol{R^T}$ is the value of the matching item on every tag at the moment $\boldsymbol{t}$ where $\boldsymbol{g(x)_i} = \frac{\boldsymbol{e^{z_i}}}{\sum_{\boldsymbol{k-1}}^{\boldsymbol{n}} \boldsymbol{e^{z_k}}}$, the weight indices that must be learnt are $\boldsymbol{\omega}$ and $\boldsymbol{b}$ [372].

There are three GRU layers in the suggested BI-GRU model. The output of each GRU layer is passed through a return sequences = True parameter, which means that the output of each GRU layer is a sequence of vectors of sizes 128, 64 and 32, respectively.

- A dense layer with 32 neurons and a **' ReLU'** activation function
- Another dropout layer with a dropout rate of 0.3
- A dense layer with 16 neurons and a **'ReLU'** activation function
- The output layer is the dense layer with one neuron and a **'sigmoid**' activation function

The model is compiled using a '**binary cross-entropy loss**' function, '**adam**' optimizer and an '**accuracy**' evaluation metric. Once the model has been trained, it can be used to predict the phishing emails of a given text. To train a bidirectional GRU model with pre-trained FastText embeddings on an English–Arabic dataset.

5.6 Results and discussion

In this study, the EAPD model was proposed, which integrates two distinct components – a module for analysing emails in English and another module specifically designed for classifying phishing emails in Arabic. Word-level and character-level methodologies were employed. At the word level, representation techniques such as TF-IDF, DTM and FastText embedding were utilised, facilitating the transformation of text data into a format amenable to ML algorithms. At the character level, a CNN (CharEmbedding) was employed to scrutinize the intrinsic structure of the text, thereby distinguishing patterns potentially indicative of phishing endeavors. Utilising the English–Arabic corpora (IWSPA-AP 2018), each feature extraction method was individually applied to 10 ML/DL classifiers. This systematic approach permitted a comparative analysis of classifier performance and the efficiency of various feature extraction mechanisms in counteracting phishing attempts. Performance evaluations of the EAPD model were conducted using the APEC testing dataset, culminating in selecting the most potent classifier. Through this comprehensive investigation, not only were the strengths and limitations of the EAPD model ascertained, but potential refinements were also identified and directions for subsequent research endeavors were provided.

5.6.1 Hypothesis testing

This section delves into a detailed examination of the EAPD model's capability to detect phishing emails in the Arabic language, with a focus on the nuances introduced by translations from English. Anchored by a primary hypothesis, the analysis seeks to unravel the challenges that Arabic phishing detectors face when differentiating between emails translated from English to Arabic and those crafted in native Arabic. Through the systematic exploration of this hypothesis, The aim is to emphasize the necessity for specialized phishing detection models in Arabic and elucidate the

most efficacious strategies to bolster such endeavors. The central hypothesis under investigation posits the following:

Hypothesis: There is a significant difference in the classification accuracy of Arabic phishing detectors when distinguishing between phishing emails that have been translated from English to Arabic and those that were originally written in Arabic.

5.6.1.1 Developing an English phishing email detector model

An English email detector model was developed and meticulously trained on a balanced and robust database containing 1258 emails. This dataset comprised 629 legitimate emails and an equal number of phishing emails, which were thoughtfully selected from the renowned IWSPA-AP 2018 dataset. Figure 5.12 illustrates the step-by-step approach that was employed to construct the model. The process commenced with text pre-processing to eliminate noise or irrelevant data, ensuring that the subsequent feature extraction phase would yield precise and meaningful insights. Word-level techniques, such as TF-IDF, DTM, FastText embedding, and CharEmbedding, were harnessed to effectively capture the distinctive linguistic elements present in phishing emails. Subsequently, the focus shifted to classification, where a diverse array of traditional ML models, including MLP, KNN, DT, LR, SVM, RF, NB, and XGBoost, along with DL models, namely CNN, LSTM, BI-LSTM, CNN-BI-LSTM, GRU, and BI-GRU, were employed.

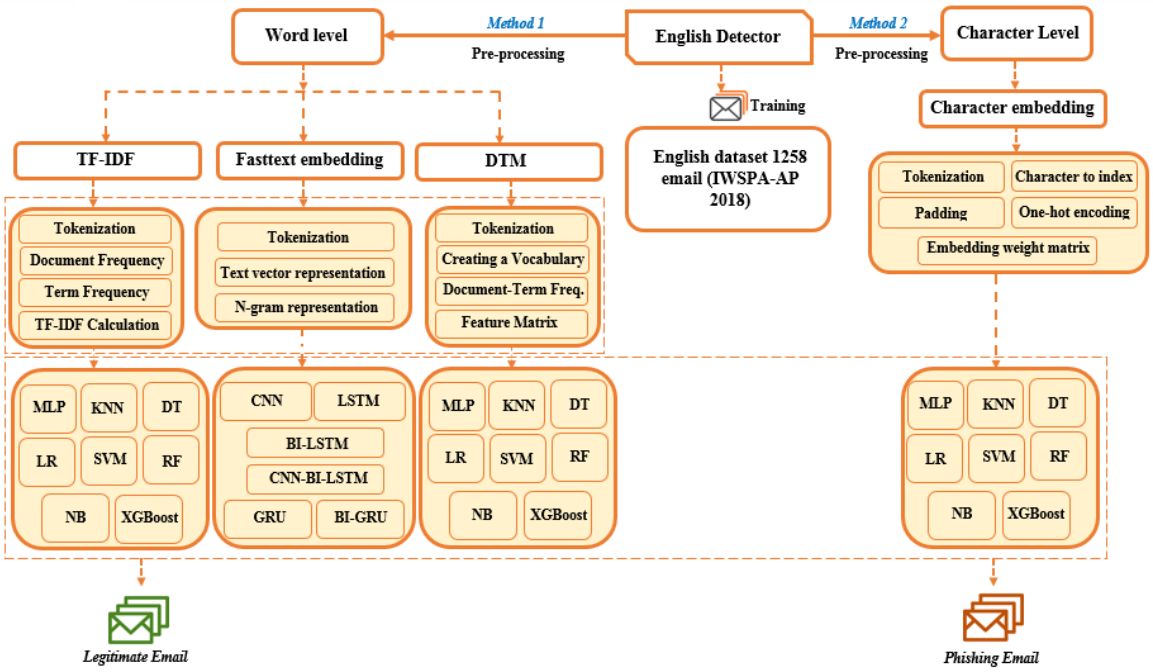


Figure 5.12 System architecture for English email detector model (training)

Two experiments were conducted to assess the model's performance.

Experiment 1: Comparison of feature extraction methods (TF-IDF, DTM and CharEmbedding) for English phishing email detection based on different traditional classifiers.

A) Comparing the effectiveness of TF-IDF with traditional ML models

Table 5.1 displays the results of using classifiers to filter phishing emails from the IWSPA-AP 2018 corpus (Original English text) with TF-IDF. SVM displayed the best performance with an accuracy of 95.7 per cent, matched by its precision and recall at 95.7 per cent, and closely followed by an F1-Score of 95.6 per cent. With respect to the processing duration, the KNN classifier demonstrated the highest speed compared to other classifiers. This noteworthy computational efficiency demonstrates the potential for the KNN classifier to be used for high-speed classification tasks on small datasets. The significance of this result lies in the fact that computational efficiency

is often a crucial factor in selecting the best classifier for a given dataset, and the KNN classifier's superior performance in this regard makes it a promising option for future classification tasks on similar datasets.

Table 5.1 Average performance metrics of conventional ML classifiers on the IWSPA-AP 2018 corpus utilising TF-IDF

Classifier	Accuracy	Precision	Recall	F1-Score	TP	FN	FP	TN	Time (sec.)
MLP	0.951	0.952	0.952	0.951	47.34	3.19	1.59	47.78	1s
KNN	0.933	0.932	0.934	0.933	47.34	3.29	3.45	45.91	**0.104s**
DT	0.888	0.889	0.888	0.888	44.80	5.83	5.36	44.01	0.141s
LR	0.939	0.941	0.940	0.939	45.68	4.96	1.11	48.25	0.107s
SVM	**0.957**	**0.957**	**0.957**	**0.956**	47.46	3.17	1.15	48.21	0.171s
RF	0.927	0.927	0.927	0.927	46.98	3.65	3.65	45.71	0.132s
NB	0.883	0.892	0.884	0.882	40.71	9.92	1.83	47.54	0.210s
XGBoost	0.901	0.906	0.902	0.901	43.02	7.62	2.26	47.10	19.4s

B) Comparing the effectiveness of DTM with traditional ML models

Table 5.2 presents the results of using DMT to apply traditional ML classifiers to the IWSPA-AP 2018 corpus. The MLP classifier achieved the highest accuracy of 95.4 per cent, with the best precision, recall and F1-Score of 95.4, 95.3 and 95.4 per cent, respectively. In terms of processing speed, the KNN classifier was the fastest among all classifiers for the IWSPA-AP 2018 corpus.

Table 5.2 Average performance metrics of conventional ML classifiers on the IWSPA-AP 2018 corpus utilizing DTM

Classifier	Accuracy	Precision	Recall	F1-Score	TP	FN	FP	TN	Time (sec.)
MLP	**0.954**	**0.954**	**0.953**	**0.954**	43.52	2.62	2.02	47.34	1s
KNN	0.809	0.840	0.811	0.804	33.07	17.26	1.864	47.50	**0.103s**
DT	0.879	0.879	0.879	0.879	44.37	6.23	5.84	43.53	0.139s
LR	0.953	0.953	0.953	0.953	47.69	2.94	1.74	47.62	0.105s
SVM	0.939	0.939	0.938	0.939	47.62	3.01	3.13	46.23	0.173s
RF	0.903	0.904	0.903	0.903	46.98	3.65	6.07	43.29	0.141s
NB	0.897	0.903	0.898	0.896	42.22	8.41	1.94	47.42	0.200s
XGBoost	0.893	0.898	0.894	0.893	42.46	8.17	2.49	46.87	18.7s

C) Comparing the effectiveness of CharEmbedding with traditional ML models

Table 5.3 displays the performance results of applying various ML classifiers to the IWSPA-AP 2018 corpus using CharEmbedding. The XGBoost classifier delivered an optimal performance with a 76.7 per cent accuracy. It also recorded the best precision, recall and F1-Score – 76.7 per cent. With respect to processing velocity, the RF classifier outpaced all other classifiers on the IWSPA-AP 2018 corpus when using CharEmbedding.

Table 5.3 Average performance metrics of conventional ML classifiers on the IWSPA-AP 2018 corpus utilising CharEmbedding

Classifier	Accuracy	Precision	Recall	F1-Score	TP	FN	FP	TN	Time (sec.)
MLP	0.752	0.754	0.751	0.751	37.94	13.19	12.06	37.30	143s
KNN	0.719	0.719	0.719	0.718	37.18	13.45	14.60	34.81	22s
DT	0.718	0.719	0.719	0.719	35.71	14.92	13.26	36.11	3s
LR	0.740	0.740	0.739	0.739	37.69	12.94	13.02	36.35	0.884s
SVM	0.733	0.734	0.733	0.733	37.02	13.61	13.06	36.31	66s
RF	0.680	0.683	0.679	0.677	38.29	12.34	19.68	29.68	**0.594s**
NB	0.636	0.637	0.636	0.635	33.57	17.06	19.37	30.00	0.985s
XGBoost	**0.767**	**0.767**	**0.767**	**0.767**	39.33	11.31	11.94	37.42	163s

Experiment 2: English phishing email detection model evaluation based on DL classifiers using FastText embedding.

Table 5.4 shows that CNN-BI-LSTM outperformed the other classifiers in terms of accuracy (92.5 per cent), precision (92.5 per cent), recall (92.6 per cent) and F1-Score (92.6 per cent). These findings suggest that the CNN-BI-LSTM model may be an effective approach for phishing email detection on English language datasets. Furthermore, the CNN-BI-LSTM model has a short training time, as it combines the strengths of CNNs and LSTM networks. CNNs are effective in capturing local features in sequential data, whereas LSTMs are good at capturing long-term dependencies. Therefore, combining these two types of networks in a bidirectional architecture

allows the CNN-BI-LSTM model to effectively capture local and global features in sequential data while requiring fewer training iterations than other models.

Additionally, the CNN-BI-LSTM model is designed with a relatively small number of parameters compared to other models, contributing to its shorter training time. This renders the model a good option for real-world applications requiring critical training time. The findings suggest that DL classifiers based on FastText embedding can be highly effective in detecting phishing emails, and CNN-BI-LSTM appears to be a promising approach for this task.

Table 5.4 Average performance metrics of conventional ML classifiers on the IWSPA-AP 2018 corpus utilising FastText.

Classifier	Accuracy	Precision	Recall	F1-Score	TP	FN	FP	TN	Time (sec.)
CNN	0.887	0.888	0.887	0.886	44.05	6.59	4.76	44.60	380.73s
LSTM	0.923	0.924	0.924	0.924	47.18	3.45	4.21	45.16	77s
BI-LSTM	0.923	0.924	0.924	0.922	46.15	4.49	3.29	46.07	131.30s
CNN-BI-LSTM	**0.925**	**0.925**	**0.926**	**0.926**	47.69	2.94	4.72	44.64	**33s**
GRU	0.923	0.924	0.925	0.925	46.79	3.85	3.69	45.67	101s
BI-GRU	0.924	0.924	0.923	0.923	45.99	4.65	3.02	46.35	262.6ss

5.6.1.2 Testing the trained EAPD model on the APEC corpus

Having successfully developed the English phishing email detection model, a rigorous experiment was conducted to evaluate its effectiveness. To ensure a comprehensive assessment, a representative sample was carefully selected from a balanced dataset comprising 300 real-world emails in Arabic. This dataset included 150 phishing emails and an equal number of legitimate emails, which were thoughtfully curated to represent both categories fairly. To emulate real-world scenarios where foreign language emails are frequently translated for analysis, GT was relied upon to accurately translate these Arabic emails into English. Figure 5.13 showcases the systematic approach adopted to test the model using a dataset of 300 emails written in Arabic. This experiment

allowed us to ascertain if English phishing email detectors, trained based on original English text, could effectively identify phishing emails that were translated from Arabic to English.

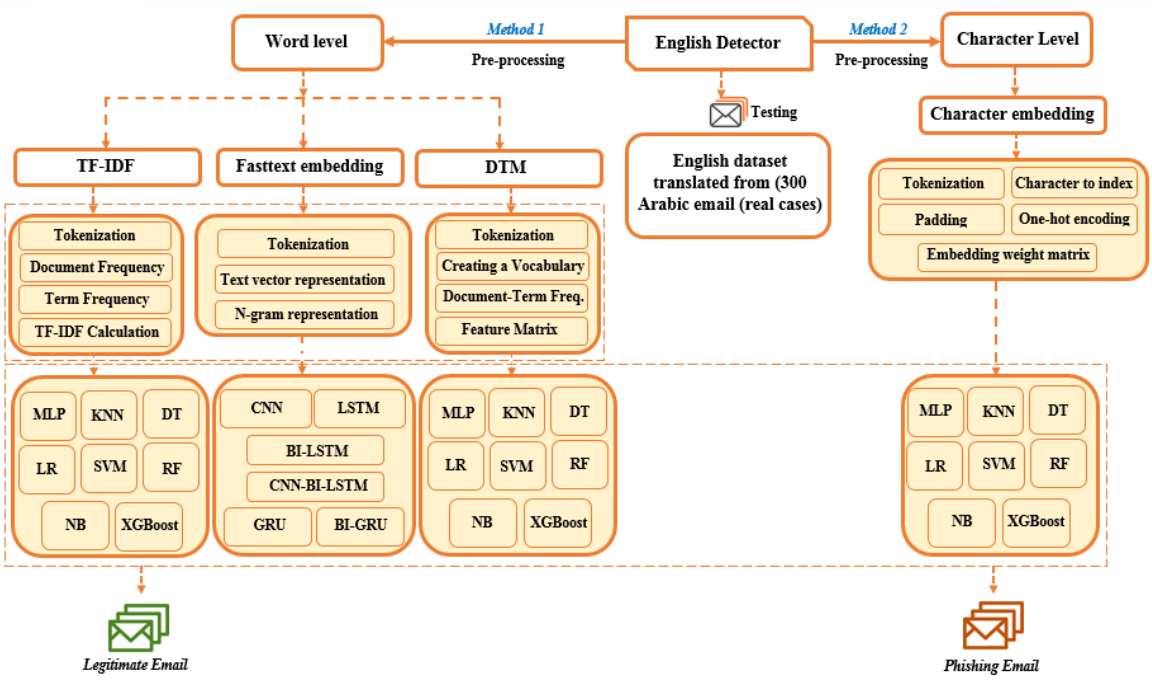


Figure 5.13 System architecture for English email detector (testing)

Table 5.5 offers a detailed comparison of two model accuracies. The first accuracy measurement pertains to the model that was trained using the original corpus (IWSPA-AP 2018). The second measurement indicates how well the same model performs when recognizing the APEC corpus that has been translated. Meanwhile, Table 5.6 provides a visual representation of the percentage improvement in accuracy. It contrasts the base performance of the trained model against its predictive accuracy on the APEC corpus. Upon examining the accompanying table, a couple of standout observations were made. Firstly, the KNN classifier, when implemented with the DTM, showcases impressive performance. Similarly, the XGBoost classifier, when paired with TF-IDF, achieves a commendable accuracy improvement rate of 4%. This highlights the strength and adaptability of these specific combinations in the classification tasks at hand.

Table 5.5 Comparing the results of phishing email detectors between the original corpus (IWSPA-AP 2018) and the translated APEC corpus.

Classifier	IWSPA-AP 2018 (Trained model)				Translated APEC - Testing			
	Char level	Word level			Char level	Word level		
	CharEmbedding	TF-IDF	DTM	FastText	CharEmbedding	TF-IDF	DTM	FastText
MLP	0.752	0.951	0.954	N/A	0.772	0.968	0.976	N/A
KNN	0.719	0.933	0.809	N/A	0.731	0.954	0.854	N/A
DT	0.718	0.888	0.879	N/A	0.724	0.897	0.889	N/A
LR	0.740	0.939	0.953	N/A	0.757	0.950	0.962	N/A
SVM	0.733	0.957	0.939	N/A	0.759	0.967	0.941	N/A
RF	0.680	0.927	0.903	N/A	0.701	0.946	0.913	N/A
NB	0.636	0.883	0.897	N/A	0.653	0.894	0.901	N/A
XGBoost	0.767	0.901	0.893	N/A	0.789	0.936	0.914	N/A
CNN	N/A	N/A	N/A	0.887	N/A	N/A	N/A	0.914
LSTM	N/A	N/A	N/A	0.923	N/A	N/A	N/A	0.935
BI-LSTM	N/A	N/A	N/A	0.923	N/A	N/A	N/A	0.936
CNN-BI-LSTM	N/A	N/A	N/A	0.925	N/A	N/A	N/A	0.947
GRU	N/A	N/A	N/A	0.923	N/A	N/A	N/A	0.935
BI-GRU	N/A	N/A	N/A	0.924	N/A	N/A	N/A	0.931

Table 5.6 The percentage improvement in accuracy

Classifier	Percentage			
	Char level	Word level		
	Char CNN	TF-IDF	DTM	FastText
MLP	2%	2%	2%	N/A
KNN	1%	2%	4%	N/A
DT	1%	1%	1%	N/A
LR	2%	1%	1%	N/A
SVM	3%	1%	0%	N/A
RF	2%	2%	1%	N/A
NB	2%	1%	0%	N/A
XGBoost	2%	4%	2%	N/A
CNN	N/A	N/A	N/A	3%

LSTM	N/A	N/A	N/A	1%
BI-LSTM	N/A	N/A	N/A	1%
CNN-BI-LSTM	N/A	N/A	N/A	2%
GRU	N/A	N/A	N/A	1%
BI-GRU	N/A	N/A	N/A	1%

The findings indicate a marginal enhancement in the prediction of phishing emails translated from Arabic to English using a model based on native English content. Despite the observable parallels in detection, the linguistic disparities between Arabic and English manifest in nuanced ways when translating phishing emails. Such emails may incorporate language intricacies that resonate with cultural allusions or emphasize distinct socio-political themes from the original language, consequently adding layers of complexity to translation. While Machine Translation (MT) tools are sophisticated, they remain susceptible to errors, which can, in turn, affect the efficacy of phishing detectors. Hence, for effective cybersecurity measures in the Arabic-speaking world, it's essential to have phishing email detectors that are attuned to both the linguistic structure and cultural intricacies of Arabic.

5.6.1.3 Developing an Arabic phishing email detector model

The need for developing an Arabic phishing email detector model arises from the increasing prominence of cyber threats targeting Arabic-speaking users. With the rising adoption of digital technologies in the Arabic-speaking world, phishing attacks have become more sophisticated and pervasive. Standard phishing email detectors, primarily designed for the English language, may struggle to effectively identify and thwart phishing attempts in Arabic emails due to language-specific complexities. Therefore, a specialised phishing email detector model that understands the nuances of the Arabic language and cultural context is essential to protect Arabic-speaking individuals and organisations from falling victim to these malicious schemes. This dedicated approach will enhance email security, contribute to cybersecurity research and bolster the defence against phishing attacks tailored for Arabic users.

One of the primary motivations for developing an Arabic phishing email detector model lies in the scarcity of available resources for training ML models in the Arabic language. Compared to English, Arabic-language datasets for phishing detection are limited, posing a challenge in building accurate and reliable models. By investing time in creating a comprehensive Arabic dataset specifically designed for phishing detection, the resource gap can be bridged, laying the foundation for robust email security measures in the Arabic-speaking world. The availability of a dedicated Arabic phishing email detector model will significantly improve the ability to detect and prevent phishing attacks tailored for Arabic users, ensuring a safer online experience for millions of individuals and organisations in the region.

An extensive experiment was conducted to address the scarcity of Arabic-language resources for phishing email detection. To create a comprehensive dataset, 1258 emails from the IWSPA-AP 2018 dataset were manually translated, simulating the content of English emails in Arabic. Using this translated dataset, ML models were trained to develop an Arabic phishing email detection model. The step-by-step construction of the model is illustrated in Figure 5.14. The process involved rigorous text pre-processing to eliminate noise and irrelevant data, ensuring precise feature extraction. The unique linguistic elements found in Arabic phishing emails were captured by employing word-level techniques and CharEmbedding. Moreover, the classification phase incorporated traditional ML/DL models.

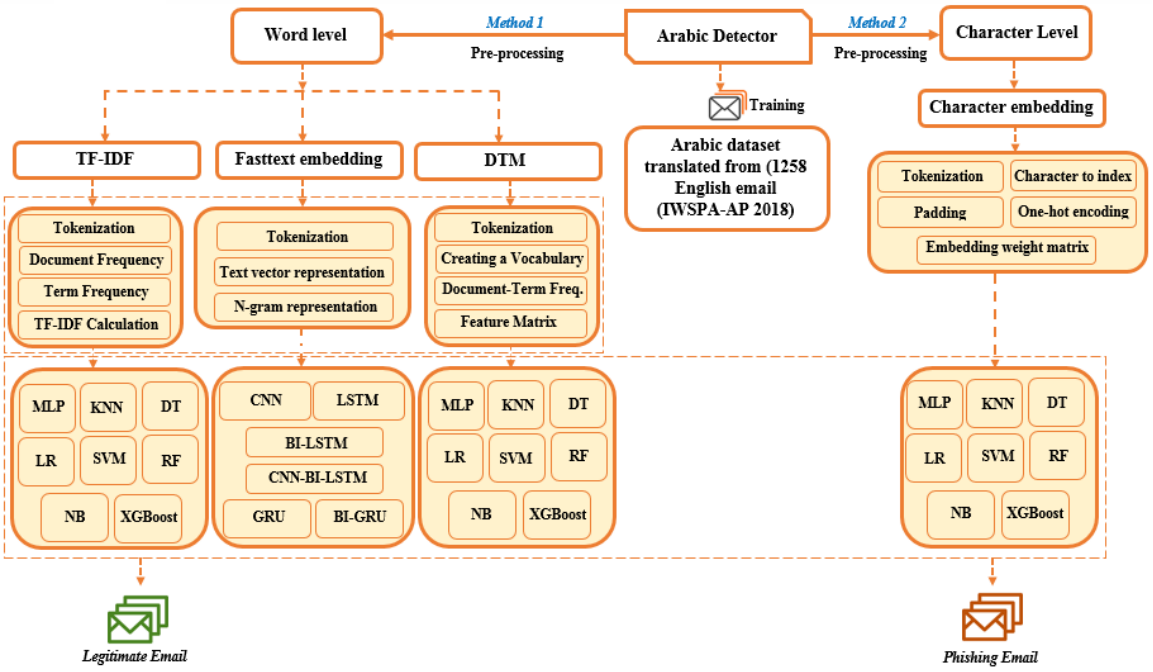


Figure 5.14 System architecture for Arabic email detector model (training)

Two experiments were executed to gauge the model's efficacy.

Experiment 1: Comparison of feature extraction methods (TF-IDF, DTM and CharEmbedding) for Arabic phishing email detection based on different traditional classifiers.

A) Comparing the effectiveness of TF-IDF with traditional ML models

The classifier performance was measured in Table 5.7 for the translated IWSPA-AP 2018 corpus (Arabic text) by applying TF-IDF. The results show that MLP achieved the top accuracy rate of 95.5 per cent, with corresponding precision and recall at 95.5 per cent and an F1-Score of 95.4 per cent. The results indicate that the performance of the classifiers was slightly lower when applied to the translated IWSPA-AP 2018 corpus as compared to the original IWSPA-AP 2018 corpus, as detailed in Table 5.1. This finding could be attributed to various factors, such as the quality of the

translation, the differences in the language structure and expressions and the availability and quality of the training data. In terms of processing time, the LR classifier outperformed other classifiers on the translated IWSPA-AP 2018 corpus.

Table 5.7 Average performance metrics of conventional ML classifiers on the translated IWSPA-AP 2018 corpus utilising TF-IDF

Classifier	Accuracy	Precision	Recall	F1-Score	TP	FN	FP	TN	Time (sec.)
MLP	**0.955**	**0.955**	**0.955**	**0.954**	47.69	2.93	1.59	47.78	5s
KNN	0.913	0.915	0.913	0.912	48.25	2.38	6.39	42.98	0.113s
DT	0.877	0.877	0.877	0.877	44.29	6.35	5.91	43.45	0.205s
LR	0.919	0.923	0.920	0.919	43.97	6.67	1.43	47.94	**0.0975s**
SVM	0.946	0.947	0.947	0.946	46.55	4.09	1.27	48.09	0.369s
RF	0.914	0.914	0.915	0.914	45.56	5.08	3.53	45.83	0.158s
NB	0.872	0.879	0.873	0.871	40.67	9.96	2.86	46.91	0.384s
XGBoost	0.902	0.905	0.903	0.902	43.37	7.26	2.54	46.83	51.5s

B) Comparing the effectiveness of DTM with traditional ML models

Table 5.8 shows the classifier performance when using DMT to the translated IWSPA-AP 2018 corpus. The MLP classifier achieved the highest accuracy of 95.2 per cent, with the best precision and recall of 95.1 and 95.2 per cent, respectively, resulting in an F1-Score of 95.1 per cent. The results indicate that classifier performance is slightly lower for the IWSPA-AP 2018 corpus than for its translated version (see Table 5.2). The KNN classifier had the quickest processing speed compared to other classifiers for the IWSPA-AP 2018 corpus and its translated version.

Table 5.8 Average performance metrics of conventional ML classifiers on the translated IWSPA-AP 2018 corpus utilising DTM

Classifier	Accuracy	Precision	Recall	F1-Score	TP	FN	FP	TN	Time (sec.)
MLP	**0.952**	**0.951**	**0.952**	**0.951**	48.09	2.54	2.30	47.06	5s
KNN	0.809	0.822	0.809	0.807	36.94	13.69	5.32	44.05	**0.101s**
DT	0.883	0.883	0.883	0.882	44.21	6.43	5.32	44.05	0.261s
LR	0.939	0.939	0.939	0.939	46.91	3.73	2.30	47.06	0.118s
SVM	0.931	0.931	0.931	0.931	46.78	3.85	3.09	46.27	0.213s
RF	0.888	0.891	0.887	0.887	46.59	4.05	7.14	42.22	0.167s
NB	0.888	0.893	0.889	0.890	42.34	8.29	2.89	46.47	0.387s
XGBoost	0.899	0.902	0.899	0.899	43.26	7.34	2.74	46.63	52.5s

C) Comparing the effectiveness of CharEmbedding with traditional ML models

Table 5.9 displays the results of classifiers when used to identify phishing emails in the translated IWSPA-AP 2018 corpus using CharEmbedding. The XGBoost classifier topped the list with accuracy, precision, recall and an F1-Score of 75.9 per cent. Classifiers tend to perform less efficiently on the Arabic corpus than the English one. The RF classifier showed the fastest processing speed among all classifiers for the IWSPA-AP 2018 corpus and its translated variant.

Table 5.9 Average performance metrics of conventional ML classifiers on translated IWSPA-AP 2018 corpus utilising CharEmbedding

Classifier	Accuracy	Precision	Recall	F1-Score	TP	FN	FP	TN	Time (sec.)
MLP	0.730	0.730	0.729	0.729	36.41	13.93	13.06	36.31	162s
KNN	0.722	0.722	0.721	0.721	38.29	12.34	15.36	33.93	26s
DT	0.684	0.684	0.684	0.683	34.36	16.27	15.36	34.01	3.5s
LR	0.758	0.76	0.759	0.758	35.11	12.42	11.75	37.62	1s
SVM	0.746	0.746	0.747	0.745	37.22	13.41	12.02	37.28	74s
RF	0.682	0.686	0.681	0.679	38.81	12.02	19.80	29.56	**0.657s**
NB	0.617	0.618	0.618	0.616	30.87	19.76	18.07	30.79	1s
XGBoost	**0.759**	**0.759**	**0.759**	**0.759**	38.47	12.46	11.67	37.69	197s

Experiment 2: Arabic Phishing email detection model evaluation based on DL classifiers based on FastText embedding.

The performance of six different classifiers, namely CNN, LSTM, BI-LSTM, CNN-BI-LSTM, GRU and BI-GRU, was evaluated on the translated IWSPA-AP 2018 corpus to determine their effectiveness in detecting Arabic phishing emails. The results presented in Table 5.10 demonstrate that the CNN classifier significantly outperformed the other classifiers in terms of average phishing email detection performance, as measured by accuracy (89.4 per cent), precision (89.6 per cent), recall (89.5 per cent) and F1-Score (89.4 per cent). These findings suggest that the CNN model may effectively detect phishing emails on Arabic language datasets.

Table 5.10 Average performance metrics of conventional ML classifiers on the translated IWSPA-AP 2018 corpus utilising FastText

Classifier	Accuracy	Precision	Recall	F1-Score	TP	FN	FP	TN	Time (sec.)
CNN	**0.894**	**0.896**	**0.895**	**0.894**	45.51	5.12	5.475	43.89	401.97s
LSTM	0.757	0.783	0.760	0.748	38.50	11.33	13	37.17	56s
BI-LSTM	0.807	0.811	0.804	0.803	39.33	10.50	8.83	41.34	161.60s
CNN-BI-LSTM	0.827	0.831	0.828	0.825	42.50	7.33	9.99	40.17	**34s**
GRU	0.777	0.798	0.787	0.774	42.50	7.33	15.00	35.17	111.1s
BI-GRU	0.787	0.796	0.792	0.784	38.67	11.16	10.17	40	303s

5.6.1.4 Testing the trained EAPD model on the APEC corpus

To ensure a comprehensive assessment, a representative sample of 300 real-world Arabic emails was carefully selected from a balanced dataset (original APEC corpus). Figure 5.15 illustrates the systematic approach adopted to test the model's performance on this dataset. The objective of this experiment was to validate the hypothesis, determining whether the Arabic phishing email detectors, trained on translated Arabic text, could effectively classify Arabic phishing emails. Through this meticulous evaluation, the aim is to fortify email security for Arabic-speaking users and contribute to the advancement of phishing detection research in the Arabic language.

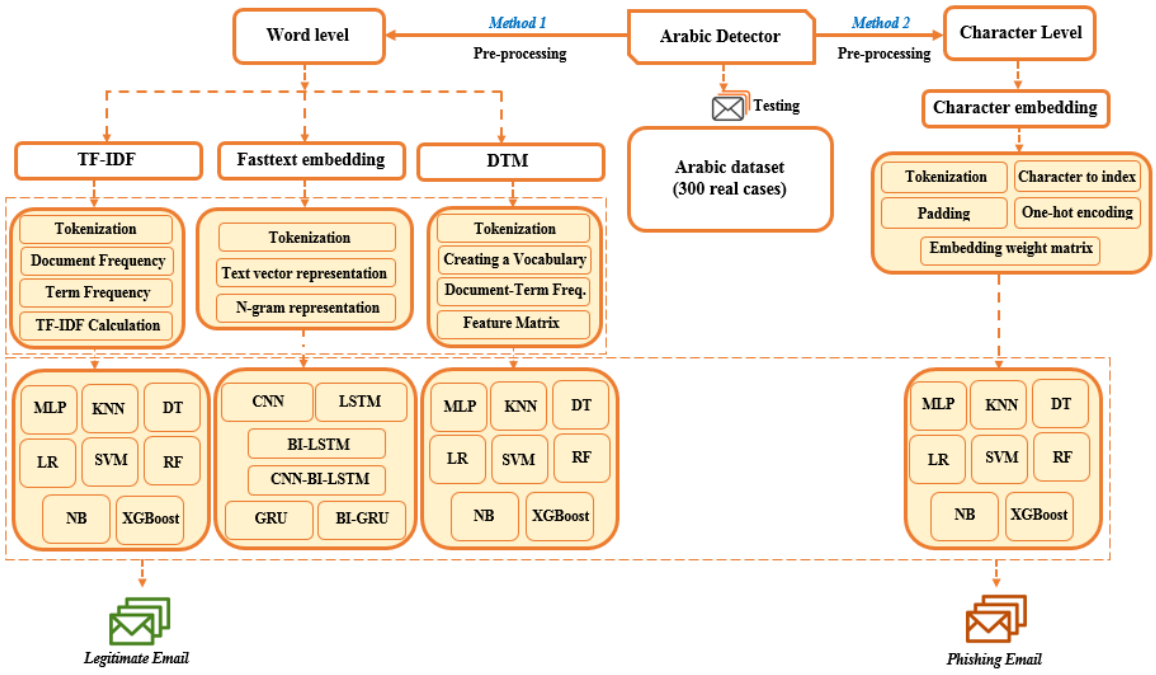


Figure 5.15 System architecture for Arabic email detector (testing)

Table 5.11 presents a meticulous comparative analysis of the accuracies of two distinct model configurations. The initial accuracy metric is derived from the model exclusively trained on the translated corpus, specifically the translated IWSPA-AP 2018 dataset. In contrast, the subsequent metric elucidates the performance proficiency of that identical model in identifying and processing the original APEC corpus. Transitioning to Table 5.12, it furnishes an illustrative depiction of the relative percentage augmentation in accuracy. This table juxtaposes the foundational performance of the trained model, as established on the translated IWSPA-AP 2018 dataset, with its predictive prowess when exposed to the APEC corpus. A thorough perusal of the concomitant table yields some salient insights. Notably, the MLP classifier, when synergized with the CharEmbedding, manifests a remarkable performance. This specific classifier and feature extraction combination culminates in a laudable enhancement in accuracy, marked at an impressive rate of 8%. This finding underscores the potential efficacy of deploying the MLP classifier in tandem with CharEmbedding for specific classification challenges within this domain.

Table 5.11 Comparing the results of phishing email detectors between the translated corpus (IWSPA-AP 2018) and the original APEC corpus

Classifier	Translated IWSPA-AP 2018 (Trained model)				Original APEC corpus (Testing)			
	Char level	Word level			Char level	Word level		
	CharEmbedding	TF-IDF	DTM	FastText	CharEmbedding	TF-IDF	DTM	FastText
MLP	0.730	0.955	0.951	N/A	0.805	0.962	0.967	N/A
KNN	0.722	0.913	0.810	N/A	0.794	0.935	0.859	N/A
DT	0.684	0.877	0.883	N/A	0.743	0.894	0.888	N/A
LR	0.758	0.919	0.940	N/A	0.772	0.928	0.940	N/A
SVM	0.746	0.946	0.931	N/A	0.807	0.956	0.965	N/A
RF	0.682	0.914	0.888	N/A	0.724	0.934	0.894	N/A
NB	0.617	0.872	0.888	N/A	0.654	0.886	0.891	N/A
XGBoost	0.759	0.902	0.899	N/A	0.795	0.924	0.912	N/A
CNN	N/A	N/A	N/A	0.894	N/A	N/A	N/A	0.923
LSTM	N/A	N/A	N/A	0.757	N/A	N/A	N/A	0.787
BI-LSTM	N/A	N/A	N/A	0.807	N/A	N/A	N/A	0.864
CNN-BI-LSTM	N/A	N/A	N/A	0.827	N/A	N/A	N/A	0.879
GRU	N/A	N/A	N/A	0.777	N/A	N/A	N/A	0.823
BI-GRU	N/A	N/A	N/A	0.787	N/A	N/A	N/A	0.835

Table 5.12 The percentage improvement in accuracy.

Classifier	Percentage			
	Char level	Word level		
	Char CNN	TF-IDF	DTM	FastText
MLP	8%	1%	2%	N/A
KNN	7%	2%	5%	N/A
DT	6%	2%	1%	N/A
LR	1%	1%	0%	N/A
SVM	6%	1%	3%	N/A

RF	4%	2%	1%	N/A
NB	4%	1%	0%	N/A
XGBoost	4%	2%	1%	N/A
CNN	N/A	N/A	N/A	3%
LSTM	N/A	N/A	N/A	3%
BI-LSTM	N/A	N/A	N/A	6%
CNN-BI-LSTM	N/A	N/A	N/A	5%
GRU	N/A	N/A	N/A	5%
BI-GRU	N/A	N/A	N/A	5%

The study reveals a subtle improvement in predicting genuine Arabic emails using a model trained on phishing emails translated from English to Arabic. Several dynamics might influence these results. For example, the translation process from English to Arabic might compromise certain phishing indicators, particularly if the emphasis is on linguistic flow rather than maintaining the original's specific cues. Furthermore, Arabic's intricate structure, characterized by its profound morphology and diverse dialects, presents challenges for automated detectors. Yet, the consistent detection accuracy implies that the essence of phishing content remains intact post-translation. To bolster the efficacy of Arabic phishing detectors, refining the Arabic stop word list could be pivotal. Such an adjustment might streamline the feature landscape, enhancing accuracy by filtering out superfluous data.

5.6.1.5 Extended Arabic stopwords list

Stop words are common words that are often removed from the text before analysis to reduce computational complexity and noise. The endeavour of expanding the list of Arabic stop words may improve the performance of ML models that use Arabic text as input by removing more unnecessary words, but it may also require more computational resources and time for pre-processing the data. The impact of expanding the stop word list on model performance will depend on the specific application and data set being used. In email security, the removal of Arabic stop

words may be an effective approach to improve the accuracy of ML classifiers in detecting phishing emails. However, it is important to carefully consider the potential benefits and drawbacks before making any changes to the stop word list, as not all stop words are created equal and removing them may not always be beneficial.

To tackle this issue, a novel list of Arabic stop words, referred to as "Salloum's list," was developed. This list is combined with other existing Arabic stop word lists. In the fourth chapter, the process behind creating this list and its potential to reduce dimensional space was delved into.. This reduction can subsequently enhance the efficiency of ML classifiers when categorising Arabic phishing emails.

A hypothesis test was performed to evaluate whether adding additional Arabic stop words significantly affected the accuracy of the phishing detection model. The following steps were taken to conduct the test:

1. **Define the null and alternative hypotheses**: The *null hypothesis (H0)* is that there is no significant difference in the classification accuracy of Arabic phishing detectors when distinguishing between phishing emails that have been translated from English to Arabic and those that were originally written in Arabic. The alternative *hypothesis (Ha)* is that there is a significant difference in the classification accuracy of Arabic phishing detectors when distinguishing between phishing emails that have been translated from English to Arabic and those that were originally written in Arabic.
2. **Determine the test statistic**: A t-test was employed to compare the mean accuracy of the two models. The test statistic was calculated as the difference between the mean accuracies divided by the standard error of the difference. The test was performed on multiple random runs/trials (10) for each method (CharEmbedding, TF-IDF, DTM, and FastText embedding) (see Table 5.13).
3. **Set the significance level**: A significance level (alpha) was established to determine the required confidence level for rejecting the null hypothesis. An alpha of 0.05 was chosen, indicating a required confidence level of 95 per cent.

4. **Calculate the p-value**: The p-value is the probability of observing the test statistic or a more extreme value, assuming the null hypothesis is true. The p-value was calculated using Microsoft Excel.
5. **Make a decision**: The null hypothesis was rejected if the p-value fell below the significance level, implying a notable difference in accuracy between the two models. Conversely, if the p-value surpassed the significance level, the null hypothesis was retained, suggesting no significant variance in accuracy. Alongside the hypothesis test, the frequency of the supplementary stop words within the training sets was examined to discern any potential correlation with accuracy.

The model was tested after excluding the expanded Arabic stop word list to gauge its influence. According to the observations in Table 5.14, conventional ML classifiers that employed CharEmbedding, TF-IDF and DTM largely remained unaffected by the augmented list of Arabic stop words (as further detailed in Table 5.13). In contrast, the DL classifiers utilising FastText embedding showed significant deviations, with LSTM registering a notable 14 per cent shift (as highlighted in Table 5.14).

Table 5.13 Applying the new Arabic stop word list on ML classifiers

Classifier	Before applying the new stop word list				After applying the new stop word list			
	Char level	Word level			Char level	Word level		
	CharEmbedding	TF-IDF	DTM	FastText	CharEmbedding	TF-IDF	DTM	FastText
MLP	0.730	0.955	0.951	N/A	0.757	0.953	0.946	N/A
KNN	0.722	0.913	0.810	N/A	0.699	0.912	0.803	N/A
DT	0.684	0.877	0.883	N/A	0.690	0.876	0.885	N/A
LR	0.758	0.919	0.940	N/A	0.756	0.918	0.940	N/A
SVM	0.746	0.946	0.931	N/A	0.737	0.945	0.931	N/A
RF	0.682	0.914	0.888	N/A	0.682	0.916	0.882	N/A
NB	0.617	0.872	0.888	N/A	0.639	0.871	0.887	N/A
XGBoost	0.759	0.902	0.899	N/A	0.754	0.899	0.900	N/A

CNN	N/A	N/A	N/A	0.894	N/A	N/A	N/A	0.901
LSTM	N/A	N/A	N/A	0.757	N/A	N/A	N/A	0.892
BI-LSTM	N/A	N/A	N/A	0.807	N/A	N/A	N/A	0.884
CNN-BI-LSTM	N/A	N/A	N/A	0.827	N/A	N/A	N/A	0.918
GRU	N/A	N/A	N/A	0.777	N/A	N/A	N/A	0.897
BI-GRU	N/A	N/A	N/A	0.787	N/A	N/A	N/A	0.895

Table 5.14 The percentage of enhancement after applying the new stop word list

Classifier	**Percentage**			
	Char level	**Word level**		
	Char CNN	**TF-IDF**	**DTM**	**FastText**
MLP	3%	0%	-1%	N/A
KNN	-2%	0%	-1%	N/A
DT	1%	0%	0%	N/A
LR	0%	0%	0%	N/A
SVM	-1%	0%	0%	N/A
RF	0%	0%	-1%	N/A
NB	2%	0%	0%	N/A
XGBoost	-1%	0%	0%	N/A
CNN	N/A	N/A	N/A	1%
LSTM	N/A	N/A	N/A	14%
BI-LSTM	N/A	N/A	N/A	8%
CNN-BI-LSTM	N/A	N/A	N/A	9%
GRU	N/A	N/A	N/A	12%
BI-GRU	N/A	N/A	N/A	11%

After filtering out the Arabic stop words based on the expanded list, the accuracy of the model trained on the translated IWSPA-AP 2018 was juxtaposed against the APEC corpus when tested on the same trained model. As shown in Table 5.16, the observed p-value stands significantly below the accepted threshold of significance. As a result, the null hypothesis was set aside. This observation highlighted a pronounced disparity in detection accuracy between the model utilizing the original set of stop words and the one employing the extended Arabic stop words, a pattern that remained consistent across all evaluated classifiers. These observations have profound implications for the research objectives. Addressing objective 1, the findings underscore the unique challenges posed by Arabic in email detection studies, particularly the variability in efficiency due to specific stop words. This directly ties to objective 6, suggesting a pronounced advantage in refining the Arabic stop words list, with deep learning classifiers like LSTM exhibiting notable performance shifts. These insights further highlight the need for improved feature extraction techniques, aligning with objective 5. The evaluation from Table 5.16 provides a deeper understanding related to objective 2, emphasizing the significance of grasping dataset limitations. Moreover, these findings potentially extend their relevance beyond Arabic, underscoring the comprehensive aim of the EAPD model to enhance phishing email detection in both English and Arabic languages.

Table 5.15 Comparison of Arabic phishing detector outcomes between translated (IWSPA-AP 2018) and original APEC corpora post stop-word list update

Classifier	Translated IWSPA-AP 2018 (Trained model)				Original APEC corpus (Testing)			
	Char level	Word level			Char level	Word level		
	Char CNN	TF-IDF	DTM	FastText	Char CNN	TF-IDF	DTM	FastText
MLP	0.757	0.953	0.946	N/A	0.835	0.972	0.965	N/A
KNN	0.699	0.912	0.803	N/A	0.765	0.943	0.869	N/A
DT	0.690	0.876	0.885	N/A	0.779	0.896	0.943	N/A
LR	0.756	0.918	0.940	N/A	0.832	0.945	0.966	N/A
SVM	0.737	0.945	0.931	N/A	0.865	0.968	0.957	N/A
RF	0.682	0.916	0.882	N/A	0.874	0.954	0.952	N/A
NB	0.639	0.871	0.887	N/A	0.736	0.923	0.922	N/A
XGBoost	0.754	0.899	0.900	N/A	0.798	0.946	0.942	N/A

CNN	N/A	N/A	N/A	0.901	N/A	N/A	N/A	0.932
LSTM	N/A	N/A	N/A	0.892	N/A	N/A	N/A	0.929
BI-LSTM	N/A	N/A	N/A	0.884	N/A	N/A	N/A	0.935
CNN-BI-LSTM	N/A	N/A	N/A	0.918	N/A	N/A	N/A	0.938
GRU	N/A	N/A	N/A	0.897	N/A	N/A	N/A	0.923
BI-GRU	N/A	N/A	N/A	0.895	N/A	N/A	N/A	0.923

Table 5.16 Hypothesis testing results

Classifier	P-Value (t-test)	Decision
CharEmbedding	0.001	Supported
TF-IDF	0.031	Supported
DTM	0048	Supported
FastText	0.001	Supported

5.6.2 Pros and cons of feature extraction methods

This experiment examines four different methods for extracting features to determine the most effective method for the proposed framework. The four methods are CharEmbedding, FastText embeddings, TF-IDF and DTM. The results are presented in Table 5.17, which shows that FastText embeddings have the highest performance among the other feature extraction methods, whereas CharEmbedding has the lowest performance. DTM and TF-IDF are the second-best methods. The pros and cons of each feature extraction method are discussed in Table 5.18.

Table 5.17 Pros and cons of each feature extraction method

Method	Pros	Cons
CharEmbedding	• Character-level features can be effective for capturing important linguistic patterns in text, such as spelling errors or unusual word formations. • Character-level features may be more robust to changes in word order or syntax, since they are based on individual character sequences rather than whole words. • Character-level features can be especially useful for languages with complex writing systems or non-standard orthographies.	• Character-level features can require more computational resources and time to extract, since they involve working with individual characters rather than whole words or phrases. • Character-level features may be less effective at capturing semantic or contextual information, since they focus on surface-level patterns rather than deeper meaning. • Character-level features may be more prone to noise or irrelevant information, such as extraneous punctuation or formatting.

FastText embeddings	• Captures both semantic and syntactic information of words, including their context and morphology, which can lead to better performance in downstream tasks. • Can handle out-of-vocabulary words and rare words better than traditional word embeddings, by representing them as a combination of subword n-grams. • Can be trained on large amounts of unlabeled text, making it a useful tool for tasks where labeled data is limited. • FastText embeddings have been shown to perform well in a variety of NLP tasks, such as text classification, sentiment analysis, and machine translation.	• FastText models can be computationally expensive to train, especially on large datasets. • The quality of the embeddings may be affected by the quality and size of the training corpus. • The subword representations may not always capture the exact meaning of the word, and can sometimes lead to noisy or ambiguous embeddings. • Since FastText embeddings rely on subword information, they may not be ideal for tasks where the exact spelling or morphology of a word is not important.
TF-IDF	• Helps in document ranking: TF-IDF is commonly used in information retrieval systems to rank documents based on their relevance to a search query. • Considers word frequency: TF-IDF takes into account the frequency of a word in a document and in the entire corpus. This means that it is able to give more importance to words that are rare in the corpus, but frequent in a particular document. • Language independent: TF-IDF can be used with any language and does not require any prior knowledge of the language. • Feature selection: TF-IDF can be used as a feature selection method to identify the most important words in a corpus. • Simple and efficient: TF-IDF is a simple and efficient method that can be easily implemented.	• Ignores word order: TF-IDF does not take into account the order in which words appear in a document. This means that it can sometimes give misleading results, especially in cases where the order of words is important. • Cannot handle synonyms: TF-IDF treats words as independent units and cannot handle synonyms or words that have similar meanings. • Cannot capture context: TF-IDF cannot capture the context in which a word is used. This means that it may not be able to distinguish between different meanings of a word, depending on the context in which it is used. • Requires pre-processing: TF-IDF requires pre-processing of the corpus, which can be time-consuming and may require domain-specific knowledge. • Can be affected by document length: TF-IDF can be affected by document length, as longer documents may contain more words and therefore have lower TF-IDF scores for each word.
DTM	• Simple and flexible: DTMs are easy to create and can be used with a variety of text analysis techniques, including clustering, topic modeling, and sentiment analysis. • Provides a quantitative representation of text data: DTMs convert textual	• Can be affected by stop words: Stop words (common words such as "the" and "and") can appear frequently in a DTM, even though they do not carry much meaning. This can make it difficult to identify important terms. • Cannot capture word order: DTMs do not take into account the order in

	data into a numerical format that can be analyzed using statistical methods. • Allows for easy data exploration: DTMs can be used to explore relationships between terms and documents, which can help identify patterns and insights. • Supports efficient data storage: DTMs can be stored in a compact format, making it possible to analyze large amounts of text data. • Can handle multiple languages: DTMs can be used with text data in any language.	which words appear in a document, which can be important for certain types of analysis. • Requires pre-processing: DTMs require pre-processing steps such as tokenization, stemming, and stop word removal, which can be time-consuming and may require domain-specific knowledge. • Limited context: DTMs do not capture the context in which a word is used, which can limit their usefulness for certain types of analysis. • Sparse data: DTMs can be very large and sparse, with many cells containing zero values. This can make it difficult to analyze and interpret the data, and may require specialized techniques such as matrix factorization.

5.6.3 Comparative analysis

To provide a comprehensive evaluation of the effectiveness of the model, a comparison was sought against other studies that utilized the same original IWSPA-AP 2018 corpus. In Table 5.18, this work has been compared with the works of a few additional researchers. Nine additional papers are similar to the current work. Additional accuracy measurements have been mentioned in some research studies. When comparing most of the current work, the accuracy was either better [10], [38], [270] and [271] or the scores were similar [280]. The highest accuracy level can be identified in the table, considering the current work datasets mentioned at the lowest table level. This table lists these studies and their respective results for easy comparison. By doing so, a better understanding of the strengths and limitations of the approach in relation to similar research efforts can be obtained.

This study stands out from others in the context of phishing email detection by utilising the most important feature extraction methods, namely, TF-IDF, DTM, FastText embedding and CharEmbedding, at the word and character levels. The unique advantages of each method can be harnessed by employing these techniques, resulting in a comprehensive evaluation of the classifiers. Additionally, this study is the first to address the detection of phishing emails written

in Arabic and English. Therefore, the architecture of the classifiers was modified to suit both languages, which may explain the relatively lower accuracy compared to the studies [10], [38], [270] and [271], which only focused on the English language. Nonetheless, this study provides valuable insights into the challenges of detecting phishing emails written in different languages.

In contrast to earlier research, the study employed a distinct methodology by leveraging ten different seed values for training and testing the model. The average was derived from these outcomes. This variation in approach could account for the observed decrease in accuracy compared to past studies. Methodically cycling through '**random_state**' values ranging from 41 to 51, the aim was to understand the model's performance across diverse data setups holistically. This procedure involved partitioning the data for each '**random_state**', training the ML models, generating predictions and evaluating the outcomes, with each accuracy metric stored for in-depth analysis. This exhaustive assessment of ten unique configurations provided a richer understanding of the model's stability and versatility.

In summary, the study is unique in several ways. First, the most important feature extraction methods for phishing email detection, including TF-IDF, DTM, FastText embedding, and CharEmbedding, at both the word and character levels, were comprehensively evaluated. Second, a balanced dataset representative of real-world scenarios was utilized, which included phishing emails written in both Arabic and English, addressing the limitations of previous studies. Finally, ten different seed values were used to train and test the model, resulting in varied accuracy results compared to previous studies, which facilitated a deeper understanding of the model's performance across diverse setups.

Table 5.18 Comparison of the work with other works of train dataset.

Ref.	No. of features	Features approach	Algorithm(s)	Train-test split	Best Algorithm	Corpus	Evaluation			
							Acc.	Prec.	Recall	F1
[11]	N/A	TF-IDF +NMF	RF, AB, NB, LR, KNN DT, & SVM	33% of training data for validation	DT and RF	English	0.999 0.999	0.994 1.000	1.000 0.994	0.997 0.997
[276]	N/A	Word embedding (*Word2vec*) +Neural Bag-of-ngrams	CNN, RNN, MLP, & LSTM	73% training and 27% testing	Word embedding + LSTM	English	0.991	-	-	-
[277]	N/A	Word embedding +Neural Bag-of-ngrams	FastText	80% for training and 20% for validation	FastText	English	-	0.990	0.990	0.990
[39]	30	TDM with SVD and TDM with NMF	RF, AB, NB, LR, KNN DT, & SVM	33% of the training data is used for validating	AB	English	0.997	1.000	0.977	0.988
[278]	40	TF-IDF + domain level features	NB, DT, & SVM	N/A	SVM	English	0.943	-	-	-
[279]	N/A	Word embedding (Word2vec)	CNN, RNN, & MLP	N/A	Word embedding + RNN	English	0.951	-	-	-
[280]	N/A	Word embedding + TF-IDF	H-LSTMs	N/A	H-LSTMs	English	-	0.973	0.953	0.963
[20]	N/A	Keras Word Embeddin g	CNN	N/A	Word Embedding + CNN	English	0.968	-	-	-
[112]	N/A	Word embedding (Doc2Vec) + TF-IDF	RF, AB, NB, LR, KNN DT, & SVM	N/A	SVM+ Doc2Vec	English	0.884	-	-	-
EAPD	N/A	TF-IDF, DTM, FastText embedding, and CharEmbe dding	MLP, KNN, DT, LR, SVM, RF, NB, XGBoost, CNN, LSTM, BI-LSTM, CNN-BI-LSTM, GRU, and BI-GRU.	80% for training and 20% for validation	English (SVM+ TF-IDF). Arabic (MLP + TF-IDF). Testing (MLP+ TF-IDF)	(IWSPA -AP 2018), translate d IWSPA-AP 2018). and APEC corpus	0.957 0.953 0.972	0.957 0.953 -	0.957 0.953 -	0.956 0.952 -

5.7 Summary

This chapter presents a novel approach for detecting phishing emails using ML/DL algorithms. Specifically, the proposal included the use of an EAPD model that utilizes both word-level (TF-IDF, DTM, and FastText embeddings) and character-level (CharEmbedding) CNNs to extract features from the text. Experiments were conducted on a dataset of 1258 emails in Arabic and English, with equal ratios of legitimate and phishing emails, to evaluate the performance of the EAPD model. The experiments indicate that, when using the MLP classifier combined with TF-IDF, the EAPD achieved an accuracy of 95.3 per cent on Arabic datasets. The English text, on the other hand, achieved a 95.7 per cent accuracy when paired with the SVM classifier and TF-IDF. The MLP classifier is clearly superior to other models for Arabic datasets. Furthermore, the LR model displayed a notably faster training time when used with the Arabic corpus and TF-IDF. This study presents a promising approach for detecting phishing emails in English and Arabic with high accuracy and efficiency.

Chapter Six

Conclusion and Future work

6.1 Overview

This thesis outlines the development of an extensible and generic model for designing and implementing phishing email detection using the EAPD model. The model describes a flexible and adaptable approach to identifying and detecting phishing emails in both Arabic and English. The study concludes with a comprehensive discussion of the findings and limitations of the proposed model and areas for future research.

6.2 Conclusions

Filtering phishing emails in the Arabic language is inherently challenging due to its intricate vocabulary, myriad dialects, and distinct writing system. Traditional email filtering systems, predominantly optimised for English, grapple with the nuances of Arabic, especially given the existence of synonyms, homographs, and diacritics, all of which can camouflage phishing content. The added hurdle is the limited availability of vast, quality-driven Arabic-language datasets suitable for ML training. This paucity restricts the development of advanced algorithms, amplifying the susceptibility of users to cyber threats. To bridge the resource gap for training ML models in Arabic phishing email detection, an innovative research method was employed. This involved the manual translation of around 1,258 emails from the acclaimed IWSPA-AP 2018 dataset, ensuring a significant augmentation in Arabic phishing email samples. Furthermore, in response to the dearth of multi-language phishing detection systems, the EAPD model was introduced. This model is tailored for both languages and employs potent word-level techniques such as TF-IDF and DTM, complemented by FastText embedding. To truly grasp the nuances of Arabic script, a CharEmbedding was integrated into the model.

In the quest to grasp the intricacies of phishing email detection across languages, a series of investigative queries was embarked upon. These aimed to decipher the intricate ties between linguistic assets and the precision of email detection, alongside feature extraction methodologies.

Within the ambit of the extensive research focusing on bilingual phishing email detection, the proficiency of both English and Arabic phishing detection instruments was assessed on transposed content. The study was initiated by formulating a model geared towards the identification of English phishing emails and training it using a dataset of 1,258 such emails. Subsequently, its competency was gauged on a batch of 300 emails, translated from Arabic to English, revealing a predictive capacity of 4%.

Shifting focus to Arabic, another model was sculpted, this time primed for detecting phishing emails in Arabic. The training was based on a set of 1,258 emails, which were earlier in English but translated into Arabic. This model's prowess was then tested against a collection of 300 original Arabic emails, demonstrating a predictive accuracy nearing 8%. This was achieved by leveraging an enriched set of Arabic stop words. A discernible enhancement was observed in the model's proficiency in predicting Arabic emails, underscoring the pivotal role of the refined stop word list in bolstering detection accuracy across diverse classifiers. Interestingly, the experimental forays yielded intriguing insights. Conventional ML classifiers such as Char embedding, TF-IDF, and DTM seemed largely unfazed by the enrichment of stop words. However, Deep Learning models, particularly those integrated with FastText embedding, showcased a distinct response. One of the most striking revelations was a substantial 14% fluctuation in LSTM's performance post the inclusion of the enhanced stop word list.

This study shows that the EAPD's accuracy stands at 95.3% for Arabic datasets when utilising the MLP classifier in conjunction with TF-IDF. This is closely matched by English text, which achieved a 95.7% accuracy using the SVM classifier paired with TF-IDF. Evidently, for Arabic datasets, MLP outperforms other classifiers. Additionally, when working with the Arabic corpus and TF-IDF, the LR model showcased a notably swift training duration. In conclusion, this study contributes significantly to the field of phishing email detection by presenting a comprehensive and practical approach to detecting phishing emails in both Arabic and English languages. The framework provided by the EAPD model can be extended and adapted to detect other types of online threats, such as malware, ransomware, and social engineering attacks.

6.3 Study implications

The study's findings have important implications for organisations looking to enhance their cybersecurity systems. Firstly, the EAPD model developed in this study can be used by organisations to detect phishing emails in both English and Arabic languages, protecting sensitive information, data, and systems from cyber-attacks. Secondly, the use of multiple techniques in the EAPD model can enhance the accuracy of phishing email detection, improving decision-making by managers when responding to potential threats. Thirdly, the creation of an English–Arabic parallel phishing email corpus can help organisations train models and improve their detection capabilities in the Arabic language, where there is a lack of available datasets, thus promoting better preparedness in identifying and responding to phishing attacks in the Arabic-speaking region. Fourthly, the high accuracy rate achieved by the EAPD model in detecting phishing emails can reduce the risk of potential data breaches and financial losses for organisations, thereby enhancing their reputation and building trust with customers. Additionally, the use of an LR classifier with TF-IDF in the EAPD model can lead to shorter training times, which can help organisations develop models faster and respond more quickly to potential threats. Finally, the findings of this study can be used for other applications beyond phishing email detection, such as detecting spam emails, social engineering attacks, and other forms of cyber-attacks, which could help to build more robust and effective cybersecurity systems for organisations.

6.4 Limitations and future work

Previous research has focused on identifying phishing emails based on specific sections of the email, such as the subject [21], [33], herder, [48], [145], [183], [238] sender (from) [373], URL [30], [38], [374], or attachments [132], [375]. However, analysing the entire email content for phishing indicators has been largely unexplored due to its complexity. Despite its potential benefits, comprehensive analysis of email content presents several challenges, such as the large amount of data involved and the need for sophisticated ML algorithms. The casualness of composing emails, as well as developing the structure of written language, which integrates both official and casual language, adds to the difficulty of identifying phishing emails. This makes it difficult to leverage NLP to analyse phishing emails with poor grammar, and misspellings might damage both the named entity (NE) method and the discovery of synonyms. Furthermore, while

the email may include words from a variety of languages, most of the studies conducted involve the use of a single language to evaluate any email.

One of the limitations of this study is that it only uses two languages in the email even though incorporating other languages could yield a better outcome. Furthermore, transcribing the pronunciation of an English word in the Arabic language has an impact on the classifier's performance. Because the Arabic word has distinct spelling, it can be recognised in two versions: one by its English pronunciation and one by its Arabic pronunciation. This problem can be addressed by creating a system that can fix or transform words depending on how they are pronounced. In addition, the approach of creating a multidimensional model for detecting phishing emails can be employed to create a model for detecting terrorist networks, phisher networks, or phishers who use other languages. All these illicit activities follow distinct behavioural patterns that may be detected if examined properly.

The meaning of a word can change dramatically depending on the perspective in which it is used. Consequently, it is conceivable that some words have more than one meaning. When those words are translated from one language into another, the difficulty becomes much more evident. As a result, it is necessary to remove the confusing words as and when they arise during translation. In a broad sense, word translation disambiguation (WSD) is the practice of finding the correct meaning of an ambiguous word based on the context in which it appears. This can be summarised as the correlation of an ambiguous word's appearance with one of its correct meanings.

In Arabic, diacritization (also known as vocalisation or voweling) is the process of placing a symbol over and underlining letters that are intended to emphasise proper pronunciation and disambiguation. The lack of diacritization in Arabic texts in Arabic printed media or online sites presents a significant problem for both NLP and translation, resulting in a high level of ambiguity. As a result, the probability of a single word having numerous meanings is considerably greater. For instance, the Arabic word **" قطر "** (**قَطَر, قَطَرَ, قُطُر, قَطْر, قُطر, قَطَّرَ**) can have the following translations in English: Qatar-country, tow, region, tug, dropping, territory, rain, land. Similarly, the Arabic word "**عين**" (**عيّن, عَيْنٌ, عُيُن, عَيَن, عَين, عَيَّنَ, عَينَ**) could result in the following translations: eye, peeper, optic, spy, assistance, specify, appoint, assign, designate, post, put, allocate. The

process of disambiguation entails two steps: 1) determine all pertinent meanings for each word; 2) attribute the right meaning to that every time it appears. For the first phase, a set of senses for each of the ambiguous words found in the daily lexicon might be used. While there is still some doubt, the second stage can be completed by examining the perspective in which the ambiguous word appears, or by consulting an additional knowledge source, such as lexical resources. It is critical to analyse the origin of information for disambiguation, develop principles utilising this information, and identify the grounds for choosing the optimum meaning for an ambiguous word. WSD techniques can be classed into three groups in terms of ML: supervised learning, unsupervised learning, and hybrids of the two.

One of the challenges faced in this study relates to the unique nature of the Arabic language. As Arabic is a right-to-left language, traditional tokenization and processing techniques may not maintain the semantic essence of the text. Although tokenized sequences of Arabic maintain the ordering of tokens similar to other languages, it's crucial to acknowledge that the order and importance of sequences could differ from other left-to-right languages. The pseudocode provided in the research didn't specifically account for this directionality when it came to padding and truncating sequences. To better cater to the right-to-left nature of the Arabic language, future research could adapt the pre-processing step to ensure that truncating and padding occur at the beginning of sequences.

6.5 Recommendations

The uniqueness of the approach proposed in this study lies in its foundation on logical principles, aiming to enhance the logic of treatments. This is significant as it allows the detection of phishing emails without completing the entire process under certain circumstances. The study introduces a new hybrid method for identifying and resisting phishing attacks, emphasizing its applicability, accuracy, and reliability. The researchers recommend a comprehensive procedure to update the model and extract new knowledge about the evolving nature of phishing. The method facilitates dynamic feature extraction, enabling valuable classification results with only a subset of all features. The research showcases feature classification using both machine learning (ML) and deep learning (DL) models simultaneously, specifically at the word-level (FastText embedding) and CharEmbedding. Despite the resource-intensive nature of DL networks due to extensive

mathematical calculations, they have demonstrated exceptional problem-solving capacity, particularly in dealing with large datasets, commonly seen in computer vision tasks. In the realm of binary classification for phishing email detection, DL models, especially Convolutional Neural Networks (CNN), exhibit not only excellent accuracy and performance but also contribute to reducing the error rate. The combination of six different algorithm methods in the current work represents a substantial contribution, introducing a novel methodology.

For future endeavours in phishing email detection, especially in English and Arabic, integrating Word2Vec with BERT is suggested. This combination leverages Word2Vec's effectiveness in capturing semantic relationships and BERT's advanced capabilities in understanding context and language nuances, potentially enhancing detection accuracy. This approach is particularly beneficial in discerning subtle phishing tactics in both languages involving sophisticated linguistic manipulations. Investigating this hybrid model could lead to breakthroughs in more effectively identifying and countering phishing threats in multilingual contexts, with no similar study undertaken to date. Given the inclusion of various features in the study, conducting a stress test on emails using a distinct approach of analysis is recommended over relying on a single mode, such as lexical or blacklist. Scaling up this method for production use in the future is suggested. Based on the study's findings and the preceding discussion, the proposed approach can be inferred to serve as a reliable solution for batch and offline use. Future researchers are encouraged to expand their understanding by addressing key questions such as:

(i) What constitutes the most effective minimum set of features that can be employed to predict a phishing email?
(ii) How can extensive data (in terms of volume, veracity, and velocity) be harnessed and integrated into deep learning models?
(iii) In what ways can the project be extended to address additional forms of attacks, particularly those involving malware attachments in emails?
(iv) What potential strategies exist to enhance performance by mitigating zero-hour attacks and proactively preventing attacks before they occur?

References

[1] "Anti-Phishing Working Group," *Phishing Activity Trends Report 3rd Quarter 2022*. [Online]. Available: https://docs.apwg.org/reports/apwg_trends_report_q3_2022.pdf?_ga=2.41334753.2114145525.1680386102-1522170639.1680386102&_gl=1*1ekzzva*_ga*MTUyMjE3MDYzOS4xNjgwMzg2MTAy*_ga_55RF0RHXSR*MTY4MDM4NjEwMi4xLjEuMTY4MDM4NjcyNC4wLjAuMA.

[2] "Anti-Phishing Working Group.," *Phishing Activity Trends Report 3rd Quarter 2020.*, 2020. [Online]. Available: https://docs.apwg.org/reports/apwg_trends_report_q3_2020.pdf.

[3] U. A. Butt, R. Amin, H. Aldabbas, S. Mohan, B. Alouffi, and A. Ahmadian, "Cloud-based email phishing attack using machine and deep learning algorithm," *Complex Intell. Syst.*, vol. 9, no. 3, pp. 3043–3070, 2023.

[4] L. Ribeiro, I. S. Guedes, and C. S. Cardoso, "Which factors predict susceptibility to phishing? An empirical study," *Comput. Secur.*, vol. 136, p. 103558, 2024.

[5] J. Doshi, K. Parmar, R. Sanghavi, and N. Shekokar, "A comprehensive dual-layer architecture for phishing and spam email detection," *Comput. Secur.*, vol. 133, p. 103378, 2023.

[6] R. Eckhardt and S. Bagui, "A User-centric Focus for Detecting Phishing Emails," *AI, Mach. Learn. Deep Learn. A Secur. Perspect.*, 2023.

[7] J. Clark, I. Koprinska, and J. Poon, "A neural network based approach to automated e-mail classification," in *Proceedings IEEE/WIC International Conference on Web Intelligence (WI 2003)*, 2003, pp. 702–705.

[8] V. Shukla and M. K. Rai, "A rule based approach for detecting phishing attacks," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 2, no. 9, pp. 92–100, 2014.

[9] M. Foozy, C. Feresa, R. Ahmad, and M. F. Abdollah, "A practical rule based technique by splitting SMS phishing from SMS spam for better accuracy in mobile device," *Int. Rev. Comput. Softw.*, vol. 9, no. 10, pp. 1776–1782, 2014.

[10] M. Pandey and V. Ravi, "Detecting phishing e-mails using text and data mining," in *2012 IEEE International Conference on Computational Intelligence and Computing Research*, 2012, pp. 1–6.

[11] N. B. Harikrishnan, R. Vinayakumar, and K. P. Soman, "A machine learning approach towards phishing email detection," in *Proceedings of the Anti-Phishing Pilot at ACM International Workshop on Security and Privacy Analytics (IWSPA AP)*, 2018, vol. 2013, pp. 455–468.

[12] J. Zhang, W. Li, L. Gong, Z. Gu, and J. Wu, "Targeted Malicious Email Detection Using Hypervisor-Based Dynamic Analysis and Ensemble Learning," in *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.

[13] A. Junnarkar, S. Adhikari, J. Fagania, P. Chimurkar, and D. Karia, "E-Mail Spam Classification via Machine Learning and Natural Language Processing," in *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, 2021, pp. 693–699.

[14] E. S. Gualberto, R. T. De Sousa, P. D. B. Thiago, J. P. C. L. Da Costa, and C. G. Duque, "The Answer is in the Text: Multi-Stage Methods for Phishing Detection based on Feature

Engineering," *IEEE Access*, 2020.
[15] M. U. Chowdhury, J. H. Abawajy, A. V Kelarev, and T. Hochin, "Multilayer hybrid strategy for phishing email zero- day filtering," *Concurr. Comput. Pract. Exp.*, vol. 29, no. 23, p. e3929, 2017.
[16] A. Kumar, J. M. Chatterjee, and V. G. Díaz, "A novel hybrid approach of SVM combined with NLP and probabilistic neural network for email phishing," *Int. J. Electr. Comput. Eng.*, vol. 10, no. 1, p. 486, 2020.
[17] X. Li, D. Zhang, and B. Wu, "Detection method of phishing email based on persuasion principle," in *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, 2020, vol. 1, pp. 571–574.
[18] E. Castillo, S. Dhaduvai, P. Liu, K.-S. Thakur, A. Dalton, and T. Strzalkowski, "Email Threat Detection Using Distinct Neural Network Approaches," in *Proceedings for the First International Workshop on Social Threats in Online Conversations: Understanding and Management*, 2020, pp. 48–55.
[19] R. Vinayakumar, K. P. Soman, P. Poornachandran, V. S. Mohan, and A. D. Kumar, "ScaleNet: scalable and hybrid framework for cyber threat situational awareness based on DNS, URL, and email data analysis," *J. Cyber Secur. Mobil.*, pp. 189–240, 2018.
[20] M. Hiransha, N. A. Unnithan, R. Vinayakumar, K. Soman, and A. D. R. Verma, "Deep learning based phishing e-mail detection," in *Proc. 1st AntiPhishing Shared Pilot 4th ACM Int. Workshop Secur. Privacy Anal.(IWSPA)*, 2018.
[21] R. Alotaibi, I. Al-Turaiki, and F. Alakeel, "Mitigating Email Phishing Attacks using Convolutional Neural Networks," in *2020 3rd International Conference on Computer Applications & Information Security (ICCAIS)*, 2020, pp. 1–6.
[22] M. Hamisu and A. Mansour, "Detecting Advance Fee Fraud using NLP Bag of word Model," in *2020 IEEE 2nd International Conference on Cyberspac (CYBER NIGERIA)*, 2021, pp. 94–97.
[23] M. S. Swetha and G. Sarraf, "Spam Email and Malware Elimination employing various Classification Techniques," in *2019 4th International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT)*, 2019, pp. 140–145.
[24] J. Rastenis, S. Ramanauskaitė, I. Suzdalev, K. Tunaitytė, J. Janulevičius, and A. Čenys, "Multi-Language Spam/Phishing Classification by Email Body Text: Toward Automated Security Incident Investigation," *Electronics*, vol. 10, no. 6, p. 668, 2021.
[25] V. D. Sharma, S. K. Yadav, S. K. Yadav, K. N. Singh, and S. Sharma, "An effective approach to protect social media account from spam mail–A machine learning approach," *Mater. Today Proc.*, 2021.
[26] P. Verma, A. Goyal, and Y. Gigras, "Email phishing: text classification using natural language processing," *Comput. Sci. Inf. Technol.*, vol. 1, no. 1, pp. 1–12, 2020.
[27] W. Niu, X. Zhang, G. Yang, Z. Ma, and Z. Zhuo, "Phishing emails detection using CS-SVM," in *2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC)*, 2017, pp. 1054–1059.
[28] Q. Yaseen, "Spam Email Detection Using Deep Learning Techniques," *Procedia Comput. Sci.*, vol. 184, pp. 853–858, 2021.
[29] Y. Lee, J. Saxe, and R. Harang, "CATBERT: Context-Aware Tiny BERT for Detecting Social Engineering Emails," *arXiv Prepr. arXiv2010.03484*, 2020.
[30] R. Vinayakumar, K. P. Soman, P. Poornachandran, S. Akarsh, and M. Elhoseny, "Deep

learning framework for cyber threat situational awareness based on email and url data analysis," in *Cybersecurity and Secure Information Systems*, Springer, 2019, pp. 87–124.
[31] G. Egozi and R. Verma, "Phishing email detection using robust nlp techniques," in *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, 2018, pp. 7–12.
[32] F. Janjua, A. Masood, H. Abbas, and I. Rashid, "Handling Insider Threat Through Supervised Machine Learning Techniques," *Procedia Comput. Sci.*, vol. 177, pp. 64–71, 2020.
[33] E. M. Bahgat, S. Rady, W. Gad, and I. F. Moawad, "Efficient email classification approach based on semantic methods," *Ain Shams Eng. J.*, vol. 9, no. 4, pp. 3259–3269, 2018.
[34] U. Ы. Malaysia, "AN ENHANCED ONLINE PHISHING E-MAIL DETECTION FRAMEWORK BASED ON" EVOLVING CONNECTIONIST SYSTEM."
[35] A. Yasin and A. Abuhasan, "An intelligent classification model for phishing email detection," *arXiv Prepr. arXiv1608.02196*, 2016.
[36] T. Gangavarapu and C. D. Jaidhar, "A novel bio-inspired hybrid metaheuristic for unsolicited bulk email detection," in *International Conference on Computational Science*, 2020, pp. 240–254.
[37] S. Gibson, B. Issac, L. Zhang, and S. M. Jacob, "Detecting Spam Email with Machine Learning Optimized with Bio-Inspired Meta-Heuristic Algorithms," *IEEE Access*, 2020.
[38] R. Banu, M. Anand, A. Kamath, S. Ashika, H. S. Ujwala, and S. N. Harshitha, "Detecting Phishing Attacks Using Natural Language Processing And Machine Learning," in *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, 2019, pp. 1210–1214.
[39] A. Vazhayil, N. B. Harikrishnan, R. Vinayakumar, K. P. Soman, and A. D. R. Verma, "PED-ML: Phishing email detection using classical machine learning techniques," in *Proc. 1st AntiPhishing Shared Pilot 4th ACM Int. Workshop Secur. Privacy Anal.(IWSPA)*, 2018, pp. 1–8.
[40] M. Arshey and A. V. KS, "An optimization-based deep belief network for the detection of phishing e-mails," *Data Technol. Appl.*, 2020.
[41] S. Sankhwar, D. Pandey, and R. A. Khan, "Email phishing: an Enhanced classification model to detect malicious URLs," *EAI Endorsed Trans. Scalable Inf. Syst.*, vol. 6, no. 21, 2019.
[42] N. Moradpoor, B. Clavie, and B. Buchanan, "Employing machine learning techniques for detection and classification of phishing emails," in *2017 Computing Conference*, 2017, pp. 149–156.
[43] E. M. Rudd, R. Harang, and J. Saxe, "Meade: Towards a malicious email attachment detection engine," in *2018 IEEE International Symposium on Technologies for Homeland Security (HST)*, 2018, pp. 1–7.
[44] V. S. Vinitha and D. K. Renuka, "Performance Analysis of E-Mail Spam Classification using different Machine Learning Techniques," in *2019 International Conference on Advances in Computing and Communication Engineering (ICACCE)*, 2019, pp. 1–5.
[45] C. Thapa *et al.*, "Evaluation of Federated Learning in Phishing Email Detection."
[46] L. Halgaš, I. Agrafiotis, and J. R. C. Nurse, "Catching the Phish: Detecting Phishing Attacks using Recurrent Neural Networks (RNNs)," in *International Workshop on Information Security Applications*, 2019, pp. 219–233.
[47] A. Baccouche, S. Ahmed, D. Sierra-Sosa, and A. Elmaghraby, "Malicious Text Identification: Deep Learning from Public Comments and Emails," *Information*, vol. 11,

no. 6, p. 312, 2020.
[48] Y. Fang, C. Zhang, C. Huang, L. Liu, and Y. Yang, "Phishing email detection using improved RCNN model with multilevel vectors and attention mechanism," *IEEE Access*, vol. 7, pp. 56329–56340, 2019.
[49] D. Xiao and M. Jiang, "Malicious Mail Filtering and Tracing System Based on KNN and Improved LSTM Algorithm," in *2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, 2020, pp. 222–229.
[50] V. Listık, Š. Let, J. Šedivý, and V. Hlavác, "Phishing Email Detection based on Named Entity Recognition," 2019.
[51] G. Sonowal, "A Model for Detecting Sounds-alike Phishing Email Contents for Persons with Visual Impairments," in *2020 Sixth International Conference on e-Learning (econf)*, 2020, pp. 17–21.
[52] M. Iwanaga, T. Tabata, and K. Sakurai, "Some fitting of naive Bayesian spam filtering for Japanese environment," in *International Workshop on Information Security Applications*, 2004, pp. 135–143.
[53] J. Dong, H. Cao, P. Liu, and L. Ren, "Bayesian Chinese spam filter based on crossed N-gram," in *Sixth International Conference on Intelligent Systems Design and Applications*, 2006, vol. 3, pp. 103–108.
[54] E. E. Eryılmaz, D. Ö. Şahin, and E. Kılıç, "Filtering turkish spam using LSTM from deep learning techniques," in *2020 8th International Symposium on Digital Forensics and Security (ISDFS)*, 2020, pp. 1–6.
[55] M. TURHANLAR, "DETECTING TURKISH PHISHING ATTACKS WITH MACHINE LEARNING CLASSIFIERS." MIDDLE EAST TECHNICAL UNIVERSITY, 2019.
[56] C. Özdemir, M. Ataş, and A. B. Özer, "Classification of Turkish spam e-mails with artificial immune system," in *2013 21st Signal Processing and Communications Applications Conference (SIU)*, 2013, pp. 1–4.
[57] E. E. Eryilmaz, D. O. Şahin, and E. Kılıç, "Machine Learning Based Spam E-mail Detection System for Turkish," in *2020 5th International Conference on Computer Science and Engineering (UBMK)*, 2020, pp. 7–12.
[58] L. Ermakova, "Spam and phishing detection in various languages," in *Обработка текста и когнитивные технологии: Когнитивное моделирование в лингвистике: тр. XII Междунар. науч. конф. Казань: Изд-во Казан. ун-та*, 2010, pp. 189–193.
[59] S. Salloum, T. Gaber, S. Vadera, and K. Sharan, "A Systematic Literature Review on Phishing Email Detection Using Natural Language Processing Techniques," *IEEE Access*, 2022.
[60] H. T. Himdi, "Classification of Arabic real and fake news based on Arabic textual analysis," 2022.
[61] "Gulf Business," 2020. [Online]. Available: https://gulfbusiness.com/saudi-arabia-led-gcc-in-number-of-phishing-attacks-in-q2-kaspersky-report/.
[62] P. Burda, L. Allodi, and N. Zannone, "Don't forget the human: a crowdsourced approach to automate response and containment against spear phishing attacks," in *2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, 2020, pp. 471–476.
[63] S. Salloum, T. Gaber, S. Vadera, and K. Shaalan, "A New English/Arabic Parallel Corpus

for Phishing Emails," *ACM Trans. Asian Low-Resource Lang. Inf. Process.*, 2023.
[64] K. Versteegh, *Arabic language*. Edinburgh University Press, 2014.
[65] A. M. El-Halees, "Filtering Spam E-Mail from Mixed Arabic and English Messages: A Comparison of Machine Learning Techniques.," *Int. Arab J. Inf. Technol.*, vol. 6, no. 1, 2009.
[66] H. Al-Mohannadi, I. Awan, J. Al Hamar, Y. Al Hamar, M. Shah, and A. Musa, "Understanding awareness of cyber security threat among IT employees," in *2018 6th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*, 2018, pp. 188–192.
[67] M. M. Ahmed Baiomy and A. Youssif, "Anti-Phishing Game Framework to Educate Arabic Users: Avoidance of URLs Phishing Attacks," *Indian J. Sci. Technol.*, vol. 12, p. 44, 2019.
[68] Y. A. Younis and M. Musbah, "A Framework to Protect Against Phishing Attacks," in *Proceedings of the 6th International Conference on Engineering & MIS 2020*, 2020, pp. 1–6.
[69] H. Ahmad and L. Erdodi, "Overview of phishing landscape and homographs in Arabic domain names," *Secur. Priv.*, p. e159, 2021.
[70] I. Alseadoon, T. Chan, E. Foo, and J. Gonzalez Nieto, "Who is more susceptible to phishing emails?: a Saudi Arabian study," 2012.
[71] M. I. Alwanain, "Effects of User-Awareness on the Detection of Phishing Emails: A Case Study," *Int. J. Innov. Technol. Explor. Eng*, vol. 8, no. 4, pp. 480–484, 2019.
[72] M. Alwanain, "An Evaluation of User Awareness for the Detection of Phishing Emails," *Int. J. Adv. Comput. Sci. Appl*, vol. 10, no. 10, pp. 238–323, 2019.
[73] S. Salloum, T. Gaber, S. Vadera, and K. Shaalan, "Phishing Email Detection Using Natural Language Processing Techniques: A Literature Survey," *Procedia Comput. Sci.*, vol. 189, pp. 19–28, 2021.
[74] S. Salloum, T. Gaber, S. Vadera, and K. Shaalan, "Phishing Website Detection from URLs Using Classical Machine Learning ANN Model," in *International Conference on Security and Privacy in Communication Systems*, 2021, pp. 509–523.
[75] N. Marangunić and A. Granić, "Technology acceptance model: a literature review from 1986 to 2013," *Univers. Access Inf. Soc.*, vol. 14, no. 1, pp. 81–95, 2015.
[76] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," 2007.
[77] D. Moher, A. Liberati, J. Tetzlaff, and D. G. Altman, "Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement," *Ann. Intern. Med.*, vol. 151, no. 4, pp. 264–269, 2009.
[78] V. Costa and S. Monteiro, "Knowledge processes, absorptive capacity and innovation: A mediation analysis," *Knowl. Process Manag.*, vol. 23, no. 3, pp. 207–218, 2016.
[79] D. Moher, A. Liberati, J. Tetzlaff, and D. G. Altman, "Preferred Reporting Items for Systematic Reviews and Meta-Analyses: The PRISMA Statement," *PLoS Med.*, vol. 6, no. 7, p. e1000097, Jul. 2009.
[80] M. Al-Emran, V. Mezhuyev, A. Kamaludin, and K. Shaalan, "The impact of knowledge management processes on information systems: A systematic review," *Int. J. Inf. Manage.*, vol. 43, pp. 173–187, 2018.
[81] S. Kitchenham, B, Charters, "Guidelines for performing systematic literature reviews in software engineering.," *Softw. Eng. Group, Sch. Comput. Sci. Math. Keele Univ. 1–57.*, 2007.

[82] "PhishTank: An Anti-Phishing Site, LLC OpenDNS, San Francisco, CA, USA, accessed: Dec. 5, 2016." .
[83] "APWG Phishing Trends Reports, Anti Phishing Working Group," 2016.
[84] G. Xiang, J. Hong, C. P. Rose, and L. Cranor, "Cantina+ a feature-rich machine learning framework for detecting phishing web sites," *ACM Trans. Inf. Syst. Secur.*, vol. 14, no. 2, pp. 1–28, 2011.
[85] G. Ramesh, I. Krishnamurthi, and K. S. S. Kumar, "An efficacious method for detecting phishing webpages through target domain identification," *Decis. Support Syst.*, vol. 61, pp. 12–22, 2014.
[86] Z. Dou, I. Khalil, A. Khreishah, A. Al-Fuqaha, and M. Guizani, "Systematization of Knowledge (SoK): A systematic review of software-based web phishing detection," *IEEE Commun. Surv. Tutorials*, vol. 19, no. 4, pp. 2797–2819, 2017.
[87] C. Karlof, U. Shankar, J. D. Tygar, and D. Wagner, "Dynamic pharming attacks and locked same-origin policies for web browsers," in *Proceedings of the 14th ACM conference on Computer and communications security*, 2007, pp. 58–71.
[88] Z. Dou, *Secure entity authentication*. New Jersey Institute of Technology, 2018.
[89] C. Whittaker, B. Ryner, and M. Nazif, "Large-scale automatic classification of phishing pages," 2010.
[90] M. Khonji, Y. Iraqi, and A. Jones, "Phishing detection: a literature survey," *IEEE Commun. Surv. Tutorials*, vol. 15, no. 4, pp. 2091–2121, 2013.
[91] E. E. H. Lastdrager, "Achieving a consensual definition of phishing based on a systematic review of the literature," *Crime Sci.*, vol. 3, no. 1, pp. 1–10, 2014.
[92] "American Heritage Dictionary of the English Language," *Fifth Ed.*, 2011.
[93] APWG, "Anti-Phishing Working Group Phishing Archive," 2014. .
[94] "Collins English Dictionary Phishing," 2013. .
[95] "Merriam-Webster Phishing," 2013. [Online]. Available: http://www.merriam-webster.com/%0Adictionary/phishing.
[96] "Oxford University Press," 2014. .
[97] "PhishTank," 2012. [Online]. Available: www.phishtank.com.
[98] A. Emigh, "The crimeware landscape: Malware, phishing, identity theft and beyond," *J. Digit. Forensic Pract.*, vol. 1, no. 3, pp. 245–260, 2006.
[99] S. Abraham and I. Chengalur-Smith, "An overview of social engineering malware: Trends, tactics, and implications," *Technol. Soc.*, vol. 32, no. 3, pp. 183–196, 2010.
[100] G. Park and J. Rayz, "Ontological detection of phishing emails," in *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2018, pp. 2858–2863.
[101] G. M. Megaw, "Phishing within e-commerce: reducing the risk, increasing the trust," 2010.
[102] G. Park, "Text-based phishing detection using a simulation model," 2013.
[103] D. Robson, "A Brief History of Phishing.," 2011.
[104] J. Milletary and C. C. Center, "Technical trends in phishing attacks," *Retrieved December*, vol. 1, no. 2007, p. 3, 2005.
[105] APWG., "Origins of the Word 'Phishing,'" 2004. [Online]. Available: http://docs.apwg.org/word_phish.html.
[106] "University of Massachusetts Amherst," 2020. [Online]. Available: https://www.umass.edu/it/freshphish.
[107] H. Kim and J. H. Huh, "Detecting DNS-poisoning-based phishing attacks from their network performance characteristics," *Electron. Lett.*, vol. 47, no. 11, pp. 656–658, 2011.

[108] B. B. Gupta and Q. Z. Sheng, *Machine learning for computer and cyber security: principle, algorithms, and practices*. CRC Press, 2019.
[109] A. Almomani, B. B. Gupta, S. Atawneh, A. Meulenberg, and E. Almomani, "A survey of phishing email filtering techniques," *IEEE Commun. Surv. tutorials*, vol. 15, no. 4, pp. 2070–2090, 2013.
[110] E. Blanzieri and A. Bryl, "A survey of learning-based techniques of email spam filtering," *Artif. Intell. Rev.*, vol. 29, no. 1, pp. 63–92, 2008.
[111] R. Dhamija, J. D. Tygar, and M. Hearst, "Why phishing works," in *Proceedings of the SIGCHI conference on Human Factors in computing systems*, 2006, pp. 581–590.
[112] N. A. Unnithan, N. B. Harikrishnan, R. Vinayakumar, K. P. Soman, and S. Sundarakrishna, "Detecting phishing E-mail using machine learning techniques," in *Proc. 1st Anti-Phishing Shared Task Pilot 4th ACM IWSPA Co-Located 8th ACM Conf. Data Appl. Secur. Privacy (CODASPY)*, 2018, pp. 51–54.
[113] R. Dazeley, J. L. Yearwood, B. H. Kang, and A. V Kelarev, "Consensus clustering and supervised classification for profiling phishing emails in internet commerce security," in *Pacific Rim Knowledge Acquisition Workshop*, 2010, pp. 235–246.
[114] S. Seifollahi, A. Bagirov, R. Layton, and I. Gondal, "Optimization based clustering algorithms for authorship analysis of phishing emails," *Neural Process. Lett.*, vol. 46, no. 2, pp. 411–425, 2017.
[115] T. Stojnic, D. Vatsalan, and N. A. G. Arachchilage, "Phishing email strategies: Understanding cybercriminals' strategies of crafting phishing emails," *Secur. Priv.*, p. e165.
[116] R. Basnet, S. Mukkamala, and A. H. Sung, "Detection of phishing attacks: A machine learning approach," in *Soft computing applications in industry*, Springer, 2008, pp. 373–383.
[117] P. R. K. Prosun, K. S. Alam, and S. Bhowmik, "Improved Spam Email Filtering Architecture Using Several Feature Extraction Techniques," in *Proceedings of the International Conference on Big Data, IoT, and Machine Learning*, 2022, pp. 665–675.
[118] M. MANASWINI and D. R. N. SRINIVASU, "Phishing Email Detection Model using Improved Recurrent Convolutional Neural Networks and Multilevel Vectors," *Ann. Rom. Soc. Cell Biol.*, vol. 25, no. 6, pp. 16674–16681, 2021.
[119] J. Lee, F. Tang, P. Ye, F. Abbasi, P. Hay, and D. M. Divakaran, "D-Fence: A Flexible, Efficient, and Comprehensive Phishing Email Detection System," in *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*, 2021, pp. 578–597.
[120] W. Pan *et al.*, "Semantic Graph Neural Network: A Conversion from Spam Email Classification to Graph Classification," *Sci. Program.*, vol. 2022, 2022.
[121] M. Crawford, T. M. Khoshgoftaar, J. D. Prusa, A. N. Richter, and H. Al Najada, "Survey of review spam detection using machine learning techniques," *J. Big Data*, vol. 2, no. 1, p. 23, 2015.
[122] A. A. Akinyelu and A. O. Adewumi, "Classification of phishing email using random forest machine learning technique," *J. Appl. Math.*, vol. 2014, 2014.
[123] I. R. A. Hamid and J. H. Abawajy, "Profiling phishing email based on clustering approach," in *2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, 2013, pp. 628–635.
[124] I. R. A. Hamid and J. Abawajy, "Phishing email feature selection approach," in *2011IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications*, 2011, pp. 916–921.

[125] W. N. Gansterer and D. Pölz, "E-mail classification for phishing defense," in *European Conference on Information Retrieval*, 2009, pp. 449–460.
[126] L. F. Gutiérrez, F. Abri, M. Armstrong, A. S. Namin, and K. S. Jones, "Phishing Detection through Email Embeddings," *arXiv Prepr. arXiv2012.14488*, 2020.
[127] L. Ma, B. Ofoghi, P. Watters, and S. Brown, "Detecting phishing emails using hybrid features," in *2009 Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing*, 2009, pp. 493–497.
[128] S. Smadi, N. Aslam, L. Zhang, R. Alasem, and M. A. Hossain, "Detection of phishing emails using data mining algorithms," in *2015 9th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)*, 2015, pp. 1–8.
[129] G. Yu, W. Fan, W. Huang, and J. An, "An Explainable Method of Phishing Emails Generation and Its Application in Machine Learning," in *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, 2020, vol. 1, pp. 1279–1283.
[130] E. Marková, T. Bajtoš, P. Sokol, and T. Mézešová, "Classification of malicious emails," in *2019 IEEE 15th International Scientific Conference on Informatics*, 2019, pp. 279–284.
[131] R. Amin, M. M. Rahman, and N. Hossain, "A Bangla Spam Email Detection and Datasets Creation Approach based on Machine Learning Algorithms," in *2019 3rd International Conference on Electrical, Computer & Telecommunication Engineering (ICECTE)*, 2019, pp. 169–172.
[132] Y. Cohen, D. Hendler, and A. Rubin, "Detection of malicious webmail attachments based on propagation patterns," *Knowledge-Based Syst.*, vol. 141, pp. 67–79, 2018.
[133] S. Maldonado and G. L'Huillier, "SVM-based feature selection and classification for email filtering," in *Pattern recognition-applications and methods*, Springer, 2013, pp. 135–148.
[134] M. A. Mohammed, S. S. Gunasekaran, S. A. Mostafa, A. Mustafa, and M. K. Abd Ghani, "Implementing an agent-based multi-natural language anti-spam model," in *2018 International Symposium on Agent, Multi-Agent Systems and Robotics (ISAMSR)*, 2018, pp. 1–5.
[135] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
[136] T. M. Mitchell, "Machine learning," 1997.
[137] F. Toolan and J. Carthy, "Phishing detection using classifier ensembles," in *2009 eCrime Researchers Summit*, 2009, pp. 1–9.
[138] R. Islam and J. Abawajy, "A multi-tier phishing detection and filtering approach," *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 324–335, 2013.
[139] L. F. Gutiérrez, F. Abri, M. Armstrong, A. S. Namin, and K. S. Jones, "Email Embeddings for Phishing Detection," in *2020 IEEE International Conference on Big Data (Big Data)*, 2020, pp. 2087–2092.
[140] S. K. Sonbhadra, S. Agarwal, M. Syafrullah, and K. Adiyarta, "Email classification via intention-based segmentation," in *2020 7th International Conference on Electrical Engineering, Computer Sciences and Informatics (EECSI)*, 2020, pp. 38–44.
[141] E.-S. M. El-Alfy and R. E. Abdel-Aal, "Using GMDH-based networks for improved spam detection and email feature analysis," *Appl. Soft Comput.*, vol. 11, no. 1, pp. 477–488, 2011.
[142] M. A. Mohammed, D. A. Ibrahim, and A. O. Salman, "Adaptive intelligent learning approach based on visual anti-spam email model for multi-natural language," *J. Intell. Syst.*, vol. 30, no. 1, pp. 774–792, 2021.
[143] M. Sethi, S. Chandra, V. Chaudhary, and Y. Dahiya, "Spam Email Detection Using

Machine Learning and Neural Networks," in *Sentimental Analysis and Deep Learning*, Springer, 2022, pp. 275–290.
[144] G. Stringhini and O. Thonnard, "That ain't you: Blocking spearphishing through behavioral modelling," in *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, 2015, pp. 78–97.
[145] S. Duman, K. Kalkan-Cakmakci, M. Egele, W. Robertson, and E. Kirda, "Emailprofiler: Spearphishing filtering with header and stylometric features of emails," in *2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC)*, 2016, vol. 1, pp. 408–416.
[146] A. Bergholz, J. H. Chang, G. Paass, F. Reichartz, and S. Strobel, "Improved Phishing Detection using Model-Based Features.," in *CEAS*, 2008.
[147] I. Fette, N. Sadeh, and A. Tomasic, "Learning to detect phishing emails," in *Proceedings of the 16th international conference on World Wide Web*, 2007, pp. 649–656.
[148] S. R. Mirhoseini, F. Vahedi, and J. A. Nasiri, "E-Mail phishing detection using natural language processing and machine learning techniques."
[149] C. E. Shyni, S. Sarju, and S. Swamynathan, "A multi-classifier based prediction model for phishing emails detection using topic modelling, named entity recognition and image processing," *Circuits Syst.*, vol. 7, no. 09, p. 2507, 2016.
[150] R. B. Basnet and A. H. Sung, "Classifying phishing emails using confidence-weighted linear classifiers," in *International Conference on Information Security and Artificial Intelligence (ISAI)*, 2010, pp. 108–112.
[151] F. Sanchez and Z. Duan, "A sender-centric approach to detecting phishing emails," in *2012 International Conference on Cyber Security*, 2012, pp. 32–39.
[152] L. M. Stuart, G. Park, J. M. Talor, and V. Raskin, "On identifying phishing emails: Uncertainty in machine and human judgment," in *2014 IEEE Conference on Norbert Wiener in the 21st Century (21CW)*, 2014, pp. 1–8.
[153] G. Park, L. M. Stuart, J. M. Taylor, and V. Raskin, "Comparing machine and human ability to detect phishing emails," in *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2014, pp. 2322–2327.
[154] L. M. Form, K. L. Chiew, and W. K. Tiong, "Phishing email detection technique by using hybrid features," in *2015 9th International Conference on IT in Asia (CITA)*, 2015, pp. 1–5.
[155] O. A. Adewumi and A. A. Akinyelu, "A hybrid firefly and support vector machine classifier for phishing email detection," *Kybernetes*, 2016.
[156] J. Abawajy and A. Kelarev, "A multi-tier ensemble construction of classifiers for phishing email detection and filtering," in *International Symposium on Cyberspace Safety and Security*, 2012, pp. 48–56.
[157] J. Yearwood, M. Mammadov, and D. Webb, "Profiling phishing activity based on hyperlinks extracted from phishing emails," *Soc. Netw. Anal. Min.*, vol. 2, no. 1, pp. 5–16, 2012.
[158] S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nair, "A distributed architecture for phishing detection using Bayesian additive regression trees," in *2008 eCrime Researchers Summit*, 2008, pp. 1–10.
[159] J. P. M. De Sa, *Pattern recognition: concepts, methods, and applications*. Springer Science & Business Media, 2001.
[160] M. N. Al-Mhiqani *et al.*, "A new intelligent multilayer framework for insider threat

detection," *Comput. Electr. Eng.*, p. 107597, 2021.
[161] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep learning for computer vision: A brief review," *Comput. Intell. Neurosci.*, vol. 2018, 2018.
[162] M. Hassaballah and A. I. Awad, *Deep learning in computer vision: principles and applications*. CRC Press, 2020.
[163] S. M. S. Islam, S. Rahman, M. M. Rahman, E. K. Dey, and M. Shoyaib, "Application of deep learning to computer vision: A comprehensive study," in *2016 5th international conference on informatics, electronics and vision (ICIEV)*, 2016, pp. 592–597.
[164] Y. Ning, S. He, Z. Wu, C. Xing, and L.-J. Zhang, "A review of deep learning based speech synthesis," *Appl. Sci.*, vol. 9, no. 19, p. 4050, 2019.
[165] H. Zen, "Deep learning in speech synthesis," 2013.
[166] A. B. Nassif, I. Shahin, I. Attili, M. Azzeh, and K. Shaalan, "Speech recognition using deep neural networks: A systematic review," *IEEE Access*, vol. 7, pp. 19143–19165, 2019.
[167] K. Noda, Y. Yamaguchi, K. Nakadai, H. G. Okuno, and T. Ogata, "Audio-visual speech recognition using deep learning," *Appl. Intell.*, vol. 42, no. 4, pp. 722–737, 2015.
[168] R. H. Abiyev, M. Arslan, and J. B. Idoko, "Sign language translation using deep convolutional neural networks," *KSII Trans. Internet Inf. Syst.*, vol. 14, no. 2, pp. 631–653, 2020.
[169] A. Pumsirirat and L. Yan, "Credit card fraud detection using deep learning based on auto-encoder and restricted boltzmann machine," *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no. 1, pp. 18–25, 2018.
[170] X. Zhang, Y. Han, W. Xu, and Q. Wang, "HOBA: A novel feature engineering methodology for credit card fraud detection with a deep learning architecture," *Inf. Sci. (Ny).*, 2019.
[171] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," *arXiv Prepr. arXiv1509.01626*, 2015.
[172] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[173] C. Wartena and R. Brussee, "Topic detection by clustering keywords," in *2008 19th international workshop on database and expert systems applications*, 2008, pp. 54–58.
[174] M. Ruiz-Casado, E. Alfonseca, and P. Castells, "Automatic assignment of Wikipedia encyclopedic entries to WordNet synsets," in *International Atlantic Web Intelligence Conference*, 2005, pp. 380–386.
[175] C. Rong, "Using Mahout for clustering Wikipedia's latest articles: A comparison between k-means and fuzzy c-means in the cloud," in *2011 IEEE Third International Conference on Cloud Computing Technology and Science*, 2011, pp. 565–569.
[176] B. Ghojogh *et al.*, "Feature selection and feature extraction in pattern analysis: A literature review," *arXiv Prepr. arXiv1905.02845*, 2019.
[177] E. Alpaydin, "Introduction to machine learning. 3rd." Cambridge, MA, USA: MIT Press, 2014.
[178] C. M. Bishop, "Pattern recognition," *Mach. Learn.*, vol. 128, no. 9, 2006.
[179] M. Zareapoor and K. R. Seeja, "Feature extraction or feature selection for text classification: A case study on phishing email detection," *Int. J. Inf. Eng. Electron. Bus.*, vol. 7, no. 2, p. 60, 2015.
[180] M. A. Hall and L. A. Smith, "Practical feature subset selection for machine learning," 1998.
[181] S. Siddiqui, M. A. Rehman, S. M. Doudpota, and A. Waqas, "Ontology driven feature engineering for opinion mining," *IEEE Access*, vol. 7, pp. 67392–67401, 2019.

[182] T. Mori, “Information gain ratio as term weight: the case of summarization of ir results,” in *COLING 2002: The 19th International Conference on Computational Linguistics*, 2002.
[183] R. Verma, N. Shashidhar, and N. Hossain, “Detecting phishing emails the natural language way,” in *European Symposium on Research in Computer Security*, 2012, pp. 824–841.
[184] E. S. Aung, C. T. Zan, and H. Yamana, “A Survey of URL-based phishing detection,” in *DEIM Forum*, 2019, pp. G2-3.
[185] G. Van Rossum and F. L. Drake Jr, *Python tutorial*, vol. 620. Centrum voor Wiskunde en Informatica Amsterdam, 1995.
[186] F. Eibe, M. A. Hall, and I. H. Witten, “The WEKA workbench. Online appendix for data mining: practical machine learning tools and techniques,” in *Morgan Kaufmann*, 2016.
[187] *TensorFlow Core | Machine Learning for Beginners and Experts*. 2019.
[188] F. Pedregosa *et al.*, “Scikit-learn: Machine learning in Python,” *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.
[189] E. Loper and S. Bird, “Nltk: The natural language toolkit,” *arXiv Prepr. cs/0205028*, 2002.
[190] S. Matlab, “Matlab,” *MathWorks, Natick, MA*, 2012.
[191] C. Sammut and G. I. Webb, *Encyclopedia of machine learning and data mining*. Springer, 2017.
[192] V. Ramanathan and H. Wechsler, “Phishing detection and impersonated entity discovery using Conditional Random Field and Latent Dirichlet Allocation,” *Comput. Secur.*, vol. 34, pp. 123–139, 2013.
[193] M. Bekkar, H. K. Djemaa, and T. A. Alitouche, “Evaluation measures for models assessment over imbalanced data sets,” *J Inf Eng Appl*, vol. 3, no. 10, 2013.
[194] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, “Learning to detect malicious urls,” *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–24, 2011.
[195] R. B. Basnet and T. Doleck, “Towards developing a tool to detect phishing URLs: A machine learning approach,” in *2015 IEEE International Conference on Computational Intelligence & Communication Technology*, 2015, pp. 220–223.
[196] P. Dewan and P. Kumaraguru, “Facebook Inspector (FbI): Towards automatic real-time detection of malicious content on Facebook,” *Soc. Netw. Anal. Min.*, vol. 7, no. 1, p. 15, 2017.
[197] H. Sha, Z. Zhou, Q. Liu, T. Liu, and C. Zheng, “Limited dictionary builder: An approach to select representative tokens for malicious URLs detection,” in *2015 IEEE International Conference on Communications (ICC)*, 2015, pp. 7077–7082.
[198] A. Das, S. Baki, A. El Aassal, R. Verma, and A. Dunbar, “SoK: a comprehensive reexamination of phishing research from the security perspective,” *IEEE Commun. Surv. Tutorials*, vol. 22, no. 1, pp. 671–708, 2019.
[199] R. Lab@UH, “First security and privacy analytics anti-phishing shared task (iwspa-ap 2018),” accessed: Jan 16, 2020.” [Online]. Available: https://dasavisha.github.io/IWSPA-sharedtask/.
[200] J. Nazario, “‘Nazario’s phishing corpora,’ accessed: Jan 16, 2020.” [Online]. Available: https://monkey.org/~jose/phishing/.
[201] C. Project, “‘Enron email dataset,’ accessed: Jan 16, 2020.” [Online]. Available: http://www.cs.cmu.edu/~enron/.
[202] “The Apache Spamassassin Public Corpus.” [Online]. Available: https://spamassassin.apache.org/old/publiccorpus.
[203] “Lingspam.” [Online]. Available:

http://www.aueb.gr/users/ion/data/lingspam_public.tar.gz.
[204] "PU." [Online]. Available: http://www.aueb.gr/users/ion/data/PU123ACorpora.tar.gz.
[205] "MalwareDomains." [Online]. Available: http://www.malwaredomains.com/.
[206] "MalwareDomainList." [Online]. Available: https://www.malwaredomainlist.com/.
[207] "JWSPAMSPY." [Online]. Available: http://www.joewein.de/sw/blacklist.htm.
[208] "MalwareURL." [Online]. Available: https://www.malwareurl.com/.
[209] "Open-Phish." [Online]. Available: https://openphish.com/.
[210] "SPAM Archive," *Public email corpus*, 2012. .
[211] R. Dhamija and J. D. Tygar, "The battle against phishing: Dynamic security skins," in *Proceedings of the 2005 symposium on Usable privacy and security*, 2005, pp. 77–88.
[212] M. Aburrous, M. A. Hossain, K. Dahal, and F. Thabtah, "Predicting phishing websites using classification mining techniques with experimental case studies," in *2010 Seventh International Conference on Information Technology: New Generations*, 2010, pp. 176–181.
[213] "PhishTank Phishing Archive," 2014. .
[214] "Apache Software Foundation (2014) Spamassassin public corpus, 2006." .
[215] B. B. Gupta, A. Tewari, A. K. Jain, and D. P. Agrawal, "Fighting against phishing attacks: state of the art and future challenges," *Neural Comput. Appl.*, vol. 28, no. 12, pp. 3629–3654, 2017.
[216] M. Khonji, Y. Iraqi, and A. Jones, "Lexical URL analysis for discriminating phishing and legitimate e-mail messages," in *2011 International Conference for Internet Technology and Secured Transactions*, 2011, pp. 422–427.
[217] Cohen, "Enron email dataset," 2014. .
[218] T. E. S. Datasets, "AEUB natural language processing group," *Athens, Greece.*, 2014. .
[219] B. Klimt and Y. Yang, "The enron corpus: A new dataset for email classification research," in *European Conference on Machine Learning*, 2004, pp. 217–226.
[220] K. Georgala, A. Kosmopoulos, and G. Paliouras, "Spam filtering: an active learning approach using incremental clustering," in *Proceedings of the 4th international conference on web intelligence, mining and semantics (WIMS14)*, 2014, pp. 1–12.
[221] G. V Cormack and T. R. Lynam, "TREC 2005 Spam Track Overview.," in *TREC*, 2005, pp. 274–500.
[222] "IronPort Anti-Spam," 2014. .
[223] T. Moore, R. Clayton, and H. Stern, "Temporal Correlations between Spam and Phishing Websites.," in *LEET*, 2009.
[224] "SpamCopWiki: SpamTrap (2014) 21 July 2006." .
[225] "The Phishload Phishing Test Database." .
[226] "'The dada engine,' accessed: Jan 16, 2020." [Online]. Available: http://dev.null.org/dadaengine/.
[227] C. Group, "Enron-Spam datasets." .
[228] J. Batra, R. Jain, V. A. Tikkiwal, and A. Chakraborty, "A comprehensive study of spam detection in e-mails using bio-inspired optimization techniques," *Int. J. Inf. Manag. Data Insights*, vol. 1, no. 1, p. 100006, 2021.
[229] M. K. Hanif, R. Talib, M. Awais, M. Y. Saeed, and U. Sarwar, "Comparison of bioinspired computation and optimization techniques.," *Curr. Sci.*, vol. 115, no. 3, 2018.
[230] A. A. Sekh, D. P. Dogra, S. Kar, P. P. Roy, and D. K. Prasad, "ELM-HTM guided bio-inspired unsupervised learning for anomalous trajectory classification," *Cogn. Syst. Res.*,

vol. 63, pp. 30–41, 2020.
[231] S. Gunawardena, D. Kulkarni, and B. Gnanasekaraiyer, "A steganography-based framework to prevent active attacks during user authentication," in *2013 8th International Conference on Computer Science & Education*, 2013, pp. 383–388.
[232] S. Gupta, A. Singhal, and A. Kapoor, "A literature survey on social engineering attacks: Phishing attack," in *2016 international conference on computing, communication and automation (ICCCA)*, 2016, pp. 537–540.
[233] G. Mujtaba, L. Shuib, R. G. Raj, N. Majeed, and M. A. Al-Garadi, "Email classification research trends: review and open issues," *IEEE Access*, vol. 5, pp. 9044–9064, 2017.
[234] E. S. Gualberto, R. T. De Sousa, P. D. B. Thiago, J. P. C. L. Da Costa, and C. G. Duque, "From Feature Engineering and Topics Models to Enhanced Prediction Rates in Phishing Detection," *IEEE Access*, vol. 8, pp. 76368–76385, 2020.
[235] G. Sonowal and K. S. Kuppusamy, "PhiDMA–A phishing detection model with multi-filter approach," *J. King Saud Univ. Inf. Sci.*, vol. 32, no. 1, pp. 99–112, 2020.
[236] A. Zamir *et al.*, "Phishing web site detection using diverse machine learning algorithms," *Electron. Libr.*, 2020.
[237] and C. Z. S. Sheng, B. Wardman, G. Warner, L. Cranor, J. Hong, "An empirical analysis of phishing blacklists," *Proc. 6th Conf. Email Anti-Spam (CEAS), Sacramento, CA, USA*, pp. 1–10, 2009.
[238] R. V. and N. Hossain, "Semantic feature selection for text with application to phishing email detection," *Proc. Int. Conf. Inf. Secur. Cryptol. Cham, Switz. Springer*, pp. 455–468, 2013.
[239] G. Park and J. M. Taylor, "Using syntactic features for phishing detection," *arXiv Prepr. arXiv1506.00037*, 2015.
[240] I. R. A. Hamid and J. Abawajy, "Hybrid feature selection for phishing email detection," in *International Conference on Algorithms and Architectures for Parallel Processing*, 2011, pp. 266–275.
[241] A. Aljofey, Q. Jiang, Q. Qu, M. Huang, and J.-P. Niyigena, "An Effective Phishing Detection Model Based on Character Level Convolutional Neural Network from URL," *Electronics*, vol. 9, no. 9, p. 1514, 2020.
[242] "No Title." [Online]. Available: www.phishtank.com.
[243] "No Title." [Online]. Available: https://joewein.net/spam/index.htm.
[244] A. Bergholz, J. De Beer, S. Glahn, M.-F. Moens, G. Paaß, and S. Strobel, "New filtering approaches for phishing email," *J. Comput. Secur.*, vol. 18, no. 1, pp. 7–35, 2010.
[245] J. Singh, "Detection of phishing e-mail," *in Proc. IJCST*, vol. 2, no. 1, pp. 547-_549, 2011.
[246] X. Gu and H. Wang, "Online anomaly prediction for robust cluster systems," in *2009 IEEE 25th International Conference on Data Engineering*, 2009, pp. 1000–1011.
[247] C. N. Gutierrez *et al.*, "Learning from the ones that got away: Detecting new forms of phishing attacks," *IEEE Trans. Dependable Secur. Comput.*, vol. 15, no. 6, pp. 988–1001, 2018.
[248] X. Glorot, A. Bordes, and Y. Bengio, "Domain adaptation for large-scale sentiment classification: A deep learning approach," in *ICML*, 2011.
[249] T. H. Nguyen and R. Grishman, "Relation extraction: Perspective from convolutional neural networks," in *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, 2015, pp. 39–48.
[250] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv Prepr. arXiv1409.0473*, 2014.

[251] T. Repke and R. Krestel, "Bringing back structure to free text email conversations with recurrent neural networks," in *European Conference on Information Retrieval*, 2018, pp. 114–126.
[252] F. Chollet, *Deep learning with Python*, vol. 361. Manning New York, 2018.
[253] G. Sonowal, "Phishing Email Detection Based on Binary Search Feature Selection," *SN Comput. Sci.*, vol. 1, no. 4, 2020.
[254] A. Ora, "Spam Detection in Short Message Service Using Natural Language Processing and Machine Learning Techniques." Dublin, National College of Ireland, 2020.
[255] "Python." [Online]. Available: https://www.python.org/.
[256] "Weka." [Online]. Available: https://www.cs.waikato.ac.nz/ml/weka/.
[257] P. Kumaraguru *et al.*, "Getting users to pay attention to anti-phishing education: evaluation of retention and transfer," in *Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit*, 2007, pp. 70–81.
[258] S. A. Salloum, M. Al-Emran, and K. Shaalan, "A Survey of Lexical Functional Grammar in the Arabic Context," *Int. J. Com. Net. Tech*, vol. 4, no. 3, 2016.
[259] A. Farghaly and K. Shaalan, "Arabic natural language processing: Challenges and solutions," *ACM Trans. Asian Lang. Inf. Process.*, vol. 8, no. 4, p. 14, 2009.
[260] A. A. Rafea and K. F. Shaalan, "Lexical analysis of inflected Arabic words using exhaustive search of an augmented transition network," *Softw. Pract. Exp.*, vol. 23, no. 6, pp. 567–588, 1993.
[261] K. Shaalan, M. Attia, P. Pecina, Y. Samih, and J. van Genabith, "Arabic word generation and modelling for spell checking.," in *LREC*, 2012, pp. 719–725.
[262] H. H. Elachachi, "Exploring cultural barriers in EFL Arab learners' writing," *Procedia-Social Behav. Sci.*, vol. 199, pp. 129–136, 2015.
[263] H. Al-Ajmi, "A new English–Arabic parallel text corpus for lexicographic applications," *Lexikos*, vol. 14, 2004.
[264] H. Salhi, "Investigating the complementary polysemy and the Arabic translations of the noun destruction in EAPCOUNT," *Meta J. des traducteurs/Meta Transl. J.*, vol. 58, no. 1, pp. 227–246, 2013.
[265] A. Eisele and Y. Chen, "MultiUN: A Multilingual Corpus from United Nation Documents.," in *LREC*, 2010.
[266] J. Tiedemann, "Parallel data, tools and interfaces in OPUS.," in *Lrec*, 2012, vol. 2012, pp. 2214–2218.
[267] "Linguistic Data Consortium (LDC)," *LDC catalog.*, 2013. .
[268] S. Izwaini, "A corpus-based study of metaphor in information technology," in *Corpus Linguistics*, 2003.
[269] A. Abdelali, F. Guzman, H. Sajjad, and S. Vogel, "The AMARA Corpus: Building Parallel Language Resources for the Educational Domain.," in *LREC*, 2014, vol. 14, pp. 1044–1054.
[270] F. Guzmán, H. Sajjad, S. Vogel, and A. Abdelali, "The AMARA corpus: Building resources for translating the web's educational content," in *Proceedings of the 10th International Workshop on Spoken Language Technology (IWSLT-13)*, 2013.
[271] "AMARA." [Online]. Available: www.amara.org.
[272] S. Alkahtani and W. J. Teahan, "A new parallel corpus of Arabic/English," in *Proceedings of the Eighth Saudi Students Conference in the UK*, 2016, pp. 279–284.
[273] S. M. O. Hassan and E. S. Atwell, "Design and implementing of multilingual Hadith corpus," *Int. J. Recent Res. Soc. Sci. Humanit.*, vol. 3, no. 2, pp. 100–104, 2016.

[274] R. M. Verma, V. Zeng, and H. Faridi, "Data Quality for Security Challenges: Case Studies of Phishing, Malware and Intrusion Detection Datasets," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 2605–2607.
[275] A. E. Aassal, L. Moraes, S. Baki, A. Das, and R. Verma, "Anti-phishing pilot at ACM IWSPA 2018: Evaluating performance with new metrics for unbalanced datasets," in *Proc. IWSPA-AP Anti Phishing Shared Task Pilot 4th ACM IWSPA*, 2018, pp. 2–10.
[276] V. Ra, B. G. HBa, A. K. Ma, S. KPa, P. Poornachandran, and A. Verma, "DeepAnti-PhishNet: Applying deep neural networks for phishing email detection," in *Proc. 1st AntiPhishing Shared Pilot 4th ACM Int. Workshop Secur. Privacy Anal.(IWSPA)*, 2018, pp. 1–11.
[277] B. G. HBa, V. Ra, A. K. Ma, and S. KPa, "Distributed Representation using Target Classes: Bag of Tricks for Security and Privacy Analytics."
[278] N. A. Unnithan, N. B. Harikrishnan, S. Akarsh, R. Vinayakumar, and K. P. Soman, "Machine learning based phishing e-mail detection," *Secur. Amrita*, pp. 65–69, 2018.
[279] C. Coyotes, V. S. Mohan, J. Naveen, R. Vinayakumar, K. P. Soman, and A. D. R. Verma, "ARES: Automatic rogue email spotter," in *Proc. 1st AntiPhishing Shared Pilot 4th ACM Int. Workshop Secur. Privacy Anal.(IWSPA)*, 2018.
[280] M. Nguyen, T. Nguyen, and T. H. Nguyen, "A deep learning model with hierarchical lstms and supervised attention for anti-phishing," *arXiv Prepr. arXiv1805.01554*, 2018.
[281] A. Bahameed, "Hindrances in Arabic-English intercultural translation," *Transl. J.*, vol. 12, no. 1, pp. 1–16, 2008.
[282] Z. A. Alfahad and J. A. A. Abd Al-Hasan, "Machine Translation of Arabic Verb Sentences into English," *Basrah J. Sci.*, vol. 26, no. 1, pp. 25–32, 2008.
[283] B. S. Dweik and M. Suleiman, "Problems encountered in translating cultural expressions from Arabic into English," *Int. J. English Linguist.*, vol. 3, no. 5, p. 47, 2013.
[284] H. Ghazala, *TRANSLATION AS PROBLEMS AND SOLUTIONS*, Special Ed. Beirut, Lebanon: Dar Elilm lilmalayin, 2008.
[285] N. Y. Habash, *Introduction to Arabic natural language processing*. Springer Nature, 2022.
[286] M. Heikal, M. Torki, and N. El-Makky, "Sentiment analysis of Arabic tweets using deep learning," *Procedia Comput. Sci.*, vol. 142, pp. 114–122, 2018.
[287] A. M. Almuhaideb *et al.*, "Homoglyph Attack Detection Model Using Machine Learning and Hash Function," *J. Sens. Actuator Networks*, vol. 11, no. 3, p. 54, 2022.
[288] T. Zerrouki, "Pyarabic, an Arabic language library for Python." Pyarabic, 2010.
[289] D. Namly, K. Bouzoubaa, R. Tajmout, and A. Laadimi, "On Arabic Stop-Words: A Comprehensive List and a Dedicated Morphological Analyzer," in *International Conference on Arabic Language Processing*, 2019, pp. 149–163.
[290] A. Alajmi, E. M. Saad, and R. R. Darwish, "Toward an ARABIC stop-words list generation," *Int. J. Comput. Appl.*, vol. 46, no. 8, pp. 8–13, 2012.
[291] I. A. El-Khair, "Effects of stop words elimination for Arabic information retrieval: a comparative study," *Int. J. Comput. Inf. Sci.*, vol. 4, no. 3, pp. 119–133, 2006.
[292] J. Atwan, M. Mohd, and G. Kanaan, "Enhanced arabic information retrieval: Light stemming and stop words," in *International Multi-Conference on Artificial Intelligence Technology*, 2013, pp. 219–228.
[293] R. Z. Al-Abdallah and A. T. Al-Taani, "Arabic single-document text summarization using particle swarm optimization algorithm," *Procedia Comput. Sci.*, vol. 117, pp. 30–37, 2017.
[294] R. Z. Al-Abdallah and A. T. Al-Taani, "Arabic text summarization using firefly algorithm,"

in *2019 amity international conference on artificial intelligence (AICAI)*, 2019, pp. 61–65.
[295] A. W. Pradana and M. Hayaty, "The effect of stemming and removal of stopwords on the accuracy of sentiment analysis on indonesian-language texts," *Kinet. Game Technol. Inf. Syst. Comput. Network, Comput. Electron. Control*, pp. 375–380, 2019.
[296] Z. J. and G. Xiaolini, "Comparison Research on Text Pre-processing Methods on Twitter Sentiment Analysis," *Digit. Object Identifier*, 2017.
[297] S. G. Bird and E. Loper, "NLTK: the natural language toolkit," 2004.
[298] "Arabic Stopwords," 2019. [Online]. Available: https://github.com/mohataher/arabic-stop-words/blob/master/list.txt#L751.
[299] M. Al-Ayyoub, A. A. Khamaiseh, Y. Jararweh, and M. N. Al-Kabi, "A comprehensive survey of arabic sentiment analysis," *Inf. Process. Manag.*, vol. 56, no. 2, pp. 320–342, 2019.
[300] A. Shoukry and A. Rafea, "Sentence-level Arabic sentiment analysis," in *2012 International Conference on Collaboration Technologies and Systems (CTS)*, 2012, pp. 546–550.
[301] A. Ghallab, A. Mohsen, and Y. Ali, "Arabic sentiment analysis: A systematic literature review," *Appl. Comput. Intell. Soft Comput.*, vol. 2020, pp. 1–21, 2020.
[302] R. M. Duwairi and I. Qarqaz, "Arabic sentiment analysis using supervised classification," in *2014 International Conference on Future Internet of Things and Cloud*, 2014, pp. 579–583.
[303] M. Alruily, "Classification of arabic tweets: A review," *Electronics*, vol. 10, no. 10, p. 1143, 2021.
[304] A. Omar, T. M. Mahmoud, T. Abd-El-Hafeez, and A. Mahfouz, "Multi-label arabic text classification in online social networks," *Inf. Syst.*, vol. 100, p. 101785, 2021.
[305] H. Sawaf, J. Zaplo, and H. Ney, "Statistical classification methods for Arabic news articles," *Nat. Lang. Process. ACL2001, Toulouse, Fr.*, 2001.
[306] A. Y. Muaad *et al.*, "Arabic document classification: performance investigation of preprocessing and representation techniques," *Math. Probl. Eng.*, vol. 2022, pp. 1–16, 2022.
[307] D. Abuaiadah, J. El Sana, and W. Abusalah, "On the impact of dataset characteristics on arabic document classification," *Int. J. Comput. Appl.*, vol. 101, no. 7, 2014.
[308] M. El Kourdi, A. Bensaid, and T. Rachidi, "Automatic Arabic document categorization based on the Naïve Bayes algorithm," p. 51, 2004.
[309] F. Strieth-Kalthoff, F. Sandfort, M. H. S. Segler, and F. Glorius, "Machine learning the ropes: principles, applications and directions in synthetic chemistry," *Chem. Soc. Rev.*, vol. 49, no. 17, pp. 6154–6168, 2020.
[310] I. Guyon, S. Gunn, M. Nikravesh, and L. A. Zadeh, *Feature extraction: foundations and applications*, vol. 207. Springer, 2008.
[311] M. Jogin, M. S. Madhulika, G. D. Divya, R. K. Meghana, and S. Apoorva, "Feature extraction using convolution neural networks (CNN) and deep learning," in *2018 3rd IEEE international conference on recent trends in electronics, information & communication technology (RTEICT)*, 2018, pp. 2319–2323.
[312] A. Mukherjee, N. Agarwal, and S. Gupta, "A SURVEY ON AUTOMATIC PHISHING EMAIL DETECTION USING NATURAL LANGUAGE PROCESSING TECHNIQUES," 2019.
[313] S. Liu *et al.*, "Visual exploration of semantic relationships in neural word embeddings," *IEEE Trans. Vis. Comput. Graph.*, vol. 24, no. 1, pp. 553–562, 2017.
[314] S. Qaiser and R. Ali, "Text mining: use of TF-IDF to examine the relevance of words to

documents," *Int. J. Comput. Appl.*, vol. 181, no. 1, pp. 25–29, 2018.
[315] T. Joachims, "A probabilistic analysis of the Rocchio algorithm with TFIDF for text," in *Int l Conf on Machine learning (ICML). Ten-7301*, 1997, no. 9.
[316] M. Anandarajan, C. Hill, T. Nolan, M. Anandarajan, C. Hill, and T. Nolan, "Term-document representation," *Pract. Text Anal. Maximizing Value Text Data*, pp. 61–73, 2019.
[317] A. Bin Raies, H. Mansour, R. Incitti, and V. B. Bajic, "Combining position weight matrices and document-term matrix for efficient extraction of associations of methylated genes and diseases from free text," *PLoS One*, vol. 8, no. 10, p. e77848, 2013.
[318] A. Kulkarni and A. Shivananda, "Converting text to features," in *Natural language processing recipes*, Springer, 2019, pp. 67–96.
[319] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv Prepr. arXiv1301.3781*, 2013.
[320] N. Bensalah, H. Ayad, A. Adib, and A. I. El Farouk, "Arabic machine translation based on the combination of word embedding techniques," in *Intelligent Systems in Big Data, Semantic Web and Machine Learning*, Springer, 2021, pp. 247–260.
[321] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov, "Fasttext. zip: Compressing text classification models," *arXiv Prepr. arXiv1612.03651*, 2016.
[322] N. Alghamdi and F. Assiri, "A comparison of fasttext implementations using Arabic text classification," in *Proceedings of SAI Intelligent Systems Conference*, 2019, pp. 306–311.
[323] I. Al-Huri, "Arabic language: Historic and sociolinguistic characteristics," *English Lit. Lang. Rev.*, vol. 1, no. 4, pp. 28–36, 2015.
[324] A. Soudi, G. Neumann, and A. van den Bosch, *Arabic computational morphology: knowledge-based and empirical methods*. Springer, 2007.
[325] A. Mahfoudhi, J. Everatt, and G. Elbeheri, "Introduction to the special issue on literacy in Arabic," *Read. Writ.*, vol. 24, no. 9, pp. 1011–1018, 2011.
[326] A. Elnagar, S. M. Yagi, A. B. Nassif, I. Shahin, and S. A. Salloum, "Systematic Literature Review of Dialectal Arabic: Identification and Detection," *IEEE Access*, vol. 9, pp. 31010–31042, 2021.
[327] B. Li, A. Drozd, T. Liu, and X. Du, "Subword-level composition functions for learning word embeddings," in *Proceedings of the second workshop on subword/character level models*, 2018, pp. 38–48.
[328] F. A. O. Santos, T. D. Bispo, H. T. Macedo, and C. Zanchettin, "Morphological Skip-Gram: Replacing FastText characters n-gram with morphological knowledge," *Intel. Artif.*, vol. 24, no. 67, pp. 1–17, 2021.
[329] A. Conneau, G. Kruszewski, G. Lample, L. Barrault, and M. Baroni, "What you can cram into a single vector: Probing sentence embeddings for linguistic properties," *arXiv Prepr. arXiv1805.01070*, 2018.
[330] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," *arXiv Prepr. arXiv1607.01759*, 2016.
[331] H. Bouamor, N. Habash, and K. Oflazer, "A Multidialectal Parallel Corpus of Arabic.," in *LREC*, 2014, pp. 1240–1245.
[332] M. Umer *et al.*, "Impact of convolutional neural network and FastText embedding on text classification," *Multimed. Tools Appl.*, vol. 82, no. 4, pp. 5569–5585, 2023.
[333] B. Kuyumcu, C. Aksakalli, and S. Delil, "An automated new approach in fast text classification (fastText) A case study for Turkish text classification without pre-processing," in *Proceedings of the 2019 3rd International Conference on Natural Language*

Processing and Information Retrieval, 2019, pp. 1–4.
[334] I. Santos, N. Nedjah, and L. de Macedo Mourelle, "Sentiment analysis using convolutional neural network with fastText embeddings," in *2017 IEEE Latin American conference on computational intelligence (LA-CCI)*, 2017, pp. 1–5.
[335] A. A. Altowayan and A. Elnagar, "Improving Arabic sentiment analysis with sentiment-specific embeddings," in *2017 IEEE International Conference on Big Data (Big Data)*, 2017, pp. 4314–4320.
[336] M. Berrimi, M. Oussalah, A. Moussaoui, and M. Saidi, "Attention mechanism architecture for arabic sentiment analysis," *ACM Trans. Asian Low-Resource Lang. Inf. Process.*, vol. 22, no. 4, pp. 1–26, 2023.
[337] T. Mikolov, E. Grave, P. Bojanowski, C. Puhrsch, and A. Joulin, "Advances in pre-training distributed word representations," *arXiv Prepr. arXiv1712.09405*, 2017.
[338] A. H. Ombabi, W. Ouarda, and A. M. Alimi, "Deep learning CNN–LSTM framework for Arabic sentiment analysis using textual information shared in social networks," *Soc. Netw. Anal. Min.*, vol. 10, pp. 1–13, 2020.
[339] A. Alwehaibi, M. Bikdash, M. Albogmi, and K. Roy, "A study of the performance of embedding methods for Arabic short-text sentiment analysis using deep learning approaches," *J. King Saud Univ. Inf. Sci.*, vol. 34, no. 8, pp. 6140–6149, 2022.
[340] M. Al-Smadi, B. Talafha, M. Al-Ayyoub, and Y. Jararweh, "Using long short-term memory deep neural networks for aspect-based sentiment analysis of Arabic reviews," *Int. J. Mach. Learn. Cybern.*, vol. 10, no. 8, pp. 2163–2175, 2019.
[341] L. Moudjari, F. Benamara, and K. Akli-Astouati, "Multi-level embeddings for processing Arabic social media contents," *Comput. Speech Lang.*, vol. 70, p. 101240, 2021.
[342] S. Elkateb *et al.*, "Arabic WordNet and the challenges of Arabic," in *Proceedings of the International Conference on the Challenge of Arabic for NLP/MT*, 2006.
[343] B. Edizel, A. Piktus, P. Bojanowski, R. Ferreira, E. Grave, and F. Silvestri, "Misspelling oblivious word embeddings," *arXiv Prepr. arXiv1905.09755*, 2019.
[344] A. Al-Alwani and M. Beseiso, "Arabic spam filtering using bayesian model," *Int. J. Comput. Appl.*, vol. 79, no. 7, 2013.
[345] V. Efstathiou and D. Spinellis, "Semantic source code models using identifier embeddings," in *2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR)*, 2019, pp. 29–33.
[346] O. Kwon, D. Kim, S.-R. Lee, J. Choi, and S. Lee, "Handling out-of-vocabulary problem in hangeul word embeddings," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 2021, pp. 3213–3221.
[347] N. Stevanović, "Character and word embeddings for phishing email detection," *Comput. Informatics*, vol. 41, no. 5, pp. 1337–1357, 2022.
[348] X. Chen, L. Xu, Z. Liu, M. Sun, and H. Luan, "Joint learning of character and word embeddings," in *Twenty-fourth international joint conference on artificial intelligence*, 2015.
[349] D. Watson, N. Zalmout, and N. Habash, "Utilizing character and word embeddings for text normalization with sequence-to-sequence models," *arXiv Prepr. arXiv1809.01534*, 2018.
[350] P. Ayuningtyas, "Whatsapp: Learning on the go," *Metathesis J. English Lang. Lit. Teach.*, vol. 2, no. 2, p. 159, 2018.
[351] M. Khalifa and K. Shaalan, "Character convolutions for Arabic named entity recognition

with long short-term memory networks," *Comput. Speech Lang.*, vol. 58, pp. 335–346, 2019.
[352] P. Ha, S. Zhang, N. Djuric, and S. Vucetic, "Improving word embeddings through iterative refinement of word-and character-level models," in *Proceedings of the 28th International Conference on Computational Linguistics*, 2020, pp. 1204–1213.
[353] C. Sun, X. Qiu, and X. Huang, "VCWE: visual character-enhanced word embeddings," *arXiv Prepr. arXiv1902.08795*, 2019.
[354] P. Hannay and C. Bolan, "Assessment of internationalised domain name homograph attack mitigation," 2009.
[355] D. Xu, P. L. Pearce, and T. Chen, "Deconstructing tourist scams: A social-practice-theory perspective," *Tour. Manag.*, vol. 82, p. 104186, 2021.
[356] M. M. Al-Daeef, N. Basir, and M. M. Saudi, "A method to measure the efficiency of phishing emails detection features," in *2014 International Conference on Information Science & Applications (ICISA)*, 2014, pp. 1–5.
[357] M. Chandrasekaran, K. Narayanan, and S. Upadhyaya, "Phishing email detection based on structural properties," in *NYS cyber security conference*, 2006, vol. 3.
[358] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning*, vol. 112. Springer, 2013.
[359] D. M. Hawkins, "The problem of overfitting," *J. Chem. Inf. Comput. Sci.*, vol. 44, no. 1, pp. 1–12, 2004.
[360] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Ijcai*, 1995, vol. 14, no. 2, pp. 1137–1145.
[361] A. Kesarwani, S. S. Chauhan, and A. R. Nair, "Fake news detection on social media using k-nearest neighbor classifier," in *2020 international conference on advances in computing and communication engineering (ICACCE)*, 2020, pp. 1–4.
[362] J. P. Singh, U. P. Singh, and S. Jain, "Model-based person identification in multi-gait scenario using hybrid classifier," *Multimed. Syst.*, pp. 1–14, 2023.
[363] P. Yang, G. Zhao, and P. Zeng, "Phishing website detection based on multidimensional features driven by deep learning," *IEEE Access*, vol. 7, pp. 15196–15209, 2019.
[364] G. Jain, M. Sharma, and B. Agarwal, "Spam detection in social media using convolutional and long short term memory neural network," *Ann. Math. Artif. Intell.*, vol. 85, no. 1, pp. 21–44, 2019.
[365] S. Pandey and R. Yadav, "E-Mail Spam Detection using Machine Learning and Deep Learning." IJRASET, 2020.
[366] B. Wei *et al.*, "A Deep-Learning-Driven Light-Weight Phishing Detection Sensor," *Sensors*, vol. 19, no. 19, p. 4258, 2019.
[367] P. Ce and B. Tie, "An analysis method for interpretability of CNN text classification model," *Futur. Internet*, vol. 12, no. 12, p. 228, 2020.
[368] W. Lim, D. Jang, and T. Lee, "Speech emotion recognition using convolutional and recurrent neural networks," in *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, 2016, pp. 1–4.
[369] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
[370] K. Iluore and J. Lu, "Long short-term memory and gated recurrent neural networks to predict the ionospheric vertical total electron content," *Adv. Sp. Res.*, vol. 70, no. 3, pp. 652–665, 2022.

[371] S. Cornegruta, R. Bakewell, S. Withey, and G. Montana, "Modelling radiological language with bidirectional long short-term memory networks," *arXiv Prepr. arXiv1609.08409*, 2016.
[372] P. Li *et al.*, "Bidirectional gated recurrent unit neural network for Chinese address element segmentation," *ISPRS Int. J. Geo-Information*, vol. 9, no. 11, p. 635, 2020.
[373] S. A. A. O. Maeli and A. U. Surwade, "Phishing E-Mail Detection and Blocking it based on the Header Elements," *AITC-2023 and CSSP-2023*, p. 61, 2023.
[374] S. R. Abdul Samad *et al.*, "Analysis of the Performance Impact of Fine-Tuned Machine Learning Model for Phishing URL Detection," *Electronics*, vol. 12, no. 7, p. 1642, 2023.
[375] X. Fang, N. Koceja, J. Zhan, G. Dozier, and D. Dipankar, "An artificial immune system for phishing detection," in *2012 IEEE Congress on Evolutionary Computation*, 2012, pp. 1–7.
[376] F. Toolan and J. Carthy, "Feature selection for spam and phishing detection," in *2010 eCrime Researchers Summit*, 2010, pp. 1–12.
[377] S. Aggarwal, V. Kumar, and S. D. Sudarsan, "Identification and detection of phishing emails using natural language processing techniques," in *Proceedings of the 7th International Conference on Security of Information and Networks*, 2014, pp. 217–222.
[378] K. Ding, N. Pantic, Y. Lu, S. Manna, and M. I. Husain, "Towards building a word similarity dictionary for personality bias classification of phishing email contents," in *Proceedings of the 2015 IEEE 9th International Conference on Semantic Computing (IEEE ICSC 2015)*, 2015, pp. 252–259.
[379] S. Kaddoura, O. Alfandi, and N. Dahmani, "A Spam Email Detection Mechanism for English Language Text Emails Using Deep Learning Approach," in *2020 IEEE 29th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, 2020, pp. 193–198.
[380] V. Ramanathan and H. Wechsler, "phishGILLNET—phishing detection methodology using probabilistic latent semantic analysis, AdaBoost, and co-training," *EURASIP J. Inf. Secur.*, vol. 2012, no. 1, p. 1, 2012.
[381] S. Smadi, N. Aslam, and L. Zhang, "Detection of online phishing email using dynamic evolving neural network based on reinforcement learning," *Decis. Support Syst.*, vol. 107, pp. 88–102, 2018.
[382] A. Alhogail and A. Alsabih, "Applying machine learning and natural language processing to detect phishing email," *Comput. Secur.*, vol. 110, p. 102414, 2021.
[383] "Phishery," 2007. .
[384] CSIRO, "https://www.csiro.au/." [Online]. Available: https://www.csiro.au/.
[385] C. University, "'Phis bowl,' accessed: Jan 3, 2021." [Online]. Available: https://it.cornell.edu/phish-bowl.
[386] J. Ulrich, G. Murray, and G. Carenini, "A publicly available annotated corpus for supervised email summarization," in *Proc. of aaai email-2008 workshop, chicago, usa*, 2008.
[387] "Alexa." [Online]. Available: https://www.alexa.com/siteinfo.
[388] "DMOZ." [Online]. Available: http://www.dmoz.org/.
[389] "Majestic." [Online]. Available: https://github.com/rlilojr/Detecting-Malicious-URL-Machine-Learning.
[390] "Honey Trap database." [Online]. Available: millersmiles.co.uk.
[391] A. Oest, Y. Safei, A. Doupé, G.-J. Ahn, B. Wardman, and G. Warner, "Inside a phisher's mind: Understanding the anti-phishing ecosystem through phishing kit analysis," in *2018 APWG Symposium on Electronic Crime Research (eCrime)*, 2018, pp. 1–12.

[392] A. Harilal, F. Toffalini, J. Castellanos, J. Guarnizo, I. Homoliak, and M. Ochoa, “Twos: A dataset of malicious insider threat behavior based on a gamified competition,” in *Proceedings of the 2017 International Workshop on Managing Insider Security Threats*, 2017, pp. 45–56.
[393] D. D. and C. Graff, “UCI machine learning repository,” 2017. [Online]. Available: http://archive.ics.uci.edu/ml.
[394] karthick veerakumar, “Spam filter,” 2017. [Online]. Available: https://www.kaggle.com/karthickveerakumar/spam-filter.
[395] “TREC corpus.” [Online]. Available: http://trec.nist.gov/data/spam.%0Ahtml.

Appendix A

TABLE A1: ANALYSIS OF PHISHING EMAIL DETECTION RESEARCH PAPERS

D	Ref.	Year	Dataset	Problem domain	Feature representation	Proposed approach/ Optimizer(s)	Reported performance	Tools	NLP technique	Sample	Feature approach	No. of features	Limitation(s)
A1	[238]	2013	Enron, Nazario	Feature Extraction/ Selection + Phishing Email Detection	Email header & body, link, and URL.	ADND	PM + POS + Word Senses + WordNet (ACC 0.950 FP 2.24)	N/A	Lexical analysis, POS, NER, WordNet, Basic NLP tasks.	14550 emails	General semantic feature selection	N/A	Performance not compared with other method.
A2	[183]	2012	Nazario	Feature Extraction/ Selection + Phishing Email Detection	Email header & body, link, and URL.	PhishNet-NLP	ACC 0.970	Perl, WordNet, SenseLearner	Lexical analysis, POS, NER , Basic NLP tasks	3000 emails	N/A	N/A	Performance not compared with other method.
A3	[144]	2015	Enron, SpamAssassin	Feature Extraction/ Selection + Phishing Email Detection	Email header & body, link, and URL.	SVM, *SMO*	ACC 0.980	N/A	POS, NER, WordNet	126,075 emails	N/A	N/A	Only accuracy of the technique was used in evaluating it performance.
A4	[145]	2016	Selected emails	Feature Extraction/ Selection + Phishing Email Detection	Email header & body text.	SVM	ACC 0.980	Stanford CoreNLP	Basic NLP tasks, POS	50 emails	General Lexical , Syntactic and Structural feature selection	N/A	Only accuracy of the technique was used in evaluating it performance.
A5	[122]	2014	SpamAssassin, Nazario	Feature Extraction/ Selection + Classification of	Email header & body, link, and URL.	RF	ACC 997 FN 2.50, FP 0.06	C#	N/A	2000 emails	IG	15	Many of the modern phish classification methods were not examined.

				Phishing Email									
A6	[179]	2015	SpamAssassin, Nazario	Feature Extraction/ Selection + Classification of Phishing Email	Email header & body, link, and URL.	Bagging with J48	LSA (ROC 0.990 ACC 0.960)	Weka	Basic NLP tasks, TF-IDF	2700 emails	CS, IG, PCA, LSA	2173	Many of the modern phish classification methods were not examined.
A7	[123]	2013	SpamAssassin, Nazario	Feature Extraction/ Selection + Classification of Phishing Email	Email header & body, link, and URL.	ProEP, NB, AB, RF, DT, OneR-	AB (ACC 0.944)	Weka	N/A	6837 emails	IG	47	Only accuracy of the technique was used in evaluating it performance.
A8	[34]	2013	SpamAssassin, Nazario	Improving/ optimize algorithms + Phishing Email Detection	Email header & body, link, and URL.	PDENFF (DENFIS & DYNFIS), NB, Random k-means, MLP, SVM, DENFIS	PDENFF (ACC 0.990 TP 0.970 TN 0.980 PRE 0.980 REC 0.970 F1- 0.970)	Python, Java	N/A	6000 emails	Entropy and IG, PCA, LSA	21	Time taken in the long vector is very high.
A9	[124]	2011	SpamAssassin, Nazario	Feature Extraction/ Selection + Phishing Email Detection	Email header & body, link, and URL.	BN AB, DT, RF	ACC 0.930	Weka	Basic NLP tasks, TF-IDF	6923 emails	IG, GR, SU	7	Only accuracy of the technique was used in evaluating it performance.
A10	[376]	2010	SpamAssassin, Nazario	Feature Extraction/ Selection + Phishing	Email header & body, link, and URL.	C5.0 algorithm/ DT	ACC 0.971	N/A	N/A	10.000 emails	Entropy, IG	40	Only accuracy of the technique was used in

				Email Detection									evaluating it performance.
A11	[35]	2016	SpamAssassin, Nazario	Feature Extraction/ Selection + Phishing Email Detection	Email header & body, link, and URL.	J48, NB, SVM, MLP, RF	RF (ACC 0.991 TP 0.991 FP 0.011 PRE 0.991 REC 0.991 F1- 0.977 ROC 0.987)	Java	Basic NLP tasks , WordNet ontology, Phishing terms weighting	10538 emails	IG	16	Many of the modern phish classification methods were not examined.
A12	[146]	2008	SpamAssassin, Nazario	Feature Extraction/ Selection + Phishing Email Detection + Improving/ Optimize Algorithms	Email header & body, link, and URL.	SVM	ACC 0.998 FN 1.07 FP 0.000 PRE 1.000 REC 0.989 F1- 0.995 Error Red 0.771)	libSVM-library	N/A	3702 emails	LTM, DMC	27	Many of the modern phish classification methods were not examined.
A13	[147]	2007	SpamAssassin, Nazario	Phishing Email Detection	Email header & body, link, and URL.	SVM	ACC 0.995	The Spoofguard and Netcraft	Basic NLP tasks, TF-IDF	7810 emails	N/A	10	Only accuracy of the technique was used in evaluating it performance.
A14	[137]	2009	SpamAssassin, Nazario	Feature Extraction/ Selection + Phishing Email Detection	Email header & body, link, and URL.	C5.0, KNN, NB, SVM, LR	C5.0 (ACC 0.986)	N/A	N/A	4116 emails	IG	10	Only accuracy of the technique was used in evaluating it performance.
A15	[138]	2013	SpamAssassin, Nazario	Phishing Email Detection	Email header & body text.	SVM,AB, NB	ACC 0.970 AUC 0.965 FP 0.020	N/A	Basic NLP tasks	N/A	N/A	21	Many of the modern phish classification methods were not examined.

A16	[192]	2013	SpamAssassin, Nazario, SPAM Archive, PhishTank	Feature Extraction/ Selection + Phishing Email Detection	Email header & body, link, and URL.	AB	TP 1.00 FP 0.000 PRE 1.00 REC 1.00 F1- 1.00 ROC 1.00	Java, Weka	Basic NLP tasks, NER	6750 phishing emails, 254,000 phishing URLs and 58,000 phishing websites.	CRF, LDA	N/A	Performance not compared with other technique.
A17	[15]	2017	Selected emails	Feature Extraction/ Selection + Phishing Email Detection	Email header & body, link, and URL.	BN, J48, KNN, RF, *SMO*	F1- 0.980	Weka, Python	Basic NLP tasks, TF-IDF	1492 emails	N/A	24	Only F1-Score of the technique was used in evaluating it performance.
A18	[125]	2009	Phishery, TREC	Feature Extraction/ Selection + Classification of Phishing Email	Email header & body, link, and URL.	J48 DT, RF, BM, SVM, KNN	SVM (ACC 0.997 FP 0.20)	Weka	N/A	11 000 phishing messages	IG	30	Only accuracy of the technique was used in evaluating it performance.
A19	[113]	2010	The industry partners of the Centre for Informatics and Applied Optimization	Feature Extraction/ Selection + Classification of Phishing Email	Email header & body, link, and URL.	*k*-means	ACC 93.55%	Python	Basic NLP tasks, TF-IDF.	3276 emails	N/A	15	Only accuracy of the technique was used in evaluating it performance.
A20	[114]	2017	Selected emails	Feature Extraction/ Selection + Classification of Phishing Email +	Email header & body, link, and URL.	*k*-means with *MS-MGKM, INCA, and DCClust*	ACC 0.827	N/A	Basic NLP tasks, TF-IDF	1277 email documents	BoW	70	Very low performance.

				Improving/ Optimize Algorithms									
A21	[45]	2021	IWSPA-AP, Nazario, Enron, CSIRO, Phisbowl	Feature Extraction/ Selection + Classification of Phishing Email	Email header & body, link, and URL.	THEMIS (RNN, BERT	ACC 0.993	Python, Tensor Flow, Keras	Basic NLP tasks, TF-IDF	23916 Email	Char-level email (header & body), word-level email (header & body)	N/A	Only accuracy of the technique was used in evaluating it performance.
A22	[26]	2020	SMS Spam Collection Data Set	Feature Extraction/ Selection + Classification of Phishing Email	Email header & body, link, and URL.	SVM LIR , NB, KNN, RF, DT, LR	SVM Linear (ACC 0.987 PRE 0.990 REC 1.00 F1- 1.00 TP 1.00 FP 0.00)	Python, Scikit-learn library	Basic NLP tasks , tagging, language detection and identification of semantic relationships	5,574 tagged (ham/spam)	N/A	N/A	Decreased amount of contextual information.
A23	[46]	2019	Enron, SpamAssassin, Nazario	Phishing Email Detection + Improving/ Optimize Algorithms	Email body text.	RNN with *Adam optimizer*	ACC 0.967 FN 4.02 FP 2.50 PRE 0.974 REC 0.9598 F1- 0.9671	N/A	Basic NLP tasks	31485 (ham/ phishing) email.	N/A	N/A	Many of the modern phish classification methods were not examined.
A24	[126]	2020	Selected emails	Feature Extraction/ Selection + Phishing Email Detection	Email header & body, link, and URL.	DT, NB, AB, LR, KNN, SVM, RF	RF- 2-D linear PCA (ACC 0.9160 F1- 0.900)	Python, Scikitlearn library-NLTK library	Basic NLP tasks, word embedding	24 (legitimate /phishing) email	Doc2Vec, PCA	N/A	The dataset size is too small.
A25	[18]	2020	Enron, APWG, BC3	Feature Extraction/ Selection + Phishing	Email body text.	BP,CNN, LSTM, SVM, NB,LR	BP (ACC 0.9568	Python, Keras - Gensim	Basic NLP tasks	130233 (legitimate	BoW, Word2vec	N/A	Only accuracy of the technique was used in

				Email Detection			)			/phishing) email			evaluating it performance.
A26	[19]	2018	Alexa, OpenDNS	Feature Extraction/ Selection + Classification of Phishing Email	Email header & body, link, and URL.	RNN, LSTM, CNN, CNN-RNN, GRU , LR, SPOOFNet, SVM	SPOOFNet (ACC 0.953, PRE 0.929, REC 0.997, F1-0.962	Python, scikit-learn library, Keras	Basic NLP tasks, TF-IDF	2723242 (legitimate /phishing) email	BoW, Word2vec, FT	N/A	They employed fundamental feature extraction techniques.
A27	[16]	2020	Selected emails	Feature Extraction/ Selection + Phishing Email Detection	Email body text.	KNN, BAYES, RF, LOGIC R, SVM	Hybrid proposed (ACC 0.980 Sensitivity 0.970 Specificity 0.975	Matlab	Basic NLP tasks, TF-IDF	1705 emails	N/A	N/A	The dataset size is small.
A28	[50]	2019	Nazario	Feature Extraction/ Selection + Phishing Email Detection	Email header & body, link, and URL.	RF	PRE 1.00, REC 1.00, F1- 1.00	Python, scikit-learn library	N/A	54724 sentences and 125711 tags.	LDA, NER	N/A	They tweaked the method to work with Czech, but throughout the testing time, the only attacks were in English.
A29	[377]	2014	Selected emails	Feature Extraction/ Selection + Phishing Email Detection	Email header & body, link, and URL.	Calculate the text score	ACC 0.994 PRE 0.996 REC/ Sensitivity 0.993 TP 596 FP 2.00 TN 398 FN 4.00	N/A	Basic NLP tasks.	600 phishing emails	POS	N/A	The dataset size is too small.
A30	[127]	2009	Selected emails (WestPac)	Feature Extraction/ Selection + Phishing	Email header & body, link, and URL.	DT, RF, MLP, NB, SVM	DT (ACC 0.992)	N/A	N/A	659,673 emails	IG	7	Only accuracy of the technique was used in

				Email Detection									evaluating it performance.
A31	[375]	2012	Honey Trap database, Enron	Phishing Email Detection	Email header & body, link, URL & attachment.	AIS	F1- 0.966	N/A	N/A	500 emails	N/A	N/A	Only F1-Score of the technique was used in evaluating it performance.
A32	[38]	2019	Enron, Nazario	Feature Extraction/ Selection + Phishing Email Detection	Email header & body, link, and URL.	NB	PRE 0.950 REC/ Sensitivity 0.910 TP 4545 FP 239 FN 464	Python, scikit-learn library	N/A	1000 phishing emails	Semantic analysis	N/A	The dataset size is small + The authors couldn't try for DL-based methods.
A33	[47]	2020	Selected emails	Feature Extraction/ Selection + Phishing Email Detection	Email body text.	LSTM, SVM, NB, and SGD	ACC 0.9270	Python, NLTK Python library	Basic NLP tasks, POS, Word Embeddings	2394 comments	Word2vec, BoW	N/A	Only accuracy of the technique was used in evaluating it performance.
A34	[11]	2018	IWSPA-AP	Feature Extraction/ Selection + Phishing Email Detection	Email header & body, link, and URL.	RF, AB, NB, DT, SVM	DT (ACC 0.999 PRE 0.994 REC 1.000 F1 0.997	N/A	Basic NLP tasks, TF-IDF	With header (4583) With No header (5700)	SVD, NMF	N/A	The authors couldn't try for DL-based methods.
A35	[39]	2018	IWSPA-AP	Feature Extraction/ Selection + Phishing Email Detection	Email header & body, link, and URL.	RF, AB, NB, DT, SVM	DT (ACC 0.967 PRE 0.883 REC 0.791 F1 0.833	N/A	Basic NLP tasks, TF-IDF	With header (4583) With No header (5700	SVD, NMF	N/A	The authors couldn't try for DL-based methods.
A36	[20]	2018	IWSPA-AP	Feature Extraction/ Selection + Phishing	Email header & body, link, and URL.	CNN	ACC 0.968 TP 3618 TN 496 FP 0.00	Python, Keras	Word Embedding	With header (4583)	N/A	N/A	Performance not compared with other method.

				Email Detection			FN 81			With No header (5700			
A37	[48]	2019	IWSPA-AP	Feature Extraction/ Selection + Phishing Email Detection + Improving/ Optimize Algorithms	Email header & body.	THEMIS (RCNN)	ACC 0.998, REC 0.990, PRE 0.996, F1-0.993, FP 0.043.	Python, Tensor Flow, Keras	Word Embeddings & char-level.	8780 emails	Word2vec, Character level CNN	N/A	Performance not compared with other method.
A38	[42]	2017	SpamAssassin, Nazario	Feature Extraction/ Selection + Phishing Email Detection and Classification	Email header & body, link, and URL.	NN	ACC 0.922 Sensitivity 1.00 Specificity 1.00	Python, Matlab	Word embedding	14,370 emails	Word2Vec	N/A	Performance not compared with other method.
A39	[115]	2021	Nazario	Feature Extraction/ Selection + Phishing Email Detection	Email header & body, link, and URL.	*k*-means	N/A	Python, NLTK & Gensim Python libraries	Basic NLP tasks, TF-IDF	N/A	LDA	6	No standard metric to evaluate its Performance.
A40	[29]	2020	Selected emails	Feature Extraction/ Selection + Phishing Email Detection	Email header & body, link, and URL.	CatBERT, DistilBERT, LSTM, LR	ACC 0.870 AUC 0.989	Python, scikit-learn library	Basic NLP tasks, TF-IDF	Five million emails	BERT and GPT	N/A	Many of the modern phish datasets were not examined.
A41	[30]	2019	Lingspam, PU, CSDMC2010, TREC Spam	Feature Extraction/ Selection + Phishing	Email header & body, link, and URL.	LR, CNN, RNN, LSTM, GRU, CNN-RNN, CNN-	CNN (ACC 0.956 REC 0.992, PRE 0.935,	Python, scikit-learn library,	Basic NLP tasks, TF-IDF, Word embedding	235578 emails	BoW	N/A	The advantageous of time split in dividing the data into train

			Assian, Enron MalwareDomains, MalwareDomainList, JWSPAMSPY, MalwareURL. PhishTank, Open-Phish. Alexa, DMOZ, Majestic.	Email Detection		LSTM, CNN-GR.	F1-0.963, FP 2691, TN 25,916, FN 322, TP 38,861)	Tensor Flow, Keras					and test datasets is not discussed.
A42	[148]	2020	SpamAssassin	Feature Extraction/ Selection + Phishing Email Detection	Email header & body, link, and URL.	SVM	ACC 0.900 PRE 0.900 REC 0.900 F1 0.900	N/A	Basic NLP tasks, TF-IDF	6,047 messages	N/A	N/A	Performance not compared with other technique.
A43	[40]	2020	Enron, UCI Machine Learning Repository	Feature Extraction/ Selection + Phishing Email Detection + Improving/ Optimize Algorithms	Email header & body, link, and URL.	NB, DBN, NN, *EWA*-DBN, *fractional EWA*-DBN	ACC 0.857, Sensitivity 0.8182, Specificity 0.880	Java	Basic NLP tasks, TF-IDF	N/A	N/A	50	Very low performance.
A44	[149]	2016	PhishTank, SpamAssassin, Nazario	Feature Extraction/ Selection + Phishing Email Detection	Email header & body, link, and URL.	SVM, RF, LogiBoost, Multi-Classifier	Multi-Classifer (ACC 0.990 FP 2.1)	Weka	Topic Modelling	5260 emails	NER	61	The authors couldn't try for DL-based methods.

A45	[150]	2010	SpamAssassin, Nazario	Feature Extraction/ Selection + Phishing Email Classification	Email body text.	CWLC, SVM	CWLC (ACC 0.987, PRE 0.9979, REC 0. 9932, F1- 0. 9955, FP 0.15, FN 0.32).	Python	Basic NLP tasks	275587 emails	CWLC	N/A	The authors couldn't try for DL-based methods.
A46	[151]	2012	SpamAssassin, Nazario, and own personal mailboxes	Phishing Email Detection	Email header & body, link, and URL.	SVM	CWLC (ACC 0.987)	N/A	N/A	2461 Messages	N/A	31	Only accuracy of the technique was used in evaluating it performance.
A47	[152]	2014	Enron, Nazario	Feature Extraction/ Selection + Phishing Email Detection	Email header & body, link, and URL.	SVM	(ACC 0.940)	Weka	Basic NLP tasks, TF-IDF	4295 Email	N/A	1545	Only accuracy of the technique was used in evaluating it performance.
A48	[153]	2014	Enron, SpamAssassin	Feature Extraction/ Selection + Phishing Email Detection	Email body text.	SVM	N/A	Weka	Basic NLP tasks, TF-IDF	12502 emails	N/A	N/A	No standard metric to evaluate its Performance.
A49	[378]	2015	Nazario	Feature Extraction/ Selection + Phishing Email Classification	Email body text.	FFM	LCH (0.8039)	Java	POS, Word similarity, WordNet, lemmatization, TF-IDF.	4550 emails	POS	N/A	Performance not compared with other technique.
A50	[154]	2015	SpamAssassin, Nazario	Feature Extraction/ Selection + Phishing	Email header & body, link, and URL.	SVM	ACC 0.9775, PRE 0.9565,	N/A	N/A	1000 emails	N/A	9	The dataset size is small.

				Email Detection			REC 0.990, Error 2.75, FN 1.000, TN 0.955, TP 0.990, FP 4.5.						
A51	[128]	2015	SpamAssassin, Nazario	Feature Extraction/ Selection + Phishing Email Detection	Email header & body, link, and URL.	RF, J48, PART	RF (ACC 0.9887 PRE 0.9951, REC 0.9822, ROC 0.9890, FN 1.78, TN 0.9952, TP 0.9822, FP 0.48)	Java	N/A	5059 emails	N/A	23	The authors couldn't try for DL-based methods.
A52	[27]	2017	Enron, Nazario	Feature Extraction/ Selection + Phishing Email Detection + Improving/ Optimize Algorithms	Email header & body, link, and URL.	CS-SVM, *RBF*, *CS*	ACC 99.52%	N/A	N/A	21455	RBF	23	Only accuracy of the technique was used in evaluating it performance.
A53	[43]	2018	Selected emails	Feature Extraction/ Selection + Phishing Email Detection	Email attachment.	DNNs, AB, DT	ROC curves with > 0.99 AUC	N/A	N/A	5,023,243 malicious	N-gram Histograms, String Length-Hash Features, Byte Entropy Features,	N/A	The authors couldn't try for DL-based methods.

											and Byte Mean-Standard Deviation Features		
A54	[31]	2018	IWSPA-AP, SpamAssassin	Feature Extraction/ Selection + Phishing Email Detection	Email header & body, link, and URL.	DT, NB, LR, NN, SVM, Gaussian NB	SVM (TP 0.830, TN 0.960)	N/A	Basic NLP tasks, TF-IDF	3865 ham emails and 735 phishing emails	N/A	26	The authors couldn't try for DL-based methods.
A55	[12]	2019	Microsoft Office	Phishing Email Detection + Improving/ Optimize Algorithms	Email header & body, link, and URL.	AB,BN, *SMO*, J48	AB (ACC 0.989, AUC 0.997, TPR 0.997, FPR 0.015)	Weka	N/A	610 legitimate emails and 325 targeted malicious emails.	N/A	59	The authors couldn't try for DL-based methods.
A56	[22]	2020	Selected emails	Feature Extraction/ Selection + Phishing Email Detection	Email body text.	DT, DA, LR, SVM, KNN	DT, DA, LR (ACC 100%)	Matlab	Basic NLP tasks	4500 fraudulent emails	BoW	N/A	Only accuracy of the technique was used in evaluating it performance.
A57	[13]	2021	Enron, Kaggle	Feature Extraction/ Selection + Phishing Email Classification	Email header & body, link, and URL.	NB, KNN,DT, RF, SVM	SVM (ACC 0.9783, PRE 0.980, REC 0.990, F1-0.990,)	Python, Gensim	Basic NLP tasks, vectorization	5574 messages	BoW, Word2vec	N/A	The authors couldn't try for DL-based methods.
A58	[51]	2020	Nazario	Phishing Email Detection	Email header & body, link, and URL.	RF	ACC 0.830	N/A	N/A	4348 emails	N/A	N/A	Very low performance.
A59	[139]	2020	Selected emails	Feature Extraction/ Selection +	Email header & body, link, and URL.	RF SVM, NB	SVM (ACC 0.8160,	Python, NLTK &	Basic NLP tasks,	24 emails	Doc2Vec, PCA	N/A	Very low performance.

				Phishing Email Detection			PRE 0.750, REC 0.750, F1- 0.766,)	Gensim Python libraries	vectorization				
A60	[23]	2019	Enron, CMU Corpus, Kaggle, Microsoft Office	Feature Extraction/ Selection + Phishing Email Classification	Email header & body, link, and URL.	KNN, SVM, NB, DT	SVM (ACC 0.990)	N/A	Basic NLP tasks	45000 emails	BoW		Only accuracy of the technique was used in evaluating it performance.
A61	[44]	2019	PU, Custom, UCI Machine Learning Repositor, SpamBase, Enron, SpamAssassin, TREC, CCERT, LingSpam	Phishing Email Classification	Email header & body, link, and URL.	J48, AB, KNN, NB, NN, SVM, RF	RF (ACC 0.8600)	N/A	N/A	182288 Email	N/A	N/A	Very low performance + Only accuracy of the technique was used in evaluating it performance.
A62	[17]	2020	SpamAssassin	Phishing Email Detection	Email header & body, link, and URL.	KNN, DT, Bayes	KNN (ACC 0.972 TP 0.952, FP 0.008, PRE 0.992)	N/A	N/A	2000 Email	IG	25	The dataset size is too small.
A63	[129]	2020	Enron, Nazario	Phishing Email Detection	Email header & body, link, and URL.	RF	ACC 0.967 TP 1676, TN 1804	N/A	N/A	1,234,387 Email	N/A	12	Performance not compared with other technique.
A64	[130]	2019	Enron	Phishing Email classification	Email header & body, link, and URL.	RF, DT, SVM, KNN	RF (ACC 0.9400)	C#	N/A	298 emails	N/A	11	Only accuracy of the technique was used in

													evaluating it performance" + The dataset size is too small.
A65	[49]	2020	TREC, Kaggle	Feature Extraction/ Selection + Phishing Email Classification	Email header & body, link, and URL.	KNN, LSTM	Bi-LSTM-Attention classifier (ACC 0.900)	Python	Basic NLP tasks, vectorization	37822 emails	Word2vec	N/A	Other features will be considered according to the experimental results, such as email title, processed email image information.
A66	[140]	2020	Enron	Feature Extraction/ Selection + Phishing Email Classification	Email header & body, link, and URL.	NB , SVM	SVM (ACC 0.900)	Python	Basic NLP tasks, TF-IDF	50,000 emails	POS	N/A	Only accuracy of the technique was used in evaluating it performance.
A67	[14]	2020	SpamAssassin, Nazario	Feature Extraction/ Selection + Phishing Email Detection	Email header & body, link, and URL.	SVM, NB, LR , KNN , DT, RF, AB, MLP	RF and AB (ACC 1.00, F1-1.00).	Stanza	Basic NLP tasks, TF-IDF	6,429 e-mails	BoW, DTM, PCA, LSA, CS, MI	N/A	Implement DL, language models, and transformers-based methods to detect phishing, with the understanding that their application may offer benefits such as increased resiliency to pre-trained models or the

													ability to use them with languages other than the original dataset's language
A68	[131]	2019	Selected emails (Bangla spam email datasets)	Feature Extraction/ Selection + Phishing Email Detection	Email header & body, link, and URL.	NB, DT, KNN, RF, AB, SVM	RF (ACC 0.936 Sensitivity 0.940, Specificity 0.931).	Python	Basic NLP tasks, TF-IDF	4766 emails	N/A	N/A	The authors couldn't try for DL-based methods.
A69	[379]	2020	Enron	Feature Extraction/ Selection + Phishing Email Detection	Email header & body, link, and URL.	FFNN, BERT	F1- 0.9915	N/A	Basic NLP tasks, TF-IDF	32638 emails	BoW	N/A	Only F-measure of the technique was used in evaluating it performance.
A70	[21]	2020	SpamAssassin, Nazario	Feature Extraction/ Selection + Phishing Email Classification	Email body& subject.	CNN	ACC 0.9942, PRE 0.9880, REC 0.9954, F1-0.9917.	Python, scikit-learn library, Tensor Flow, Keras	Basic NLP tasks	6,428 emails	N/A	N/A	Performance not compared with other technique.
A71	[356]	2014	Nazario, TREC	Feature Extraction/ Selection + Phishing Email Detection	Email header & body, link, and URL.	EM	N/A	EditPad Pro	N/A	8000 emails	N/A	9	No standard metric to evaluate its Performance.
A72	[155]	2016	SpamAssassin, Nazario	Phishing Email Detection + Improving/	Email header & body, link, and URL.	FFA_SVM	ACC 0.9994 PRE 0.9994, REC 0.9992,	C#	N/A	4,000 emails	N/A	16	Performance not compared with other technique.

				Optimize Algorithms			FN 0.08, FP 0.01.						
A73	[24]	2021	Nazario, SpamAssassin, Vilnius Gediminas Technical University (VilniusTech).	Feature Extraction/ Selection + Phishing Email Classification	Email header & body, link, and URL.	NB, GLM , DT, RF, Gradient boosted trees, SVM	SVM (ACC 0.8400 PRE 0.7800 REC 0.9520 F1 0.8560 AUC 52.90)	Python, Rapid Miner	Basic NLP tasks, TF-IDF	1400 (700 spam and 700 phishing emails)	N/A	N/A	Very low performance.
A74	[41]	2019	PhishTank, DMOZ, Alexa	Phishing Email Detection + Improving/ Optimize Algorithms	Email header & body, link, and URL.	NB, SVM	EMUD with SVM (ACC 0.9301 TPR 0.9000 FPR 4.90 PRE 0.9126)	Weka	N/A	2000 phishing URLs and legitimate	N/A	13	The dataset size is small.
A75	[380]	2012	Enron, SpamAssassin, Nazario, SPAM Archive, PhishTank	Feature Extraction/ Selection + Phishing Email Detection	Email header & body, link, and URL.	AB	F-measure of 100%	Weka	Basic NLP tasks, TF-IDF	400,000 emails	POS	N/A	Only F-measure of the technique was used in evaluating it performance.
A76	[381]	2018	Nazario, SpamAssassin, PhishTank	Feature Extraction/ Selection + Phishing Email Detection	Email header & body, link, and URL.	FEaR	ACC 98.6% FP 1.8%	N/A	N/A	Phishing 4559 Legitimate 4559	N/A	50	Performance not compared with other technique.

A77	[132]	2018	Microsoft Office	Feature Extraction/ Selection + Phishing Email Detection	Email attachment.	RoF , RF, AB, Decorate	RoF (FPR 0.017 PRE 0.978 REC 0.968 F1 0.973 AUC 0.995).	Python, Weka	N/A	3.9 million	N/A	N/A	The authors couldn't try for DL-based methods.
A78	[141]	2011	Spambase	Feature Extraction/ Selection + Phishing Email Detection	Email body text.	GMDH, NN, NB	GMDH (ACC 0.917 FPR 5.9 FN 11.8).)	N/A	N/A	2844 cases	N/A	57	The dataset size is small.
A79	[32]	2020	The TWOS dataset	Feature Extraction/ Selection + Phishing Email Detection	Email body text.	AB, NB, LR, KNN, LIR SVM	AB (ACC 0.983, REC 0.980, AUC 0.983, TP 495 , FP 8, FN 9, TN 488).	Python, Tensor Flow	Basic NLP tasks, TF-IDF	Twelve instances of the masquerader,	N/A	N/A	The authors couldn't try for DL-based methods.
A80	[28]	2021	UCI Machine Learning Repository , Kaggle	Feature Extraction/ Selection + Phishing Email Detection	Email body text.	BiLSTM, KNN, NB, BBC	BBC (ACC 0.9867 F1 0.9866).	Python, Keras	Basic NLP tasks, word embedding	5226 emails	N/A	N/A	The phish detection task can be applied to another text language for e.g.: Arabic.
A81	[33]	2018	Enron	Feature Extraction/ Selection + Phishing Email Classification	Email body& subject.	LR, NB, SVM, J48, RF, RBFN	LR (ACC 0.95 REC 0.950, PRE 0.940, F1- 0.950,	Weka	Basic NLP tasks	1000 emails	PCA,CFS	N/A	The dataset size is small.

							).						
A82	[25]	2021	UCI Machine Learning Repository	Feature Extraction/ Selection + Phishing Email Detection	Email body text.	DT, KNN	DT (ACC 0.90 PRE 0.908,, Sensitivity 0.922,, Specificity 0.856 F1- 0.915).	N/A	N/A	3000 emails	IG, GIC	N/A	The authors couldn't try for DL-based methods.
A83	[156]	2012	The industry partners of the Centre for Informatics and Applied Optimization	Feature Extraction/ Selection + Classification of Phishing Email + Improving/ Optimize Algorithms	Email header & body.	AB, Bagging, Dagging, Decorate, Grading, MultiBoost, Stacking, J48, LibSVM, PART, NNge, *SMO*	*SMO* (ACC 0.950)	Weka	Basic NLP tasks, TF-IDF, embedded hyperlinks	3276 emails	N/A	N/A	Only accuracy of the technique was used in evaluating it performance.
A84	[240]	2011	SpamAssassin, Nazario	Feature Extraction/ Selection + Phishing Email Detection"	Email header & body.	BN	ACC 0.960, FP 0.4.	Weka	N/A	6135 emails	N/A	7	To derive the optimal feature set, more features should add and improve.
A85	[157]	2012	Hyperlink Based (H), Hyperlink Suspected Component Based (HS), and Hyperlink	Feature Extraction/ Selection + Phishing Email Detection	A hyperlink.	AB, SVM	N/A	N/A	N/A	5881 emails	N/A	9	No standard metric to evaluate its Performance.

			Template Based (HT)										
A86	[116]	2008	SpamAssassin, Nazario	Feature Extraction/ Selection + Phishing Email Detection Email + Improving/ Optimize Algorithms	Email header & body.	SVM, NN, *SOMs* , *k*-Means	NN (ACC 0.9799).	Python	N/A	2000 emails	N/A	16	Only accuracy of the technique was used in evaluating it performance" + The dataset size is small.
A87	[133]	2013	Spambase Data Set (Spam), SpamAssassin, Nazario	Feature Extraction/ Selection + Classification of Phishing Email	Email header & body.	KP-SVM, SVM-RFE, FSV and Fisher Criterion Score	KP-SVM (ACC 0.9868)	N/A	N/A	4,601 emails	N/A	57	Only accuracy of the technique was used in evaluating it performance.
A88	[253]	2020	Nazario, Csmining Group	Feature Extraction/ Selection + Phishing Email Detection Email	Email header & body, link, and URL.	RF, BSFS	BSFS (ACC 0.9741 PRE 0.9624 REC 0.9967 F1-0.9778)	N/A	Basic NLP tasks, TF-IDF	3428 emails	SFFS, BSFS	11	To derive the optimal feature set, more features should add and improve.
A89	[37]	2020	Ling-Spam, Enron, PUA, and SpamAssassin	Feature Extraction/ Selection + Phishing Email Detection Email + Improving/ Optimize Algorithms	Email header & body, link, and URL.	NB, SVM, RF, DT and MLP with *The bio-inspired and Genetic algorithm*	GA optimization (ACC 1.000 PRE 1.000 REC 1.000)	Weka, Python with Scikit-learn, Keras, and Tensorflow	Basic NLP tasks, TF-IDF, BoW	50,000 emails	CFS	N/A	The phish detection task can be applied to another text language for e.g.: Arabic.

A90	[36]	2020	UBE datasets	Feature Extraction/ Selection + Phishing Email Detection Email + Improving/ Optimize Algorithms	Email header & body, link, and URL.	MLP with (Cuckoo-Firefly-GR)	ACC 0.9978	Python	Syntactic and semantic information	3, 844 emails	Doc2Vec	164, 167, and 172	Only accuracy of the technique was used in evaluating it performance.
A91	[65]	2009	Selected emails	Feature Extraction/ Selection + Classification of Phishing Email	Email body text	DT, NN, NB, SVM, ME, & KNN	SVM-Eng (ACC 0.9903 F1-0.9832) NN-Arb (ACC 0.8977 F1-SVM 0.7625)	N/A	Basic NLP tasks, TF-IDF, BoW	1047 messages	MI	41883 tokens	The dataset size is small + mixed corpus + Very low performance for Arabic messages.
A92	[247]	2018	Selected emails + SpamAssassin	Feature Extraction/ Selection + Classification of Phishing Email	Email header & body, and Subject.	Semi-Automated Feature generation for Phish Classification (SAFE-PC) and RUSBoost	ACC 0.971	Python, C and Sklearn	Basic NLP tasks	A total of 425K phishing and 158K legitimate emails	word stemming, sentence structure analysis	806 features	Only accuracy of the technique was used in evaluating it performance.
A93	[142]	2021	Selected emails	Feature Extraction/ Selection + Classification of Phishing Email	Email header & body	NB	(ACC 0.984, FP 0.08, FN 2.90, PRE 0.99 REC 0.969, and F-measure 0.976)	Java & JADE agent platform	Basic NLP tasks	2,000 emails	MI	N/A	The dataset size is small.

A94	[134]	2018	Selected emails	Feature Extraction/ Selection + Classification of Phishing Email	Email header & body	MNLAS model (RF)	(ACC 0.919, PRE 0.902 REC 0.918, and F-measure 0.932)	Java & JADE agent platform	Basic NLP tasks	200 emails	N/A	N/A	The dataset size is small.
A95	[143]	2022	SpamAssassin	Feature Extraction/ Selection + Classification of Phishing Email	Email header & body	NB, LR, SVM, and ANN	(ACC 0.990, PRE 0.960 REC 0.980, and F-measure 0.970)	Python's sklearn and NLTK library	Basic NLP tasks	2500 non-spam messages	N/A	N/A	The authors couldn't try for DL-based methods.
A96	[120]	2022	Enron-Spam, Spambase, and TREC Spam	Feature Extraction/ Selection + Classification of Phishing Email	Email header & body	Semantic Graph Neural Network (SGNN) and CNN	Enron Spam (ACC 0.9887	Python's Glove	Basic NLP tasks	596790 emails	N/A	N/A	Only accuracy of the technique was used in evaluating it performance.
A97	[117]	2022	Spam Assassin dataset	Feature Extraction/ Selection + Classification of Phishing Email	Email header & body	NB, LR, DT, SVM, RF, GB, AB, Bagging, and LSTM-CNN model.	BOW (ACC 0.980, and F-measure 0.980)	N/A	Basic NLP tasks	15736 emails	BoW and TF-IDF	N/A	They did not apply cross-validation.
A98	[118]	2021	Phish corpus and Spam Assassin dataset	Feature Extraction/ Selection + Classification of Phishing Email	Email header & body	RCNN	ACC 0.998, FPR 0.042%.	Python and Matlab	Word Embeddings & char-level.	N/A	Common Bag of Words (CBOW) as Multi-level Word2Vec	N/A	Performance not compared with other technique.
A99	[119]	2021	Enterprise email samples	Feature Extraction/ Selection +	Email header & body, link, and URL.	NB, RF, XGBoost,	REC 0.990	Python's scikit-	Basic NLP tasks	377K of emails	Word2vec and BERT	63 features	Only Recall of the technique was used in

				Classification of Phishing Email		SVM, and CNN-LSTM		learn librar					evaluating it performance.
A100	[382]	2021	Fraud dataset	Feature Extraction/ Selection + Classification of Phishing Email	Email body	Graph convolutional network (GCN)	BOW (ACC 0.982, PRE 0.985 REC 0.983, and F-measure 0.985)	Python	Basic NLP tasks	3685 phishing Emails and 4894 legitimate Emails	N/A	N/A	The dataset size is limited.

TABLE A2: SOURCE OF DATA.

No.	Dataset	Publicly?	Type	Link	Ref.	Used by	Article ID	Count
1	First Security and Privacy Analytics Anti-Phishing Shared Task (IWSPA-AP)	No	Legitimate and Phishing Emails	https://dasavisha.github.io/IWSPA-sharedtask	[199]	[11], [20], [31], [39], [45], [48]	A21, A34, A35, A36, A37, A54	6
2	Nazario's phishing corpora (Nazario)	Yes	Phishing emails	https://monkey.org/~jose/phishing/	[200]	[14], [21], [50], [51], [115], [116], [118], [122]–[124], [128], [129], [24], [133], [137], [138], [146], [147], [149]–[152], [154], [27], [155], [179], [183], [192], [238], [240], [253], [356], [376], [378], [34], [380],	A1, A2, A5, A6, A7, A8, A9, A10, A11, A12, A13, A14, A15, A16, A21, A23, A28, A32, A38, A39, A44, A45, A46, A47, A49, A50, A51, A52, A58, A63, A67, A70, A71, A72, A73, A75, A76, A84, A86, A87, A88, A98	42

						[381], [35], [38], [42], [45], [46]		
3	Enron Email Dataset (Enron)	Yes	Legitimate and Phishing Emails	http://www.cs.cmu.edu/~enron/	[201]	[13], [18], [45], [46], [120], [129], [130], [140], [144], [152], [153], [238], [23], [375], [379], [380], [27], [30], [33], [37], [38], [40], [44]	A1, A3, A21, A23, A25, A31, A32, A41, A43, A47, A48, A52, A57, A60, A61, A63, A64, A66, A69, A75, A81, A89, A96	23
4	The Spam Assassin project	Yes	Legitimate and Phishing Emails	http://spamassassin.apache.org/publiccorpus/	[202]	[14], [17], [44], [46], [116]–[118], [122]–[124], [128], [133], [21], [137], [138], [143], [144], [146]–[151], [24], [153]–[155], [179], [192], [240], [247], [376], [380], [381], [30], [31], [34], [35], [37], [42]	A3, A5, A6, A7, A8, A9, A10, A11, A12, A13, A14, A15, A16, A23, A38, A41, A42, A44, A45, A46, A48, A50, A51, A54, A61, A62, A67, A70, A72, A73, A75, A76, A84, A86, A87, A89, A92, A95, A97, A98	40
5	Phish Tank	Yes	Phishing URLs	http://www.phishtank.com	[97]	[30], [41], [149], [192], [380], [381]	A16, A41, A44, A61, A74, A75, A76	7
6	Lingspam	Yes	Legitimate Emails	http://www.aueb.gr/users/ion/data/lingspam_public.tar.gz.	[203]	[30], [37]	A41, A89	2
7	PU	Yes	Legitimate and Phishing Emails	http://www.aueb.gr/users/ion/data/PU123ACorpora.tar.gz.	[204]	[30], [37]	A41, A89	2

8	MalwareDomains	Yes	Malicious URL	http://www.malwaredomains.com/.	[205]	[30]	A41	1
9	MalwareDomainList	Yes	Malicious URL	https://www.malwaredomainlist.com/.	[206]	[30]	A41	1
10	JWSPAMSPY	Yes	Malicious URL	http://www.joewein.de/sw/blacklist.htm.	[207]	[30]	A41	1
11	MalwareURL	Yes	Malicious URL	https://www.malwareurl.com/.	[208]	[30]	A41	1
12	Open-Phish	Yes	Phishing URLs	https://openphish.com/.	[209]	[30]	A41	1
13	CCERT	Yes	Legitimate and Phishing Emails	N/A	N/A	[44]	A61	1
14	SPAM Archive	Yes	Legitimate and Phishing Emails	http://untroubled.org/spam/	[210]	[192], [380]	A16, A75	2
15	Phishery	Yes	Phishing messages	http://phishery.internetdefence.net/	[383]	[125]	A18	1
16	CSIRO (private emails)	Yes	Phishing emails	https://it.cornell.edu/phish-bowl	[384]	[45]	A21	1
17	Phishbowl	Yes	Phishing emails	http://dev.null.org/dadaengine/	[385]	[45]	A21	1
18	The Short message service Spam Collection v.1 7	Yes	Legitimate and Phishing messages	https://archive.ics.uci.edu/ml/datasets/sms+spam+collection	[226]	[26]	A22	1

19	BC3	Yes	Legitimate emails	N/A	[386]	[18]	A25	1
20	Alexa	Yes	Legitimate URL	https://www.alexa.com/siteinfo.	[387]	[19], [30], [41]	A26, A41, A74	3
21	DMOZ	Yes	Legitimate URL	http://www.dmoz.org/.	[388]	[19], [30], [41]	A26, A41, A74	3
22	Majestic	Yes	Legitimate URL	https://github.com/rlilojr/Detecting-Malicious-URL-Machine-Learning.	[389]	[30]	A41	1
23	Honey Trap database	Yes	Phishing emails	millersmiles.co.uk	[390]	[375]	A31	1
24	Anti-Phishing Working Group website (APWG)	Yes	Phishing/Malware	www.antiphishing.org	[391]	[18]	A25	1
25	The industry partners of the Centre for Informatics and Applied Optimization (3276 emails)	Yes	Phishing emails	N/A	N/A	[113], [156]	A19, A83	2
26	Hyperlink Based (H)	Yes	Phishing emails (hyperlink)	N/A	N/A	[157]	A85	1
27	Hyperlink Suspected Component Based (HS))	Yes	Phishing emails (hyperlink)	N/A	N/A	[157]	A85	1

			Suspected part					
28	Hyperlink Template Based (HT)	Yes	Phishing emails (hyperlink)	N/A	N/A	[157]	A85	1
29	Spambase Data Set (Spam)	Yes	Phishing emails	http://mlearn.ics.uci.edu/databases/spambase/.		[44], [120], [133], [141]	A61, A78, A87, A96	4
30	Csmining group	Yes	Legitimate emails	http://csmining.org/index.php/spam-email - datas ets-.html.	[227]	[30], [253]	A41, A88	2
31	Microsoft Office	Yes	Malicious File	N/A	N/A	[12], [23], [132]	A55, A60, A77	3
32	The TWOS dataset	Yes	Legitimate user data and malicious insider instances (masquerades and traitors).	N/A	[392]	[32]	A79	1
33	UCI machine learning repository	Yes	Legitimate and Phishing Emails	http://archive.ics.uci.edu/ml.	[393]	[25], [28], [40], [44]	A43, A61, A80, A82	4

34	UBE datasets	Yes	Legitimate and Phishing Emails	N/A	N/A	[36]	A90	1
35	Kaggle	Yes	Malicious File	https://www.kaggle.com/karthickveerakumar/spam-filter.	[394]	[13], [23], [28], [49]	A57, A60, A65, A80	4
36	TREC corpus	Yes	Legitimate and Phishing Emails	http://trec.nist.gov/data/spam.html	[221], [395]	[30], [44], [49], [120], [125], [356]	A18, A41, A61, A65, A71, A96	6
37	CMU Corpus	Yes	Legitimate and Phishing Emails	N/A	N/A	[23]	A60	1
38	LingSpam	Yes	Legitimate and Phishing Emails	N/A	N/A	[44]	A61	1
39	Fraud dataset	No	N/A	N/A	N/A	[382]	A100	1

TABLE A3: METHODS USED IN PHISHING EMAIL DETECTION

Type	Algorithm	Ref.	Article ID	Count
Supervised	AB	[11], [12], [123], [124], [126], [131], [132], [138], [156], [157], [192], [380], [13], [14], [32], [39], [43], [44], [117], [119]	A7, A9, A15, A16, A24, A34, A35, A53, A55, A57, A61, A67, A68, A75, A77, A79, A83, A85, A97, A99	20
	Bayes Multinomial classifier (BM)	[125]	A18	1

	BN	[12], [15]–[17], [124], [240]	A9, A17, A27, A55, A62, A84	6
	DT / J48/ C5.0 algorithm	[11], [12], [31], [33], [35], [37], [39], [43], [44], [65], [117], [124], [13], [125]–[128], [130], [131], [137], [156], [179], [376], [14], [15], [17], [22]–[25]	A6, A9, A10, A1, A17, A18, A24, A30, A34, A35, A51; A53, A54, A55, A56, A57, A60, A61, A62, A64, A67, A68, A73, A81, A82, A83, A89, A91, A97	29
	Decision Table (DT)	[26], [123], [125]	A7, A18, A22	3
	Decorate	[132], [156]	A77, A83	2
	KNN	[13], [14], [44], [49], [65], [125], [126], [130], [131], [137], [15]–[17], [23], [25], [26], [28], [32]	A14, A17, A18, A22, A24, A27, A57, A60, A61, A62, A64, A65, A67, A68, A79, A80, A82, A91	18
	Logistics Regression (LR)	[14], [18], [117], [126], [137], [143], [19], [22], [26], [29]–[33]	A14, A22, A24, A25, A26, A40, A41, A54, A56, A67, A79, A81, A95, A97	14
	NB / Gaussian Naive Bayes	[11], [14], [34], [35], [37]–[41], [44], [47], [65], [18], [117], [119], [123], [126], [127], [131], [137]–[140], [23], [141]–[143], [24], [26], [28], [31]–[33]	A7, A8, A11, A14, A15, A22, A24, A25, A30, A32, A33, A34, A35, A43, A54, A59, A60, A61, A66, A67, A68, A73. A74, A78, A79, A80, A81, A89, A91, A93, A95, A97, A99	33
	Radom Forest (RF)	[11], [13], [39], [44], [50], [51], [117], [119], [122]–[125], [14], [126]–[134], [15], [16], [24], [26], [33], [35], [37]	A5, A7, A9, A11, A17, A18, A22, A24, A27, A28, A30, A34, A35 , A57, A58, A59, A61, A63, A64, A67, A68, A73, A77, A81, A88, A89, A94, A99	29
	Rotation Forest (RoF)	[132]	A77	1
	RNN/ LSTM /RCRR	[18], [19], [118], [119], [29], [30], [45]–[49], [117]	A21, A23, A25, A26, A33, A37, A40, A41, A65, A97, A98, A99	12
	BiLSTM	[28]	A80	1
	CNN	[18]–[21], [120]	A25, A26, A36, A70, A96	5

	CNN-RNN	[19], [30]	A26, A41	2
	CNN-GRU	[30]	A41	1
	GRU	[19], [30]	A26, A41	2
	Bert Base Cased (BBC)	[28]	A80	1
	Stacking	[156]	A83	1
	SVM	[11], [13], [27], [31]–[35], [37], [39], [41], [44], [14], [47], [65], [116], [117], [119], [125]–[127], [130], [131], [16], [133], [137]–[140], [143]–[147], [18], [148]–[157], [19], [22]–[24], [26]	A3, A4, A8, A11, A12, A13, A14, A15, A18, A22, A24, A25, A26, A27, A30, A33, A34, A35, A42, A44, A45, A46, A47, A48, A50, A52, A54; A56, A57, A59, A60, A61, A64, A66, A67, A68, A72, A73, A74, A79, A81, A83, A85, A86, A87, A89, A91, A95, A97, A99	50
	LIR	[26], [32]	A22, A79	2
	NN	[31], [40], [42]–[44], [65], [116], [141], [143]	A38, A53, A54, A61, A78; A86, A91, A95	8
	Back-propagation (BP)	[18]	A25	1
	Deep belief network (DBN)	[40]	A43	1
Unsupervised	k-Means Clustering	[22], [34], [113]–[116]	A8, A19, A20, A39, A56, A86	6
	EM Clustering	[356]	A71	1
Online Learning	MLP	[14], [34]–[37], [127]	A8, A11, A30, A67, A89, A90	6
Rule/Pattern based	PART	[128], [156]	A51, A83	2
	LOGIC R	[16]	A27	1
Others	Bagging	[117], [179]	A6, A97	2
	Boosting	[156]	A83	1
	OneR	[123]	A7	1

	GMDH-based learning approach	[141]	A78	1
	Feature Evaluation and Reduction (FEaR)	[381]	A76	1
	Gradient boosted trees	[24]	A73	1
	PhishNet-NLP	[183]	A2	1
	Action-detector and Nonsensical-detector (ADND)	[238]	A1	1
	Sequential Minimal Optimization (SMO)	[12], [15], [116], [144], [156]	A3, A17, A55, A83, A86	5
	Profiling email-born phishing (ProEP)	[123]	A7	1
	SPOOFNet	[19]	A26	1
	Artificial Immune Systems (AIS)	[375]	A31	1
	Fractional-earthworm optimization algorithm (FEWA)	[40]	A43	1
	LogiBoost	[149]	A44	1
	Five Factor Model (FFM)	[378]	A49	1
	Discriminant Analysis (DA)	[22]	A56	1
	Feed Forward Neural Network (FFNN)	[379]	A69	1
	Generalized Linear Model (GLM)	[24]	A73	1
	Semi-Automated Feature generation for Phish Classification (SAFE-PC) and RUSBoost	[247]	A92	1
	Semantic Graph Neural Network (SGNN)	[120]	A96	1
	Graph convolutional network (GCN)	[382]	A100	1

TABLE A4: OPTIMISATIONS TECHNIQUES USED IN PHISHING EMAIL DETECTION.

Algorithm	Ref.	Article ID	Count
SMO	[12], [15], [116], [144], [156]	A3, A17, A55, A83, A86	5
FEWA	[40]	A43	1
The multi-start modified global *k*-means (MS-MGKM)	[114]	A20	1
The incremental nonsmooth optimization clustering algorithm (INCA)	[114]	A20	1
An algorithm based on difference of convex representation of clustering functions (DCClust)	[114]	A20	1
RNN with Adam optimizer	[46]	A23	1
Refined Feature Matrix	[127]	A30	1
Adam optimizer	[21], [28]–[30], [47], [379]	A33, A40, A41, A69, A70, A80	6
Hessian-free optimization	[30]	A41	1
Radial Basis Function (RBF)	[27]	A52	1
CS	[27]	A52	1
Firefly algorithm (FFA)	[155]	A72	1
Genetic Algorithm	[37]	A89	1
The bio-inspired algorithms (Particle Swarm Optimization (PSO))	[37]	A89	1

TABLE A5: DISTRIBUTION OF METRICS USED FOR EVALUATING PHISHING EMAIL DETECTION

Eval. Metrics	Ref.	Article ID	Count
Accuracy	[11], [12], [22]–[29], [32], [33], [13], [34]–[37], [39]–[42], [44], [45], [14], [46]–[49], [51], [113], [114], [116]–[118], [16], [120], [122]–[130], [17], [131], [133], [134], [137]–[143], [18], [144]–[152], [154], [19], [155], [156], [179], [183], [238], [240], [247], [253], [376], [377], [20], [381], [382], [21]	A1, A2, A3, A4, A5, A6, A7, A8, A9, A10, A11; A12, A13, A14, A15, A18, A19, A20, A21, A22, A23, A24, A25, A26, A27; A29, A30, A33, A34, A35, A36, A37, A38, A40, A41, A42, A43, A44, A45, A46, A47, A50, A51, A52, A55, A56, A57, A58, A59, A60, A61, A62, A63, A64, A65, A66, A67, A68, A70, A72, A73, A74, A76, A78, A79, A80, A81, A82, A83, A84, A86, A87, A88, A89, A90, A92, A93, A94, A95, A96, A97, A98, A100	83
Area Under Curve (AUC)/ ROC	[12], [24], [29], [32], [35], [43], [128], [132], [179], [192]	A6, A11, A16, A40, A51, A53, A55, A73, A77, A79	10
Confusion Matrix ((True Positive, False Positive, True Negative, False Negative, TPR, FPR, TNR, FNR, Specificity, and Sensitivity).	[12], [16], [35], [38], [40]–[42], [46], [48], [118], [122], [125], [17], [128], [129], [131], [132], [138], [141]–[143], [146], [149], [20], [150], [154], [155], [192], [238], [240], [377], [381], [25], [26], [30]–[32], [34]	A1, A5, A8, A11; A12, A15, A16, A22, A23, A27, A29, A32, A36, A37, A38, A41, A43, A44, A45, A50, A51, A54, A55, A62, A63, A68, A72, A74, A76, A77, A78, A79, A82, A84, A93, A95, A98	36
Error Rate	[146], [154]	A12, A50	2
F-score	[11], [13], [30], [33]–[35], [37], [39], [46], [48], [50], [65], [14], [117], [126], [132], [134], [139], [142], [143], [146], [148], [150], [15], [154], [192], [253], [375], [377], [379], [380], [382], [19], [21], [24]–[26], [28]	A8, A11, A12, A16, A17, A22, A23, A24, A26, A28. A29, A31, A34, A35, A37, A41, A42, A45, A50, A57, A59, A67, A69, A70, A73, A75, A77, A80, A81, A82, A88,	38

		A89, A91, A93, A94, A95, A97, A100	
Recall	[11], [13], [34]–[36], [38], [39], [46], [48], [50], [65], [119], [19], [128], [132], [134], [139], [142], [143], [146], [148], [150], [154], [21], [155], [192], [253], [377], [382], [24]–[26], [30], [32], [33]	A8, A11, A12, A16, A22, A23, A26, A28, A29, A32, A34, A35, A37, A41, A42, A45, A50, A51, A57, A59, A70, A72, A73, A77, A79, A81, A82, A88, A90, A91, A93, A94, A95, A99, A100	35
Precision	[11], [13], [34]–[36], [38], [39], [46], [48], [50], [65], [128], [17], [132], [134], [139], [142], [143], [146], [148], [150], [154], [155], [19], [192], [253], [377], [382], [21], [24]–[26], [30], [33]	A8, A11, A12, A16, A22, A23, A26, A28, A29, A32, A34, A35, A37, A41, A42, A45, A50, A51, A57, A59, A62, A70, A72, A73, A77, A81, A82, A88, A90, A91 A93, A94, A95, A100	34

TABLE A6: TOOLS.

Tool	Ref.	Article ID	Count
Perl	[183]	A2	1
WordNet	[183]	A2	1
Sense Learner	[183]	A2	1
Stanza	[14]	A67	1
Stanford Core NLP	[145]	A4	1
C#	[122], [130], [155], [247]	A5, A64, A72, A92	4
Weka	[12], [15], [152], [153], [156], [179], [192], [240], [380], [33], [37], [41], [123]–[125], [132], [149]	A6, A7, A9, A16, A17, A18, A44, A47, A48, A55, A74, A75, A77, A81, A83, A84, A89	17

Python	[13], [15], [30], [32], [34], [36]–[38], [42], [45], [47], [48], [18], [49], [50], [113], [115], [116], [118]–[120], [126], [131], [19], [132], [139], [140], [143], [150], [247], [382], [20], [21], [24], [26], [28], [29]	A8, A17, A19, A21, A22, A24, A25, A26, A28, A32, A33, A36, A37, A38, A39, A40, A41, A45, A57, A59, A65, A66, A68, A70, A73, A77, A79, A80, A86, A89, A90, A92, A95, A96, A98, A99, A100	37
Java	[34], [35], [40], [128], [134], [142], [192], [378]	A8, A11, A16, A43, A49, A51, A93, A94	8
libSVM-library	[146]	A12	1
The Spoofguard	[147]	A13	1
Netcraft	[147]	A13	1
TensorFlow	[21], [30], [32], [37], [45], [48]	A21, A37, A41, A70, A79, A89	6
Keras	[18]–[21], [28], [30], [37], [45], [48]	A21, A25, A26, A36, A37, A41, A70, A80, A89	9
Scikit-learn library	[19], [21], [143], [247], [26], [29], [30], [37], [38], [50], [119], [126]	A22, A24, A26, A28, A32, A40, A41, A70, A89, A92, A95, A99	12
NLTK library	[47], [115], [126], [139], [143]	A24, A33, A39, A59, A95	5
Gensim	[13], [18], [115], [139]	A25, A39, A57, A59	4
Matlab	[16], [22], [42], [118]	A27, A38, A56, A98	4
EditPad Pro	[356]	A71	1
RapidMiner	[24]	A73	1

TABLE A7: FEATURE EXTRACTION/SELECTION.

Feature	Ref.	Article ID	Count
General semantic feature selection	[38], [238], [247]	A1, A32, A92	3
General lexical, syntactic and structural feature selection	[145]	A4	1
Information gained (IG)	[17], [25], [34], [35], [122]–[125], [179], [376]	A5, A6, A7, A8, A9, A10, A11, A18, A62, A82	10
CS	[14], [179]	A6, A8, A67	3
PCA	[14], [33], [34], [126], [139], [179]	A6, A8, A24, A59, A67, A81	6
LSA	[14], [34], [179]	A6, A8, A67	3
Entropy	[34], [376]	A8, A10	2
Gain Ratio (GR)	[124]	A9	1
Symmetrical Uncertainty (SU)	[124]	A9	1
LTM	[146]	A15	1
DMC	[146]	A15	1
Conditional Random Field (CRF)	[192]	A16	1
LDA	[50], [115], [192]	A16, A28, A39	3
Character level CNN	[45], [48], [118]	A21, A37, A98	3
Word-level email	[45], [118]	A21, A98	2
BoW	[13], [14], [117], [118], [379], [18], [19], [22], [23], [30], [37], [47], [114]	A20, A25, A26, A33, A41, A56, A57, A60, A67, A69, A89, A97, A98	13
Doc2Vec	[36], [126], [139]	A24, A59, A90	3
Word2vec	[13], [18], [19], [42], [47]–[49], [118], [119]	A25; A26, A33, A37, A38, A57, A65, A98, A99	9
FastText	[19]	A26	1

NER	[50], [149]	A28, A44	2
POS	[140], [377], [378], [380]	A29, A49; A66, A75	4
Singular value decomposition (SVD)	[11], [39]	A34, A35	2
Non-negative Matrix Factorization (NMF)	[11], [39]	A34, A35	2
Bidirectional Encoder Representations from Transformers (BERT)	[29]	A40	1
GPT model	[29]	A40	1
Confidence-Weighted Learning (CWLC)	[150]	A45	1
Radial Basis Function Networks (RBFN)	[27]	A52	1
N-gram Histograms	[43]	A53	1
String Length-Hash Features	[43]	A53	1
Byte Entropy Features	[43]	A53	1
Byte Mean-Standard Deviation Features	[43]	A53	1
MI	[14], [142]	A67, A93	2
DTM	[14]	A67	1
Correlation Feature Selection (CFS)	[33], [37]	A81, A89	2
Gini impurity criterion (GIC)	[25]	A82	1
Sequential Forward Feature Selection (SFFS)	[253]	A88	1
BSFS	[253]	A88	1
Stochastic Gradient Descent (SGD)	[47]	A33	1

TABLE A8: QUALITY ASSESSMENT RESULTS.

Study	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Total	Percentage
A1	1	1	1	1	1	1	1	7	100%
A2	1	1	1	0.5	0.5	0.5	1	5.5	79%
A3	1	1	1	0.5	1	0.5	1	6	86%
A4	1	1	1	0.5	0	0.5	1	5	71%
A5	1	1	0.5	1	1	0.5	1	6	86%
A6	1	1	1	1	1	1	0.5	6.5	93%
A7	1	1	1	0.5	1	1	1	6.5	93%
A8	1	0.5	1	1	1	1	1	6.5	93%
A9	1	1	1	0.5	1	1	0.5	6	86%
A10	1	1	1	0.5	1	1	1	6.5	93%
A11	1	1	1	1	1	1	0.5	6.5	93%
A12	1	1	1	1	1	1	1	7	100%
A13	1	1	1	0.5	1	1	1	6.5	93%
A14	1	1	1	1	1	0.5	0.5	6	86%
A15	1	1	1	1	0	0.5	1	5.5	79%
A16	1	1	1	1	1	1	1	7	100%
A17	1	1	1	0.5	0.5	0.5	0.5	5	71%
A18	1	1	1	1	1	1	1	7	100%
A19	1	1	1	0.5	1	0.5	1	6	86%
A20	1	1	1	0.5	0.5	1	1	6	86%
A21	1	1	1	0.5	1	1	0.5	6	86%
A22	1	0.5	1	1	1	0.5	0.5	5.5	79%
A23	1	1	1	1	1	0.5	1	6.5	93%
A24	1	1	1	1	0	1	1	6	86%
A25	1	1	1	0.5	1	1	1	6.5	93%
A26	1	1	1	1	1	1	1	7	100%
A27	1	1	1	1	0.5	0.5	1	6	86%
A28	1	0.5	1	1	1	1	0.5	6	86%
A29	1	1	0	1	0.5	1	0.5	5	71%
A30	1	1	1	0.5	1	1	1	6.5	93%
A31	1	1	0	0.5	0.5	0.5	1	4.5	64%
A32	1	1	1	1	0.5	1	0.5	6	86%
A33	1	1	1	0.5	1	1	1	6.5	93%
A34	1	1	1	1	1	1	0.5	6.5	93%
A35	1	1	1	1	1	1	1	7	100%

A36	1	1	1	1	1	0.5	0.5	6	86%
A37	1	1	1	1	1	1	1	7	100%
A38	1	1	1	1	1	1	1	7	100%
A39	1	1	1	0	0	1	1	5	71%
A40	1	1	1	1	1	1	1	7	100%
A41	1	1	1	1	1	1	1	7	100%
A42	0.5	0.5	0.5	1	1	0.5	0	4	57%
A43	1	1	1	1	0	0.5	1	5.5	79%
A44	1	1	1	1	1	1	0.5	6.5	93%
A45	1	1	1	1	1	1	1	7	100%
A46	1	1	1	0.5	1	0.5	1	6	86%
A47	1	1	1	0.5	1	0.5	1	6	86%
A48	1	1	1	0	1	0.5	1	5.5	79%
A49	1	1	0	0.5	1	1	1	5.5	79%
A50	1	1	1	1	0.5	0.5	1	6	86%
A51	1	1	1	1	1	0.5	1	6.5	93%
A52	1	1	1	0.5	1	0.5	1	6	86%
A53	1	1	1	1	1	1	1	7	100%
A54	1	1	1	1	1	0.5	1	6.5	93%
A55	1	1	1	1	0.5	0.5	1	6	86%
A56	1	1	1	1	1	1	1	7	100%
A57	1	1	1	1	1	1	1	7	100%
A58	1	1	1	0.5	1	0.5	1	6	86%
A59	1	1	1	1	0	1	1	6	86%
A60	1	1	1	0.5	1	1	1	6.5	93%
A61	1	1	1	0.5	1	0.5	1	6	86%
A62	1	1	1	1	1	0.5	1	6.5	93%
A63	1	1	1	1	1	0.5	1	6.5	93%
A64	1	1	1	0.5	0	0.5	1	5	71%
A65	1	1	1	1	1	1	1	7	100%
A66	1	1	1	0.5	1	1	1	6.5	93%
A67	1	1	1	1	1	1	1	7	100%
A68	1	1	1	1	1	0.5	1	6.5	93%
A69	1	1	1	0.5	1	1	1	6.5	93%
A70	1	1	1	1	1	0.5	1	6.5	93%
A71	1	1	1	0	1	0.5	1	5.5	79%
A72	1	1	1	1	1	0.5	1	6.5	93%
A73	1	1	1	1	0.5	0.5	1	6	86%

A74	1	1	1	1	1	0.5	1	6.5	93%
A75	1	1	1	0.5	1	1	1	6.5	93%
A76	1	1	0	0.5	1	0.5	1	5	71%
A77	1	1	1	1	1	0.5	1	6.5	93%
A78	1	1	1	1	1	0.5	1	6.5	93%
A79	1	1	1	1	0	0.5	1	5.5	79%
A80	1	1	1	1	1	0.5	1	6.5	93%
A81	1	1	1	1	0.5	0.5	1	6	86%
A82	1	1	1	1	1	0.5	1	6.5	93%
A83	1	1	1	0.5	1	0.5	1	6	86%
A84	1	1	1	0.5	1	0.5	1	6	86%
A85	1	1	1	0	1	0.5	1	5.5	79%
A86	1	1	1	0.5	1	0.5	0.5	5.5	79%
A87	1	1	1	0.5	1	0.5	1	6	86%
A88	1	1	1	1	1	1	1	7	100%
A89	1	1	1	1	1	1	1	7	100%
A90	1	1	1	0.5	1	1	1	6.5	93%
A91	1	1	1	0.5	0.5	0.5	0.5	5	71%
A92	1	1	0.5	1	1	1	1	6.5	93%
A93	1	1	1	0.5	1	0.5	1	6	86%
A94	1	1	1	1	0	0.5	1	5.5	79%
A95	1	1	1	1	0.5	0.5	1	6	86%
A96	1	1	0.5	1	1	1	1	6.5	93%
A97	1	1	1	1	1	1	1	7	100%
A98	1	1	1	1	1	1	1	7	100%
A99	1	1	1	1	1	1	1	7	100%
A100	1	1	1	1	0	0.5	1	5.5	79%

TABLE A9: SNAPSHOT OF ENGLISH–ARABIC PARALLEL TEXT CORPUS FOR PHISHING EMAIL

No.	English Text	Arabic Text (Humans)	Arabic Text (Machine)
1	You have 1 new Important security notification regarding 2017 payroll schedule. View Message Now.	لديك إشعار أمان واحد جديد مهم بخصوص جدول الرواتب لعام 2017. أعرض الرسالة الآن.	لديك 1 إشعار أمان مهم جديد فيما يتعلق بجدول الرواتب لعام 2017. عرض الرسالة الآن.
2	You have new messages for your organization account. Continue here now to receive your new messages. If no action is taken in less than 24 hours, all new messages will be permanently deleted on our database Have a great day!	لديك رسائل جديدة لحساب مؤسستك. تواصل هنا الآن لتلقي رسائلك الجديدة. إذا لم يتم اتخاذ أي إجراء في أقل من 24 ساعة ، فسيتم حذف جميع الرسائل الجديدة نهائيًا من قاعدة البيانات الخاصة بنا أتمنى لك يوما عظيما	لديك رسائل جديدة لحساب مؤسستك. تابع هنا الآن لتلقي رسائلك الجديدة. إذا لم يتم اتخاذ أي إجراء في أقل من 24 ساعة ، حذف جميع الرسائل الجديدة نهائيا من قاعدة بياناتنا !أتمنى لك يوما رائعا
3	Our record shows that your Mailbox is Out-dated which has caused some incoming mails to be placed on pending. Kindly Click Here to update your Mailbox in order to be able to receive new mails. We apologies for any inconvenience this might cause	يُظهر سجلنا أن صندوق البريد الخاص بك قديم مما تسبب في وضع بعض الرسائل الواردة في الانتظار. يرجى النقر هنا لتحديث صندوق البريد الخاص بك لتتمكن من استقبال رسائل بريد إلكتروني جديدة نحن نعتذر عن أي إزعاج قد يسببه هذا الأمر	يظهر سجلنا أن صندوق البريد الخاص بك قديم مما تسبب في وضع بعض الرسائل الواردة في وضع معلق. يرجى النقر هنا لتحديث صندوق البريد الخاص بك حتى تتمكن من تلقي رسائل البريد الجديدة. نعتذر عن أي إزعاج قد يسببه ذلك
4	You have (2) important unread messages, Click on review read it.	لديك (2) رسائل مهمة غير مقروءة ، انقر فوق مراجعة لقراءتها	لديك (2) رسائل مهمة غير مقروءة ، انقر فوق مراجعة اقرأها.
5	Your mailbox has exceeded the storage limit 1 GB, which is defined by the administrator, you are running at 99.8 gigabytes, you cannot send or receive new messages until you re-validate your mailbox. To renew the mailbox, Click Here	صندوق البريد الخاص بك قد تجاوز حد التخزين 1 جيجا بايت ، والذي تم تحديده من قبل المسؤول ، أنت تعمل على 99.8 جيجا بايت ، لا يمكنك إرسال أو استقبال رسائل جديدة حتى تقوم بإعادة التحقق من صندوق البريد الخاص بك. لتجديد صندوق البريد ، اضغط هنا	تجاوزت علبة البريد الخاصة بك حد التخزين 1 غيغابايت، الذي يحدده المسؤول، وأنت تعمل بسرعة 99.8 غيغابايت، ولا يمكنك إرسال رسائل جديدة أو تلقيها حتى تقوم بإعادة التحقق من صحة علبة البريد الخاصة بك. لتجديد صندوق البريد، انقر هنا
6	This organization Account is Subject to mandatory upgrade, Failure to comply would lead to Permanent closure of your account. Upgrade Account Now	يخضع حساب المؤسسة هذا للترقية الإلزامية ، وسيؤدي عدم الامتثال إلى الإغلاق الدائم لحسابك. قم بترقية الحساب الآن.	يخضع حساب المؤسسة هذا للترقية الإلزامية ، وسيؤدي عدم الامتثال إلى إغلاق حسابك بشكل دائم. ترقية الحساب الآن
7	To whom it may concern: Please contact your financial institution to get the necessary updates of the Direct Deposit software.	إلي من يهمه الامر يرجى الاتصال بمؤسستك المالية للحصول على التحديثات اللازمة لبرنامج الإيداع المباشر.	:لمن يهمه الأمر يرجى الاتصال بمؤسستك المالية للحصول على التحديثات اللازمة لبرنامج الإيداع المباشر.
8	You have used 98.9% of the total data allocated to your mailbox. To avoid placing your incoming messages on hold or loose them	لقد استخدمت 98.9% من إجمالي البيانات المخصصة لصندوق البريد الخاص بك. لتجنب وضع رسائلك الواردة قيد الانتظار أو فقدانها بشكل دائم ، نطلب منك إعادة التحقق من صحة صندوق البريد الخاص بك لتوسيع حجم تخصيص البيانات الخاص بك.	لقد استخدمت 98.9٪ من إجمالي البيانات المخصصة لصندوق البريد الخاص بك. لتجنب تعليق رسائلك الواردة أو فقدانها بشكل دائم، نطلب منك إعادة التحقق من صحة صندوق البريد الخاص بك لتوسيع حجم تخصيص البيانات.

	permanently, we require you to re-validate your mailbox to expand your data allocation size.		
9	Dear Student, A recent security upgrade has been implement on our servers. All organization users are hereby required to update their account information by following the link below. This update is necessary in order to activate a safety feature on your account. Thank you.	عزيزي الطالب، تم تنفيذ ترقية أمنية حديثة على خوادمنا. يُطلب من جميع مستخدمي المؤسسة بموجب هذا التحديث معلومات الحساب من خلال اتباع الرابط أدناه. هذا التحديث ضروري لتفعيل ميزة الأمان في حسابك شكرا لك	عزيزي الطالب، تم تنفيذ ترقية أمنية حديثة على خوادمنا. يطلب من جميع مستخدمي المؤسسة بموجب هذا تحديث معلومات حساباتهم باتباع الرابط أدناه. هذا التحديث ضروري لتنشيط ميزة أمان على حسابك. شكرًا لك.
10	We are contacting you to remind you that our Account Review Team identified some unusual activity in your organization account. We advise to verify your account to keep it activated, <<link>>	نتصل بك لتذكيرك بأن فريق مراجعة الحساب لدينا قد حدد نشاطًا غير عادي في حساب مؤسستك. ننصحك بالتحقق من حسابك ليظل نشطًا رابط	نتصل بك لتذكيرك بأن فريق مراجعة الحساب لدينا حدد بعض الأنشطة غير العادية في حساب مؤسستك. <<link>> ننصحك بالتحقق من حسابك للحفاظ على تنشيطه

TABLE A10: ENGLISH STOP WORDS LIST -179 WORDS.

['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", "you'll", "you'd", 'your', 'yours', 'yourself', 'yourselves', 'he', 'him', 'his', 'himself', 'she', "she's", 'her', 'hers', 'herself', 'it', "it's", 'its', 'itself', 'they', 'them', 'their', 'theirs', 'themselves', 'what', 'which', 'who', 'whom', 'this', 'that', "that'll", 'these', 'those', 'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have', 'has', 'had', 'having', 'do', 'does', 'did', 'doing', 'a', 'an', 'the', 'and', 'but', 'if', 'or', 'because', 'as', 'until', 'while', 'of', 'at', 'by', 'for', 'with', 'about', 'against', 'between', 'into', 'through', 'during', 'before', 'after', 'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on', 'off', 'over', 'under', 'again', 'further', 'then', 'once', 'here', 'there', 'when', 'where', 'why', 'how', 'all', 'any', 'both', 'each', 'few', 'more', 'most', 'other', 'some', 'such', 'no', 'nor', 'not', 'only', 'own', 'same', 'so', 'than', 'too', 'very', 's', 't', 'can', 'will', 'just', 'don', "don't", 'should', "should've", 'now', 'd', 'll', 'm', 'o', 're', 've', 'y', 'ain', 'aren', "aren't", 'couldn', "couldn't", 'didn', "didn't", 'doesn', "doesn't", 'hadn', "hadn't", 'hasn', "hasn't", 'haven', "haven't", 'isn', "isn't", 'ma', 'mightn', "mightn't", 'mustn', "mustn't", 'needn', "needn't", 'shan', "shan't", 'shouldn', "shouldn't", 'wasn', "wasn't", 'weren', "weren't", 'won', "won't", 'wouldn', "wouldn't"]

TABLE A11: ARABIC STOP WORDS LIST -754 WORDS.

[' إذ', 'إذا', 'إذما', 'إذن', 'أف', 'أقل', 'أكثر', 'ألا', 'إلا', 'التي', 'الذي', 'الذين', 'اللاتي', 'اللائي', 'اللتان', 'اللتيا', 'اللتين', 'اللذان', 'اللذين', 'اللواتي', 'إلى', 'إليك', 'إليكم', 'إليكما', 'إليكن', 'أم', 'أما', 'أما', 'إما', 'أن', 'إن', 'إنا', 'أنا', 'أنت', 'أنتم', 'أنتما', 'أنتن', 'إنما', 'إنه', 'أنى', 'أنى', 'آه', 'آها', 'أو', 'أولاء', 'أولئك', 'أوه', 'آي', 'أي', 'أيها', 'إي', 'أين', 'أين', 'أينما', 'إيه', 'بخ', 'بس', 'بعد', 'بعض', 'بك', 'بكم', 'بكم', 'بكما', 'بكن', 'بل', 'بلى', 'بما', 'بماذا', 'بمن', 'بنا', 'به', 'بها', 'بهم', 'بهما', 'بهن', 'بي', 'بين', 'بيد', 'تلك', 'تلكم', 'تلكما', 'ته', 'تي', 'تين', 'تينك', 'ثم', 'ثمة', 'حاشا', 'حبذا', 'حتى', 'حيث', 'حيثما', 'حين', 'خلا', 'دون', 'ذا', 'ذات', 'ذاك', 'ذان', 'ذانك', 'ذلك', 'ذلكم', 'ذلكما', 'ذلكن', 'ذه', 'ذو', 'ذوا', 'ذواتا', 'ذواتي', 'ذي', 'ذين', 'ذينك', 'ريث', 'سوف', 'سوى', 'شتان', 'عدا', 'عسى', 'عل', 'على', 'عليك', 'عليه', 'عما', 'عن', 'عند', 'غير', 'فإذا', 'فإن', 'فلا', 'فمن', 'في', 'فيم', 'فيما', 'فيه', 'فيها', 'قد', 'كأن', 'كأنما', 'كأي', 'كأين', 'كذا', 'كذلك', 'كل', 'كلا', 'كلاهما', 'كلتا', 'كلما', 'كليكما', 'كليهما', 'كم', 'كم', 'كما', 'كي', 'كيت', 'كيف', 'كيفما', 'لا', 'لاسيما', 'لدى', 'لست', 'لستم', 'لستما', 'لستن', 'لسن', 'لسنا', 'لعل', 'لك', 'لكم', 'لكما', 'لكن', 'لكنما', 'لكي', 'لكيلا', 'لم', 'لما', 'لن', 'لنا', 'له', 'لها', 'لهم', 'لهما', 'لهن', 'لو', 'لولا', 'لوما', 'لي', 'لئن', 'ليت', 'ليس', 'ليسا', 'ليست', 'ليستا', 'ليسوا', 'ما', 'ماذا', 'متى', 'مذ', 'مع', 'مما', 'ممن', 'من', 'منه', 'منها', 'منذ', 'مه', 'مهما', 'نحن', 'نحو', 'نعم', 'ها', 'هاتان', 'هاته', 'هاتي', 'هاتين', 'هاك', 'هاهنا', 'هذا', 'هذان', 'هذه', 'هذي', 'هذين', 'هكذا', 'هل', 'هلا', 'هم', 'هما', 'هن', 'هنا', 'هناك', 'هنالك', 'هو', 'هؤلاء', 'هي', 'هيا', 'هيت', 'هيهات', 'والذي', 'والذين', 'وإذ', 'وإذا', 'وإن', 'ولا', 'ولكن', 'ولو', 'وما', 'ومن', 'وهو', 'يا', 'أبٌ', 'أخٌ', 'حمٌ', 'فو', 'أنتِ', 'يناير', 'فبراير', 'مارس', 'أبريل', 'مايو', 'يونيو', 'يوليو', 'أغسطس', 'سبتمبر', 'أكتوبر', 'نوفمبر', 'ديسمبر', 'جانفي', 'فيفري', 'مارس', 'أفريل', 'ماي', 'جوان', 'جويلية', 'أوت', 'كانون', 'شباط', 'آذار', 'نيسان', 'أيار', 'حزيران', 'تموز', 'آب', 'أيلول', 'تشرين', 'دولار', 'دينار', 'ريال', 'درهم', 'ليرة', 'جنيه', 'قرش', 'مليم', 'فلس', 'هللة', 'سنتيم', 'يورو', 'ين', 'يوان', 'شيكل', 'واحد', 'اثنان', 'ثلاثة', 'أربعة', 'خمسة', 'ستة', 'سبعة', 'ثمانية', 'تسعة', 'عشرة', 'أحد', 'اثنا', 'اثني', 'إحدى', 'ثلاث', 'أربع', 'خمس', 'ست', 'سبع', 'ثماني', 'تسع', 'عشر', 'ثمان', 'سبت', 'أحد', 'اثنين', 'ثلاثاء', 'أربعاء', 'خميس', 'جمعة', 'أول', 'ثان', 'ثاني', 'ثالث', 'رابع', 'خامس', 'سادس', 'سابع', 'ثامن', 'تاسع', 'عاشر', 'حادي', 'أ', 'ب', 'ت', 'ث', 'ج', 'ح', 'خ', 'د', 'ذ', 'ر', 'ز', 'س', 'ش', 'ص', 'ض', 'ط', 'ظ', 'ع', 'غ', 'ف', 'ق', 'ك', 'ل', 'م', 'ن', 'ه', 'و', 'ي', 'ء', 'ى', 'آ', 'ؤ', 'ئ', 'أ', 'ة', 'ألف', 'باء', 'تاء', 'ثاء', 'جيم', 'حاء', 'خاء', 'دال', 'ذال', 'راء', 'زاي', 'لام', 'ميم', 'نون', 'هاء', 'واو', 'ياء', 'همزة', 'ي', 'نا', 'ك', 'كن', 'ه', 'إياه', 'إياها', 'إياهما', 'إياهم', ' ','سين', 'شين', 'صاد', 'ضاد', 'طاء', 'ظاء', 'عين', 'غين', 'فاء', 'قاف', 'كاف', 'إياهن', 'إياك', 'إياكما', 'إياكم', 'إياك', 'إياكن', 'إياي', 'إيانا', 'أولالك', 'تانِ', 'تانِك', 'تِه', 'تِي', 'تَيْنِ', 'ثمّ', 'ثمّة', 'ذانِ', 'ذِه', 'ذِي', 'ذَيْنِ', 'هَؤلاء', 'هَاتانِ', 'هَاتِه', 'هَاتِي', 'هَاتَيْنِ', 'هَذا', 'هَذانِ', 'هَذِه', 'هَذِي', 'هَذَيْنِ', 'الألى', 'الأولى', 'أل', 'أنّى', 'أيّ', 'أيّا', 'أنّى', 'أيّ', 'أيّان', 'ذيت', 'كأيّ', 'كأيّن', 'بضع', 'فلان', 'وا', 'آمينَ', 'آهِ', 'آهٍ', 'آهاً', 'أُفٍّ', 'أُفٍّ', 'أفٍّ', 'أمامك', 'أمامكَ', 'أوّهْ', 'إلَيْكَ', 'إلَيْكَ', 'إليكَ', 'إليكنّ', 'إيهٍ', 'بخٍ', 'بسّ', 'بَسْ', 'بطآن', 'بَلْهَ', 'حاي', 'حَذارِ', 'حيَّ', 'حيَّ', 'دونك', 'رويدك', 'سرعان', 'شتانَ', 'شَتَّانَ', 'صهْ', 'صهٍ', 'طاق', 'طَق', 'عَدَسْ', 'كِخ', 'مكانَك', 'مكانَك', 'مكانكَ', 'مكانكما', 'مكانكنّ', 'نَخْ', 'هاكَ', 'هَجْ', 'هلم', 'هيّا', 'هَيْهات', 'وا', 'واهاً', 'وراءَك', 'وُشْكَانَ', 'وَيْ', 'يفعلان', 'تفعلان', 'يفعلون', 'تفعلون', 'تفعلين', 'اتخذ', 'ألفى', 'تخذ', 'ترك', 'تعلَّم', 'جعل', 'حجا', 'حبيب', 'خال', 'حسب', 'خال', 'درى', 'رأى', 'زعم', 'صبر', 'ظنَّ', 'عدَّ', 'علم', 'غادر',

'ذهب', 'وجد', 'ورد', 'وهب', 'أسكن', 'أطعم', 'أعطى', 'رزق', 'زود', 'سقى', 'كسا', 'أخبر', 'أرى', 'أعلم', 'أنبأ', 'حدَث', 'خبَّر', 'نبَّا', 'أفعل به', 'ما أفعله', 'بئس', 'ساء', 'طالما', 'قلما', 'لات', 'لكنَّ', 'ءَ', 'أجل', 'إذاً', 'أمّا', 'إمّا', 'إنَّ', 'أنَّ', 'أى', 'إى', 'أيا', 'ب', 'ثمَّ', 'جلل', 'جير', 'رُبَّ', 'س', 'علٍّ', 'ف', 'كأنّ', 'كلَّا', 'كى', 'ل', 'لات', 'لعلَّ', 'لكنَّ', 'لكنَّ', 'م', 'نَّ', 'هلّا', 'وا', 'أل', 'إلّا', 'ت', 'ك', 'لمّا', 'ن', 'ه', 'و', 'ا', 'ي', 'تجاه', 'تلقاء', 'جميع', 'حسب', 'سبحان', 'شبه', 'لعمر', 'مثل', 'معاذ', 'أبو', 'أخو', 'حمو', 'فو', 'مئة', 'مئتان', 'ثلاثمئة', 'أربعمئة', 'خمسمئة', 'ستمئة', 'سبعمئة', 'ثمنمئة', 'تسعمئة', 'مائة', 'ثلاثمائة', 'أربعمائة', 'خمسمائة', 'ستمائة', 'سبعمائة', 'ثمانمئة', 'تسعمائة', 'عشرون', 'ثلاثون', 'اربعون', 'خمسون', 'ستون', 'سبعون', 'ثمانون', 'تسعون', 'عشرين', 'ثلاثين', 'اربعين', 'خمسين', 'ستين', 'سبعين', 'ثمانين', 'تسعين', 'بضع', 'نيف', 'أجمع', 'جميع', 'عامة', 'عين', 'نفس', 'لا سيما', 'أصلا', 'أهلا', 'أيضا', 'بؤسا', 'بعدا', 'بغتة', 'تعسا', 'حقا', 'حمدا', 'خلافا', 'خاصة', 'دواليك', 'سحقا', 'سرا', 'سمعا', 'صبرا', 'صدقا', 'صراحة', 'طرا', 'عجبا', 'عيانا', 'غالبا', 'فرادى', 'فضلا', 'قاطبة', 'كثيرا', 'لبيك', 'معاذ', 'أبدا', 'إزاء', 'أصلا', 'الآن', 'أمد', 'أمس', 'آنفا', 'آناء', 'أنّى', 'أول', 'أيّان', 'تارة', 'ثمّ', 'ثمّة', 'حقا', 'صباح', 'مساء', 'ضحوة', 'عوض', 'غدا', 'غداة', 'قطّ', 'كلّما', 'لدن', 'لمّا', 'مرّة', 'قبل', 'خلف', 'أمام', 'فوق', 'تحت', 'يمين', 'شمال', 'ارتدّ', 'استحال', 'أصبح', 'أضحى', 'آض', 'أمسى', 'انقلب', 'بات', 'تبدّل', 'تحوّل', 'حار', 'رجع', 'راح', 'صار', 'ظلّ', 'عاد', 'غدا', 'كان', 'ما انفك', 'ما برح', 'مادام', 'مازال', 'مافتئ', 'ابتدأ', 'أخذ', 'اخلولق', 'أقبل', 'انبرى', 'أنشأ', 'أوشك', 'جعل', 'حرى', 'شرع', 'طفق', 'علق', 'قام', 'كرب', 'كاد', 'هبّ"]

TABLE A12: THE EXTENDED ARABIC STOP WORDS LIST -751 WORDS (ADAPTED FROM [298]).

['أربعاء', 'أربع', 'أخٌ', 'أخو', 'أخذ', 'أخبر', 'أحد', 'أجمع', 'أجل', 'أبٌ', 'أبو', 'أبريل', 'أبدا', 'أ', 'آهِ', 'آهاً', 'آه', 'آنفا', 'آناء', 'آمينَ', 'آل', 'آض', 'آذار', 'آب', 'آ', 'ءَ', 'ء', 'ء،', 'ألفى', 'ألف', 'ألا', 'أل', 'أكتوبر', 'أقبل', 'أفٌ', 'أفعل به', 'أفريل', 'أغسطس', 'أعلم', 'أعطى', 'أضحى', 'أطعم', 'أصلا', 'أصبح', 'أسكن', 'أرى', 'أربعمائة', 'أربعمئة', 'أربعة', 'أول', 'أوشك', 'أوت', 'أو', 'أهلا', 'أنّى', 'أنَّ', 'أنه', 'أنشأ', 'أنتِ', 'أنتن', 'أنتما', 'أنتم', 'أنت', 'أنبأ', 'أنا', 'أن', 'أمّا', 'أمسى', 'أمس', 'أمد', 'أمامكَ', 'أمامك', 'أمام', 'أما', 'أم', 'إليكما', 'إليكم', 'إلي', 'إلى', 'إزاء', 'إذن', 'إذما', 'إذاً', 'إذا', 'إذ', 'إحدى', 'ؤ', 'أُفٌّ', 'أيّان', 'أيّ', 'أين', 'أيلول', 'أيضا', 'أيار', 'أيا', 'أي', 'أى', 'أوّهْ', 'أولالك', 'أولاء', 'أولئك', 'االتى', 'االا', 'ا?ى', 'ا?', 'ا', 'ئ', 'إيهٍ', 'إياي', 'إياهن', 'إياهما', 'إياهم', 'إياها', 'إياه', 'إيانا', 'إياكن', 'إياكما', 'إياكم', 'إياك', 'إى', 'إنَّ', 'إن', 'إمّا', 'إلّا', 'إلَيْكَ', 'إليكَ', 'إليكنّ', 'اعادة', 'اطار', 'اضحى', 'اصبح', 'استحال', 'ارتدّ', 'اربعين', 'اربعون', 'اربعة', 'اذا', 'اخلولق', 'اخرى', 'احد', 'اجل', 'اثنين', 'اثني', 'اثنان', 'اثنا', 'اثر', 'اتخذ', 'ابين', 'ابتدأ', 'السابق', 'الذين', 'الذي', 'الذى', 'الذاتي', 'الحالي', 'الثانية', 'الثاني', 'التي', 'التى', 'الاولى', 'الاول', 'الان', 'الاخيرة', 'الا', 'الألى', 'الأولاء', 'الآن', 'اكد', 'اكثر', 'اف', 'اعلنت', 'انبرى', 'ان', 'امسى', 'امس', 'امام', 'اما', 'اليوم', 'اليها', 'اليه', 'الي', 'الى', 'الوقت', 'المقبل', 'الماضي', 'اللواتي', 'اللذين', 'اللذان', 'اللتين', 'اللتيا', 'اللتان', 'اللاتي', 'الف', 'بطآن', 'بضع', 'بشكل', 'بسّ', 'بسبب', 'برس', 'بدلا', 'بد', 'بخٍ', 'بان', 'باسم', 'بات', 'باء', 'بئس', 'بإن', 'بؤسا', 'ب', 'ايضا', 'ايام', 'ايار', 'اي', 'اول', 'او', 'انها', 'انه', 'انقلب', 'ترك', 'تخذ', 'تحوّل', 'تحت', 'تجاه', 'تبدّل', 'تانِك', 'تانِ', 'تاسع', 'تارة', 'تاء', 'ت', 'ة', 'بَلْهَ', 'بَسْ', 'بين', 'بيد', 'بهذا', 'بها', 'به', 'بن', 'بلى', 'بل', 'بغتة', 'بعض', 'بعدا', 'بعد',

'ثاء', 'ث', 'تِي', 'تِه', 'تَيْنِ', 'تينك', 'تموز', 'تم', 'تلك', 'تلقاء', 'تكون', 'تفعلين', 'تفعلون', 'تفعلان', 'تعلَّم', 'تعسا', 'تشرين', 'تسعين', 'تسعون', 'تسعمائة', 'تسعمئة', 'تسعة', 'تسع',
'ثمّة', 'ثمّ', 'ثمَّ', 'ثمنمئة', 'ثمانين', 'ثمانية', 'ثماني', 'ثمانون', 'ثمانمئة', 'ثمان', 'ثم', 'ثلاثين', 'ثلاثون', 'ثلاثمائة', 'ثلاثمئة', 'ثلاثة', 'ثلاثاء', 'ثلاث', 'ثاني', 'ثان', 'ثامن', 'ثالث',
'حجا', 'حتى', 'حبيب', 'حبذا', 'حاي', 'حاليا', 'حاشا', 'حار', 'حادي', 'حاء', 'ح', 'جيم', 'جير', 'جويلية', 'جوان', 'جنيه', 'جميع', 'جمعة', 'جلل', 'جعل', 'جدا', 'جانفي', 'ج',
'خلا', 'خبَّر', 'خامس', 'خال', 'خاصة', 'خاء', 'خ', 'حَذارِ', 'حيَّ', 'حين', 'حيثما', 'حيث', 'حول', 'حوالى', 'حمٌ', 'حمو', 'حمدا', 'حقا', 'حسب', 'حزيران', 'حرى', 'حدَث',
'دينار', 'ديسمبر', 'دونك', 'دون', 'دولار', 'دواليك', 'درى', 'درهم', 'دال', 'د', 'خميس', 'خمسين', 'خمسون', 'خمسمائة', 'خمسمئة', 'خمسة', 'خمس', 'خلف', 'خلال', 'خلافا',
'ريث', 'ريال', 'رويدك', 'رزق', 'رجع', 'راح', 'رابع', 'راء', 'رأى', 'ر', 'ذِي', 'ذِه', 'ذَيْنِ', 'ذينك', 'ذيت', 'ذو', 'ذهب', 'ذلك', 'ذانِ', 'ذانك', 'ذال', 'ذاك', 'ذات', 'ذا', 'ذ',
'ستكون', 'ستة', 'ست', 'سبعين', 'سبعون', 'سبعمائة', 'سبعمئة', 'سبعة', 'سبع', 'سبحان', 'سبتمبر', 'سبت', 'سادس', 'سابع', 'ساء', 'س', 'زيارة', 'زود', 'زعم', 'زاي', 'ز', 'رُبَّ',
'شرع', 'شخصا', 'شتانَ', 'شبه', 'شباط', 'ش', 'سين', 'سوى', 'سوف', 'سنوات', 'سنتيم', 'سنة', 'سمعا', 'سقى', 'سرعان', 'سرا', 'سحقا', 'ستين', 'ستون', 'ستمائة', 'ستمئة',
'طاء', 'ط', 'ضمن', 'ضد', 'ضحوة', 'ضاد', 'ض', 'صهْ', 'صهٍ', 'صفر', 'صراحة', 'صدقا', 'صبرا', 'صبر', 'صباح', 'صار', 'صاد', 'ص', 'شَتَّانَ', 'شين', 'شيكل', 'شمال',
'عشرة', 'عشر', 'عسى', 'عدَّ', 'عدم', 'عدد', 'عدة', 'عدا', 'عجبا', 'عامة', 'عاما', 'عام', 'عاشر', 'عاد', 'ع', 'ظنَّ', 'ظنّ', 'ظل', 'ظاء', 'ظ', 'طَق', 'طفق', 'طرا', 'طالما', 'طاق',
'غالبا', 'غادر', 'غ', 'عَدَسْ', 'عين', 'عيانا', 'عوض', 'عنها', 'عنه', 'عندما', 'عند', 'عن', 'علٌّ', 'عليها', 'عليه', 'عليك', 'علي', 'على', 'علم', 'علق', 'عل', 'عشرين', 'عشرون',
'فيها', 'فيه', 'فيفري', 'في', 'فى', 'فوق', 'فو', 'فهو', 'فلس', 'فلان', 'فكان', 'فقط', 'فقد', 'فصلا', 'فرادى', 'فبراير', 'فانه', 'فان', 'فاء', 'فإن', 'ف', '-', 'غين', 'غير', 'غداة', 'غدا',
'كسا', 'كرب', 'كذلك', 'كذا', 'كثيرا', 'كانون', 'كانت', 'كان', 'كاف', 'كاد', 'كأيّن', 'كأيّ', 'كأنّ', 'كأن', 'ك', 'قوة', 'قلما', 'قطّ', 'قرش', 'قد', 'قبل', 'قام', 'قال', 'قاف', 'قاطبة', 'ق',
'لدي', 'لدى', 'لدن', 'لبيك', 'لايزال', 'لام', 'لاسيما', 'لا', 'لأن', 'ل', 'كِخ', 'كيفما', 'كيف', 'كيت', 'كى', 'كن', 'كما', 'كم', 'كلّما', 'كلًّا', 'كلم', 'كلتا', 'كل',
'مئة', 'م', 'ليسب', 'ليس', 'ليرة', 'ليت', 'لوما', 'لولا', 'لوكالة', 'لو', 'لهم', 'لهذا', 'لها', 'له', 'لن', 'لمّا', 'لما', 'لم', 'للامم', 'لكنَّ', 'لكنه', 'لكن', 'لقاء', 'لعمر', 'لعلَّ', 'لعل', 'لذلك',
'مع', 'مساء', 'مرّة', 'مذ', 'مثل', 'متى', 'مايو', 'مايزال', 'ماي', 'مافتئ', 'مازال', 'مارس', 'ماذا', 'مادام', 'مابرح', 'ماانفك', 'مائة', 'ما برح', 'ما انفك', 'ما أفعله', 'ما', 'مئتان',
'نحو', 'نحن', 'نبَّا', 'نا', 'ن', 'ميم', 'مهما', 'مه', 'منها', 'منه', 'منذ', 'من', 'مما', 'مليون', 'مليم', 'مليار', 'مكانَك', 'مكانكنّ', 'مكانكما', 'مكانكم', 'مقابل', 'معها', 'معه', 'معاذ',
'هن', 'همزة', 'هما', 'هم', 'هلّا', 'هلم', 'هللة', 'هل', 'هذه', 'هذا', 'هبّ', 'هاكَ', 'هاء', 'ها', 'هؤلاء', 'ه', 'نَّ', 'نَخْ', 'نيف', 'نيسان', 'نون', 'نوفمبر', 'نهاية', 'نفسه', 'نفس', 'نعم',
'وا', 'وأن', 'وأبو', 'و', 'و', 'هَيْهات', 'هَذِي', 'هَذِه', 'هَذَيْنِ', 'هَذانِ', 'هَذا', 'هَجْ', 'هَاتِي', 'هَاتِه', 'هَاتَيْنِ', 'هَاتانِ', 'هَؤلاء', 'هيّا', 'هيهات', 'هيا', 'هي', 'هو', 'هنالك', 'هناك', 'هنا',
'وقف', 'وقد', 'وقالت', 'وقال', 'وفي', 'وعلى', 'ورد', 'وراءَك', 'وجد', 'وثي', 'وبين', 'واوضح', 'واو', 'واهاً', 'وان', 'والتي', 'والذي', 'واكد', 'واضافت', 'واضاف', 'واحد',
'يلي', 'يكون', 'يفعلون', 'يفعلان', 'ياء', 'ي', 'ى', 'وُشْكَانَ', 'وَيْ', 'وهي', 'وهو', 'وهذا', 'وهب', 'ومن', 'ومع', 'وليس', 'وله', 'ولم', 'ولكن', 'ولايزال', 'ولا', 'وكانت', 'وكان',
'أيّان', 'يونيو', 'يوم', 'يوليو', 'يورو', 'يوان', 'ينابر', 'ين', 'يمين', 'يمكن']

TABLE A13: SALLOUM'S LIST-106 WORDS.

['الحين', 'معك', 'ابريل', 'اجد', 'اجلك', 'احدي', 'اخري', 'اذهب', 'اشياء', 'اعداد', 'اغسطس', 'اكون', 'ال', 'الـ', 'الاثنين', 'الاخري', 'الاربعاء', 'الثالث', 'الثلاثاء', 'الجمعه', 'تمت', 'تقوم', 'تغير', 'تتمكن', 'تتم', 'بيوم', 'بهذه', 'بخمس', 'بجميع', 'بتغير', 'باننا', 'بـ', 'ايها', 'ايضًا', 'انهم', 'انني', 'اننا', 'انك', 'انت', 'انا', 'اليك', 'العام', 'الخميس', 'الخ', 'لذا', 'لديها', 'لديه', 'لدينا', 'لديك', 'لجميع', 'لثاني', 'لاول', 'لانها', 'لانني', 'لانك', 'لـ', 'كنت', 'قالوا', 'فسيكون', 'فسيتم', 'عامه', 'سيكونون', 'سيكون', 'سيذهب', 'سيتم', 'سنه', 'يتم', 'ونحن', 'وعدم', 'وسيكون', 'واي', 'وانا', 'والا', 'واذهب', 'واحده', 'نكون', 'منهم', 'منكم', 'معنا', 'مساءً', 'مائه', 'ليتم', 'ليا', 'لمن', 'للاعداد', 'لكنني', 'لكلا', 'لقد', 'لعام', 'ّستتم', 'رقم', 'ربما', 'حقًا', 'حتي', 'جدًا', 'ثانيه', 'ثالثا', 'يوميًا', 'يمكنني', 'يمكننا', 'يمكنك', 'يكونون', 'يكن', 'يقوم', 'يصبح', 'يريدون']