

أريج العتيبي

Automatic Detection of Arabic Spam Messages Using Machine Learning Techniques...docx

 Midocean University

Document Details

Submission ID

trn:oid::2945:405395056

Submission Date

Aug 7, 2026, 5:25 AM GMT+3

Download Date

Aug 7, 2026, 10:42 PM GMT+3

File Name

Automatic Detection of Arabic Spam Messages Using Machine Learning Techniques...docx

File Size

3.9 MB

70 Pages

11,948 Words

71,104 Characters

21% detected as AI

The percentage indicates the combined amount of likely AI-generated text as well as likely AI-generated text that was also likely revised by AI.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.



34 AI-generated 21%

Likely generated or likely generated and revised by AI.

Disclaimer

Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (it may misidentify writing that is likely human generated as AI generated and likely AI generated as human generated) so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.



Automatic Detection of Arabic Spam Messages Using Machine Learning Techniques

Prepared By

Name: Asma Sayer Alotaibi

ID No:253001753

Name: Areej Abdualaziz Alotaibi

ID No:253001816

Submitted in fulfillment of the requirements for the degree Master of
Science in Artificial Intelligence

At

Collage of Informatics

Midocean University

UAE

Supervisor Name: Dr. Khaled Eskaf

Collage of Informatics, Midocean University, Comoros

Email: khaledeskaf@midocean.edu.km

Acknowledgements

Special thanks to Dr. Khaled Eskaf

List of Contents

ii

Acknowledgements iii

List of Abbreviations vi

List of Figures..... vii

List of Tables viii

Chapter 1 Introduction 1

1.1 Overview 1

1.2 Problem Definition.....2

1.3 Thesis Objectives2

1.4 Thesis Contribution.....3

1.5 Thesis Organization3

Chapter 2 Background and Related Work.....5

2.1 Introduction5

2.2 Background5

2.3 Related works.....10

CHAPTER THREE PROPOSED APPROACH (METHODOLOGY).....27

3.1 Introduction27

3.2 Research Methodology27

3.3 Dataset Description28

3.4 Development Environment30

3.5 Arabic Text Preprocessing.....31

3.6 Feature Extraction Using TF-IDF33

3.7 Dataset Partitioning.....34

3.8 Machine Learning Classification Models34

3.9 Model Training.....36

3.10 Performance Evaluation.....36

3.11 Confusion Matrix Analysis38

3.12 Experimental Workflow.....	38
3.13 Chapter Summary.....	39
CHAPTER FOUR EXPERIMENTAL RESULTS AND DISCUSSION	40
4.1 Introduction	40
4.2 Experimental Setup	40
4.3 Dataset Distribution	41
4.4 Performance Evaluation Metrics.....	42
4.5 Experimental Results	42
4.6 Discussion of Classification Results.....	43
4.7 Confusion Matrix Analysis	45
4.8 Comparative Analysis	47
4.9 Comparison with Previous Studies	48
4.10 Summary of Experimental Findings	51
4.11 Chapter Summary.....	51
CHAPTER FIVE CONCLUSION AND FUTURE WORK	53
5.1 Introduction	53
5.2 Research Summary.....	53
5.3 Research Findings	54
5.4 Research Contributions	55
5.5 Practical Implications.....	56
5.6 Research Limitations.....	57
5.7 Future Work	57
5.8 Chapter Summary.....	58
References	59

List of Abbreviations

- **SMS:** Short Message Service
- **NLP:** Natural Language Processing
- **ML:** Machine Learning
- **DL:** Deep Learning
- **SVM:** Support Vector Machine
- **NB:** Naïve Bayes
- **RF:** Random Forest
- **CNN:** Convolutional Neural Network
- **RNN:** Recurrent Neural Network
- **LSTM:** Long Short-Term Memory
- **BiLSTM:** Bidirectional Long Short-Term Memory
- **GRU:** Gated Recurrent Unit
- **TF-IDF:** Term Frequency – Inverse Document Frequency
- **BoW:** Bag of Words
- **GAN:** Generative Adversarial Network
- **IoT:** Internet of Things
- **URL:** Uniform Resource Locator
- **API:** Application Programming Interface
- **AI:** Artificial Intelligence
- **ELM:** Extreme Learning Machine
- **F1-score:** Harmonic Mean of Precision and Recall

List of Figures

Figure 1. NLP text preprocessing pipeline diagram	7
Figure 2. spam extraction methods NLP diagram	8
Figure 3. CNN for text classification architecture diagram.....	9
Figure 4 Proposed Framework.....	28
Figure 5 Confusion Matrix.....	46
Figure 6 comparative performance of the three machine learning algorithms.....	48

List of Tables

Table 1 Related Work Summary.....	17
Table 2 Dataset Distribution	29
Table 3 Development Environment	31
Table 4 Dataset Distribution	41
Table 5 Performance Comparison of Machine Learning Models.....	42
Table 6 Confusion Matrix of the SVM Classifier.....	46
Table 7 Comparison with Previous Studies	49

Abstract

The automatic detection of Arabic phishing emails is an important research challenge since phishing attacks are becoming a serious threat in cyber security. The morphology, spelling variations, dialectal variations and scarcity of publicly annotated datasets of Arabic makes phishing email classification an especially challenging task. This work introduces a lightweight machine learning model for the recognition of Arabic phishing emails that employs a wide Arabic text pre-processing and feature extraction techniques called Term Frequency-Inverse Document Frequency (TF-IDF). The experiments have been performed on the publicly available Arabic Phishing Emails Dataset (Kaggle) consisting of 1,250 Arabic emails (1000 legitimate and 250 phishing). A stratified train-test split was used to split the data into a training set of 80% and a testing set of 20%. Three supervised machine learning classifiers (Naïve Bayes (NB), Support Vector Machine (SVM), and Random Forest (RF)) were trained and evaluated with the same conditions of preprocessing and experiments. The results indicated that SVM and RF gave the highest overall performance, each with the accuracy of 98.0%, the precision of 100.0%, the recall of 90.0% and the F1-score of 94.74% on the test set. The results show that the traditional machine learning techniques, coupled with efficient Arabic text preprocessing and the TFIDF feature extraction can be an efficient and accurate phishing email detection system for Arabic. The results are, however, limited due to the relatively small number of datasets in Arabic phishing email, and future work could include larger and more diverse Arabic phishing email datasets to validate the results.

Keywords : Arabic Phishing Detection; Machine Learning; Natural Language Processing; TF-IDF; Support Vector Machine; Random Forest.

Chapter 1 Introduction

1.1 Overview

Digital communication has expanded rapidly and has become a major channel for personal, commercial, and financial interactions. This fast growth has also resulted in a corresponding rise in the number of phishing attacks, which involve illegitimate messages containing malicious links, or fraudulent requests to get details from users. Phishing is one of the most serious threats plaguing cyber security, capable of causing financial damage, identity theft and access to personal or organizational accounts.

It is especially difficult to recognize Arabic-based phishing messages because of the morphology richness and complexity of the Arabic language, its wide use of different dialects, writing styles and informal expressions. It's harder to do automatic text analysis than in many other languages because of these attributes. Moreover, quality annotated Arabic phishing data sets are not readily available, and the preprocessing of Arabic text remains a challenge, especially in the context of developing a well-performing phishing detection system [1-3].

Previous studies have investigated various machine learning and deep learning techniques for detecting malicious Arabic text. The differentiation of malicious and legitimate Arabic messages using traditional machine learning algorithms has been found promising, e.g., Decision Tree and Naïve Bayes [1]. Researchers have used more recent models, such as supervised machine learning and deep learning, and found that the preprocessing, tokenization and feature extraction of these models have a significant effect on the improvement of classification accuracy [2]. To overcome the lack of labeled Arabic phishing data, the use of data augmentation techniques has been proposed by other researchers which yielded better recall and F1 score [3]. Furthermore, recent research on phishing attacks on Arabic emails and SMS messages showed that deep learning models such as BiGRU can successfully extract context from short Arabic texts and boost phishing detection accuracy [4] , [5], respectively, using hybrid deep learning architectures.

The current thesis is based on the assumption that the Arabic phishing messages can be detected more accurately by the fusion of effective Arabic Text Preprocessing techniques with appropriate

machine learning classifiers. Phishing messages are crafted with a strategy of deceiving the recipient through the use of persuasive messages, malicious URLs and social engineering. This necessitates that an efficient phishing detector be able to detect both lexical and contextual characteristics in Arabic text. Thus, the present study suggests a machine learning-based framework for detecting Arabic phishing messages by performing extensive text preprocessing and TF-IDF feature extraction. The final goal is to provide a solution for detecting Arabic phishing emails in real world cyber security application that is accurate, efficient and practical.

1.2 Problem Definition

While phishing detection in Arabic language has developed reasonably well, the binary classification of Arabic phishing emails and legitimate Arabic messages has not been sufficiently explored as compared to English-language systems. The linguistic complexity of Arabic email classification is compounded by the presence of rich morphology, dialectal variation, spelling variation, non-standard writing and code switching. Further, machine learning models are not effective and generalizable because of the availability of limited datasets of Arabic phishing emails and class imbalance. The main focus of this study is to consider the classification of Arabic phishing and legitimate emails in a binary classification using the publicly available Arabic Phishing Emails Dataset from Kaggle. The proposed framework includes Arabic text pre-processing, TF-IDF feature extraction, and supervised machine learning classifiers to create an efficient, accurate and practical solution for detecting Arabic phishing emails.

1.3 Thesis Objectives

The main goal of this thesis is to create an effective framework of machine learning techniques for detecting Arabic phishing emails from textual content. The study starts by preparing the Arabic Phishing Emails Dataset and implementing a comprehensive Arabic text preprocessing pipeline to clean and normalize the text, remove repeated characters, eliminate stop words, tokenize the text, and extract the TF-IDF features from the dataset.

This study features the training and comparison of three supervised machine learning classifiers:

- Support Vector Machine (SVM)

- Naïve Bayes (NB)
- Random Forest (RF)

The evaluation of these classifiers is done by using accuracy, precision, recall, f1-score and confusion matrix analysis to select the best model for detecting Arabic phishing emails. The proposed work is an accurate, efficient and feasible approach to detect phishing emails in an environment such as Arabic language.

1.4 Thesis Contribution

This research is adding to the world of Arabic Natural language processing and Cybersecurity in the following ways:

- Developing a comprehensive Arabic phishing email preprocessing pipeline, including text cleaning, normalization, repeated character removal, stop-word removal, and tokenization .
- Employing TF-IDF as an effective feature representation technique for transforming Arabic phishing email text into numerical feature vectors suitable for machine learning classification .
- Training and comparing three supervised machine learning classifiers—Naïve Bayes (NB), Support Vector Machine (SVM), and Random Forest (RF)—using the same Arabic Phishing Emails Dataset and identical preprocessing pipeline to ensure a fair comparison .
- Providing a detailed analysis of false-negative predictions and discussing their practical implications for cybersecurity, demonstrating the effectiveness and computational efficiency of the proposed framework for Arabic phishing email detection .

1.5 Thesis Organization

Chapter 2: The chapter provides a review of related literature on spam and phishing detection, specifically, Arabic text classification techniques and the current challenges.

Chapter 3: In this chapter, the methodology, such as the preparation of the dataset, the preparation of preprocessing, extracting features through TF-IDF and word embeddings, and the execution of SVM, Naive Bayes, CNN, and BiLSTM models, are described.

Chapter 4: This chapter explains the experimental results and measures the model performance based on Accuracy, Precision, Recall, and F1-score and reflects on the results, compares the performance of the various models and discusses their strengths and weaknesses.

Chapter 5: This chapter will end the thesis with a summary of the main findings and propose the direction of the research in the future.

Chapter 2 Background and Related Work

2.1 Introduction

This chapter outlines theoretical background and past research that will back the proposed work. It presents the main concepts of spam and phishing detection, the fact that Arabic text is hard to process automatically and it discusses the primary methods of text processing, feature extraction and classification. This chapter aims at demonstrating the technical background of the research and situating the suggested methodology within the current body of literature on Arabic spam and phishing detection. Studies in this field demonstrate that the classification of Arabic texts has gained more and more relevance as messages with malicious intent are expressed in numerous forms (such as web spam, social spam, SMS spam, phishing) and the Arabic language presents new processing difficulties due to morphology, dialects, and noisy writing styles [6], [7], [8], [13].

2.2 Background

2.2.1 Spam and Phishing Concepts

Spam can be defined as unsolicited or unwanted digital content that is delivered in large volumes with the purpose of advertising, manipulation or malicious use. Spam can be presented in the form of web pages, tweets, comments, emails, or SMS messages, and can be used in text classification tasks to deceive users, propagate malicious links or create artificial popularity. Phishing is a more specific threat where attackers can pretend to be trusted organizations to dupe the user to share sensitive information like passwords, banking information, or verification codes. The two issues are closely related to cybersecurity since they use human trust and natural language as opposed to technical vulnerabilities. Past research indicates that Arabic spam is posted in web pages, social media, and SMS messages, with phishing being developed with the help of text-based spamming and deceptive URLs [6], [7], [13].

2.2.2 Arabic Text Characteristics

Specific difficulties of automatic classification exist in Arabic text. It is morphologically abundant, highly inflected and tends to be written in both Modern Standard Arabic and local dialects. Messages on the web often lack diacritics, stress with repetition, combine Arabic and English words or numbers and employ informal spellings or abbreviations. These characteristics generate noise and enhance ambiguity, rendering straightforward methods based on keywords less efficient. The studies on Arabic SMS phishing in particular remark that the rich morphology and the intricate structure of the language may decrease the effectiveness of the text classification standard procedures unless the text is properly normalized and modeled using the relevant techniques [13]. Arabic spam investigations also reveal that brief, noisy and informal Arabic posts need a quality preprocessing prior to effective classification [7], [8], [13].

2.2.3 Text Preprocessing Techniques

Text preprocessing is a crucial step in any text classification pipeline, as it involves cleaning and normalizing the raw text data into a format that can be used by machine learning algorithms. Some of the most common pre-processing steps are removing punctuation, normalization of the text, removal of stop words, normalization of repeated characters, tokenization, and elimination of noise like URLs and emojis. In the past, many applications have been shown to significantly benefit from the use of effective preprocessing techniques, even in the face of old fashioned machine learning algorithms [9]. The preprocessing pipeline used in the experiments of this study includes text cleaning, Arabic text normalization, removal of repeated characters, removal of stop words and tokenization. Before feature extraction, these steps were done to the Arabic Phishing Emails Dataset. Other preprocessing methods cited in the literature are only mentioned to serve as background information and were not implemented in the experiments. Preprocessing is also crucial for Arabic phishing email detection since other variations in spelling, dialectal words and noisy text may jeopardize the classification accuracy. In this sense, preprocessing was viewed as an indispensable part of the proposed detection framework instead of as an auxiliary process [7, 8, 9, 13].

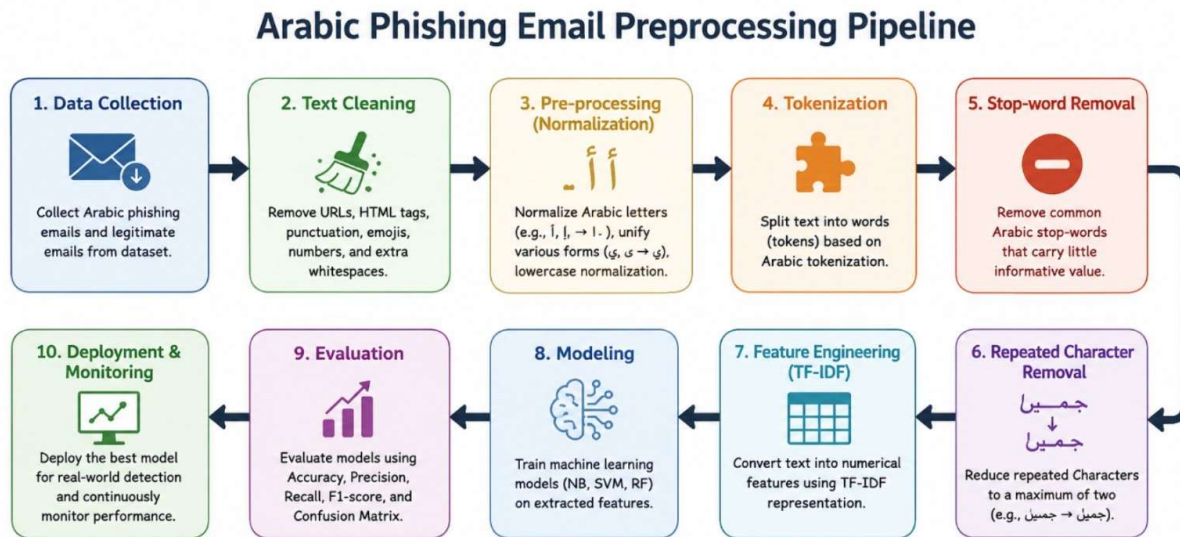


Figure 1. NLP text preprocessing pipeline diagram

2.2.4 Feature Extraction Methods

After pre-processing the text, it is then transformed into numerical feature vectors that can be used by machine learning algorithms. In this study, the authors use Term Frequency–Inverse Document Frequency (TF-IDF) to convert the Arabic phishing emails into numeric vectors. TF-IDF is an effective and efficient feature representation in text classification for small text documents in a large collection. Embedding-based representations like Word2Vec, FastText and BERT (BERT [10] is a state-of-the-art model) have shown to be more effective in learning semantic and contextual relationships in complex language tasks, but were not tested in the present work. They are only addressed theoretically in the context of the discussions. Due to its simplicity, efficiency, and effectiveness in the traditional machine learning-based phishing email detection, the experimental framework is based solely on TF-IDF. In Arabic phishing email detection, the importance of feature representation is highlighted, due to the influence of the meaning of short

messages on morphological variations and contextual expressions. Hence, the TF-IDF method was chosen for feature extraction for the entire experimental evaluation [10] , [13].

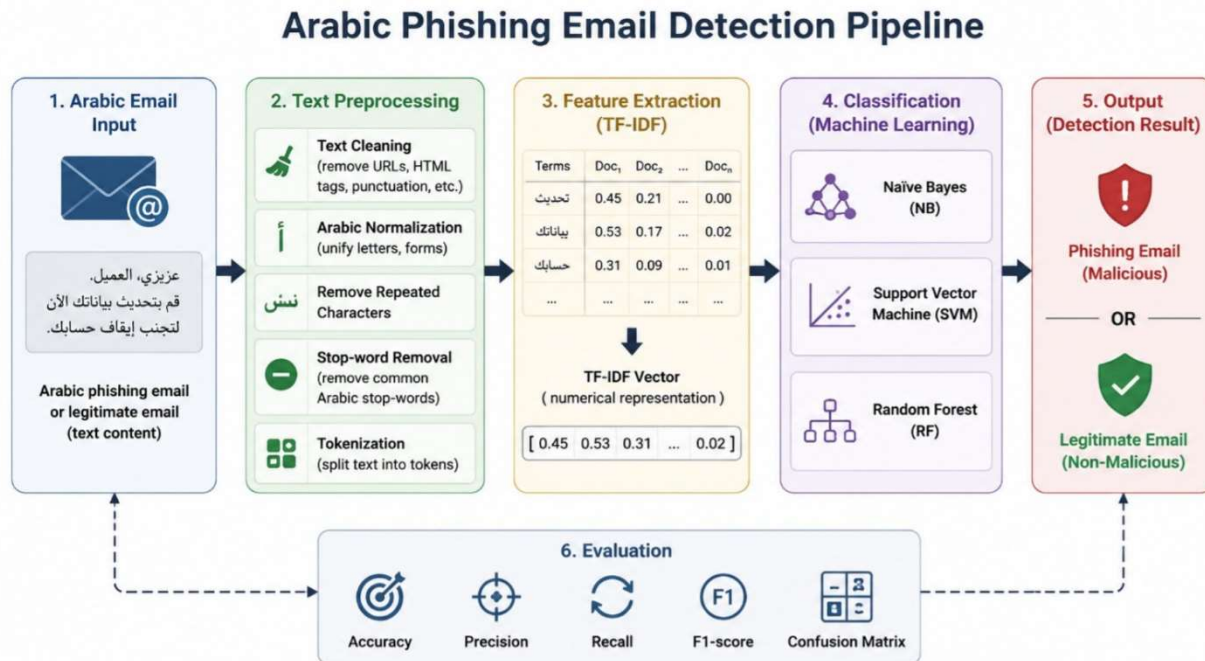


Figure 2. spam extraction methods NLP diagram

2.2.5 Machine Learning Algorithms

Conventional machine learning models are still very applicable in the text classification process with an excellent preprocessing and feature engineering. Naive Bayes is a common algorithm due to its simplicity, speed and ability to work with high dimensional sparse data like TF-IDF vectors. The other powerful classifier is Support Vector Machine (SVM) since it is capable of complex boundaries and tends to work well with text data. Comparative experiments on spam classification reveal that, both of these methods are efficient, although SVM frequently obtains better results in cases when the feature space is well-constructed [10]. Classical algorithms continue to offer a useful basis in Arabic spam detection and can be competitive, particularly in cases where the dataset is small, or the features are well-constructed [7], [8], [10].

2.2.6 Deep Learning Models

The advantage of deep learning over traditional methods is that it can automatically extract complex semantic and context patterns from textual data, which makes it one of the best methods for text classification. CNN, BiLSTM and BiGRU models have proven to be powerful in the field of NLP due to their ability to model long-range context dependency and local linguistic properties. Previously, it was demonstrated that these models can yield very high phishing and spam detection classification accuracy when adequate labeled data and computational resources are available, in the context of Arabic phishing and spam detection [11, 12]. Though these benefits, deep learning models generally need a large of annotated dataset, longer training time and high computation resources. For this reason, the current study focused on shallow traditional machine learning models rather than deep learning models. With the relatively small dataset of Arabic Phishing Emails (1250 emails), the traditional classifiers with effective Arabic text pre-processing and TF-IDF feature extraction are computationally efficient, interpretable and practical classifiers with competitive classification performance. In this section only deep learning models are mentioned as background, and not implemented for the experimental evaluation.

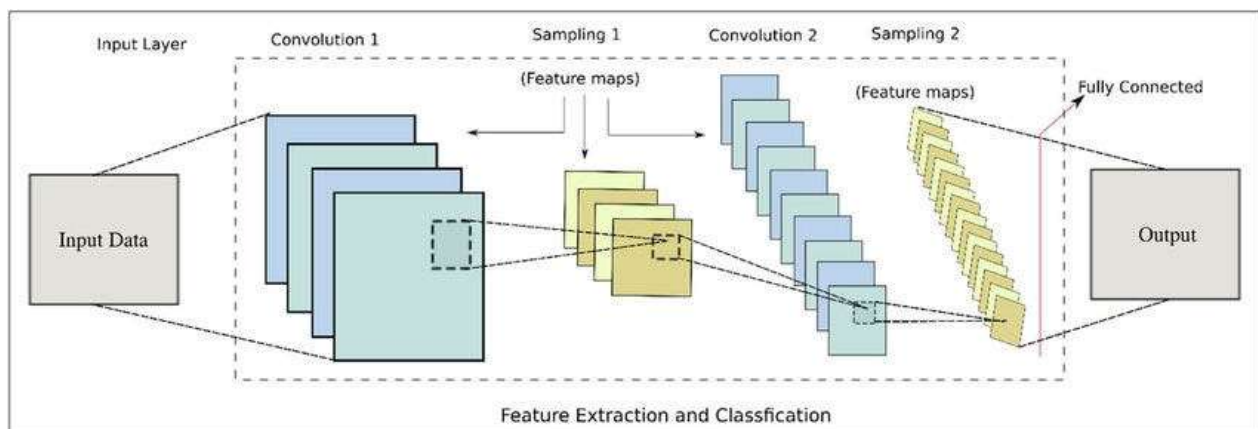


Figure 3. CNN for text classification architecture diagram

2.3 Related works

[14] Al Saidat et al. (2024)

Al Saidat et al. (2024) examined the problem of Arabic SMS spam detection with the emphasis on linguistic complexity of Arabic text, including rich morphology and contextual structure. The research suggested a hybrid deep learning network, which integrated Convolutional Neural Networks (CNN) and Bidirectional Long Short-Term Memory (Bi-LSTM), to learn both local features and long-term relationships within text messages. The data set included Arabic SMS messages that were classified as spam and legitimate. Some of the preprocessing steps used by the authors include text cleaning, tokenization, and padding. The model performed well with an accuracy of 96.99, precision of 0.9739, recall of 0.9675 and F1-score of 0.9707.

[15] Alzubaidi and Ghanem (2025)

Alzubaidi and Ghanem (2025) studied the Arabic SMS spam detection using the prism of social engineering and phishing schemes prevalent in fraudulent text messages. They employed a real-life Saudi Arabian dataset comprising of spam keywords, suspicious URLs and typical phone-number patterns that are related to spam behavior. A number of machine learning classifiers were experimented with such as Random Forest, Naive Bayes, and Support Vector Machines (SVM). Another hybrid model suggested by the study is a Random Forest-SVM to enhance performance. Preprocessing and feature engineering were significant factors in the ultimate accuracy. The hybrid model achieved an accuracy of 94, a precision of 0.94, recall of 0.93 and an F1-score of 0.94.

[16] Alkaabi et al. (2025)

Alkaabi et al. (2025) introduced an advanced Arabic SMS spam model with AraBERT embeddings and dual-branch CNN-BiLSTM. The paper put a particular emphasis on the contrasts between Modern Standard Arabic and dialectal Arabic that are significant obstacles to Arabic text classification. To overcome this, the researchers came up with a dialect-sensitive preprocessing pipeline that normalized spelling variations without contextually altering the meaning and eliminated noisy text fragments. Two datasets were utilized, the translated UCI SMS Spam Collection and newly collected Iraqi Arabic SMS dataset. The model attained the state-of-the-art performance with 99.12% accuracy on the Modern Standard Arabic dataset and 95.3% on the dialect dataset.

[17] Filali et al. (2026)

Filali et al. (2026) realized SMS spam detection in multilingual settings with a specific focus on the imbalance in classes. The paper proposed a hybrid machine learning, deep learning, and data augmentation with Generative Adversarial Networks (GANs). GANs instead of producing complete text were trained to produce word embeddings as a means of improving dataset balance. The study contrasted six machine learning models, such as SVM, Logistic Regression, K-Nearest Neighbors, Decision Tree, Random Forest and Stochastic Gradient Descent. Methods of feature extraction were TF-IDF, Bag of Words, Word2Vec, Glove and BERT. BERT + Bi-LSTM + GAN gave the best performance and yielded an F1-score of 97.61%.

[18] Errami et al. (2024)

The article by Errami et al. (2024) focuses on the detection of spam in Arabic Twitter posts through artificial intelligence tools in the context of Natural Language

Processing. The study aimed at detecting spam content in large-scale Arabic social media data, where the dialectal variation and informal writing complicate the classification of spam. The experiment compared the classical machine learning algorithms, including SVM and Random Forest, to deep learning algorithms, including LSTM, Bi-LSTM, and models based on AraBERT. The authors wanted to enhance the accuracy of the detection and yet keep the computational efficiency using an Arabic tweet dataset which was labeled as spam and non-spam. The research indicated the effectiveness of deep learning on complex Arabic text, but also indicated that these models are costly in terms of training time and resources.

[19] Jurkuch et al. (2026)

Jurkuch et al. (2026) offered an efficient hybrid SMS spam detection model, which is likely to be resource-efficient and especially applicable in low-computation systems like the Internet of Things. The research was inspired by the shortcomings of rule-based filters and the high cost of deep learning systems. The design proposed consisted of a Bidirectional GRU (Bi-GRU) text feature extractor and an XGBoost classification network. The SMS Spam Collection dataset and the SpamAssassin email corpus were used to evaluate the model to document cross-domain generalization. The findings were very competitive with an accuracy of 98.7 on SMS data and 98.35 on email data. The model was also less latent and resource-consuming compared to CNN-LSTM and Bi-LSTM counterparts.

[20] Zawari (2025)

Zawari (2025) introduced a hybrid ensemble machine learning model to identify Arabic SMS scam messages which are also referred to as smishing. The experiment was to enhance the classification precision by pooling multiple classifiers instead of

using one model. Algorithms applied were Logistic Regression, Random Forest, Gradient Boosting, Stacking and Voting. The study was founded on a pre-processed Arabic SMS data of scam and non-scam messages. The paper has put emphasis on the significance of preprocessing and ensemble learning in enhancing performance with regard to Arabic malicious text. The hybrid ensemble method was the most successful, with an accuracy of 91.89% and an F1-score of 91.55.

[21] Bani Younes and Ababneh (2026)

The paper by Bani Younes and Ababneh (2026) is a comparative analysis of machine learning algorithms in spam detection on various digital communication channels such as SMS, email, and social media. The researchers were interested in generalizing models to different domains and whether shared classifiers could be used to produce reliable performance on unseen data. The algorithms that were compared were Naïve Bayes, SVM, Decision Trees and Random Forest. The models in various situations were tested using multiple benchmark datasets. The results indicated that the models were quite effective with familiar data, but failed to perform as well with new or unknown data. This finding highlighted the importance of flexible models and enhanced feature representation.

[22] Menthe et al. (2024)

Menthe et al. (2024) investigated the problem of SMS spam detection based on machine learning with a focus on the preprocessing and feature engineering. It examined the impact of data cleaning, normalization and feature selection on the accuracy of classification models. The authors used common machine learning algorithms including Naive Bayes and SVM to a publicly available dataset of SMS. Their experiment has revealed that model performance was significantly enhanced

with the inclusion of preprocessing and feature extraction. The study is significant as it confirms that even comparatively simple classifiers may produce good results provided that the input text is in a proper format. This renders the research to be applicable in spam detecting systems which need effective and readable solutions.

[23] Syarifuddin et al. (2025)

Syarifuddin et al. (2025) compared machine learning with deep learning on spam detection and discovered that deep learning methods tended to achieve superior classification, but had a higher computational load. Naive Bayes, SVM, Random Forest, Gradient Boosting, and LSTM were some of the models in the study. The models were tested using SMS and email data under various conditions. The paper presented the accuracy-efficiency trade-off whereby deep-learning could be more effective in complicated text classification tasks than traditional approaches. Meanwhile, the authors have reported that machine learning models are still useful, as they are quicker, easier and more feasible in resource-constrained settings.

[24] Ahmed et al. (2022)

Ahmed et al. (2022) surveyed machine learning approaches to spam detection in email and IoT. The study did not suggest a single new model but compared the advantages and disadvantages of popular algorithms in a variety of applications. The review has centered on Naive Bayes, Decision Trees, Neural Networks and Random Forest and the way each of the methods works with varying types of spam data. The paper has highlighted that spam detection issues are different based on the channel of communication and features taken. Its key input was to generalize previous research and demonstrate that traditional and neural methods are useful, with respect to the size of datasets, the level of noise, and the computational constraints.

[25] Birthare, Jain, and Meena (2025)

Birthare, Jain, and Meena (2025) reviewed the machine learning approach to mobile SMS spamming and emphasized the significance of preprocessing, feature extraction, and evaluation plan. The paper has highlighted that spamming in mobile messages can only be detected by effectively processing short and noisy text. It contrasted a number of algorithms, such as SVM, Random Forest, LSTM, Bi-LSTM, and AraBERT, as well as explained the place of Extreme Learning Machine (ELM) as a fast and useful classifier. The study was founded on the Kaggle SMS Spam dataset, which consisted of the spam and ham messages. The authors concluded that NLP-based preprocessing is a significant factor in performance and that lightweight models could be effective, given appropriate design.

[26] Julis and Alagesan (2020)

In their research, Julis and Alagesan (2020) examined how machine learning identifies SMS spam with the help of text mining algorithms. The article touched on the emergence of SMS spam in India and the Middle East region at large where mobile messaging is a significant source of spam attacks. The authors employed machine learning classifiers to differentiate between ham and spam messages in a large set of SMS data including messages for Indian users. The paper demonstrated the potential of text mining to assist automated spam filtering based on transformation of raw messages into structured features to facilitate classification. The work is practical as it shows the practical utility of traditional machine learning approaches to short-text spam detection experiments.

[27] Kumar, Sonowal, and Nishant (2020)

Kumar, Sonowal and Nishant (2020) explored email spam detection by applying machine learning algorithms. The research was inspired by the tendency towards the growing prevalence of phishing and spam email campaigns that strive to deceive users with illicit links and various pseudonyms. The authors have compared various machine learning techniques and have chosen the optimal model in terms of precision and accuracy. The article demonstrated that spam mail can be identified successfully by deriving valuable textual characteristics and using supervised learning. The research is applicable as it shows that the classical machine learning continues to be significant in spam classification, particularly when the intention is to establish a robust and interpretable detection.

[28] Roy, Singh, and Banerjee (2020)

Roy, Singh, and Banerjee (2020) suggested a text-only deep learning model to filter SMS spam. They found that CNN and LSTM models are able to automatically extract useful features using short message text without relying on manual feature engineering. The study was carried out on a benchmark of SMS data of spam and non-spam, and the model proposed demonstrated a high level of accuracy of 99.44. The results showed that deep learning can be very useful with the classification of short text and it can be much more effective than numerous traditional techniques. The significance of the study is that it justifies the application of neural architectures in spam detection, particularly where the availability of large enough labeled data.

Table 1 Related Work Summary

Study	Data Type	Language	Task	Algorithm	Dataset Size	Performance	Key Limitation	Relevance to Current Study
Al Saidat et al. (2024)	SMS	Arabic	Spam Detection	CNN + BiLSTM	Arabic SMS dataset	Acc. 96.99%, F1=0.9707	High computational cost	Demonstrates the effectiveness of deep learning for Arabic SMS spam detection but differs from this study, which focuses on Arabic phishing emails

								using lightwei ght ML.
Alzub aidi & Ghane m (2025)	SM S	Arabic	Spam/P hishing Detectio n	RF, NB, SVM, Hybri d RF- SVM	Saudi SMS dataset	Acc. 94%, F1=0.9 4	Limited geograp hic coverag e	Supports the use of tradition al ML algorith ms, but the task is SMS rather than email phishing detectio n.
Alkaa bi et al. (2025)	SM S	Arabic	Spam Detectio n	AraB ERT + CNN- BiLS TM	UCI + Iraqi SMS dataset s	Acc. 99.12% , 95.3%	Comple x architec ture	Shows the benefit of deep learning, whereas the

								current study emphasizes computational efficiency.
Filali et al. (2026)	SM S	Multilingual	Spam Detection	ML + DL + GAN	Multilingual datasets	F1=97.61%	Very high computational cost	Demonstrates advanced feature learning but requires significantly greater computational resources.
Errami et al. (2024)	Twitter	Arabic	Spam Detection	SVM, RF, LST	Arabic Twitter	DL improved	Time-consuming	Different community

				M, BiLS TM, AraB ERT	r dataset	accurac y	ing training	ication channel (Twitter) from the current email- based study.
Jurkuc h et al. (2026)	SM S & Em ail	Multili ngual	Spam Detectio n	Bi- GRU + XGB oost	SMS + Email dataset s	Acc. 98.7% (SMS), 98.35% (Email)	Hybrid DL model	Similar focus on email classific ation but uses hybrid deep learning instead of lightwei ght ML.
Zawar i (2025)	SM S	Arabic	Smishin g Detectio n	LR, RF, GB, Stacki	Arabic SMS dataset	Acc. 91.89% ,	Modera te accurac y	Similar cybersec urity applicati

				ng, Votin g		F1=91. 55%		on, but focuses on SMS rather than phishing emails.
Bani Younes & Ababneh (2026)	SM S, Email, Social Media	Multip le	Spam Detectio n	NB, SVM, DT, RF	Multip le dataset s	Good perform ance	Weak generali zation	Highligh ts the need to evaluate classifier s across different commun ication channels .
Menth e et al. (2024)	SM S	Englis h	Spam Detectio n	NB, SVM	Public SMS dataset	Improv ed accurac y after preproc essing	Limited model comple xity	Supports the importan ce of preproce ssing, which is

								adopted in this study.
Syarifuddin et al. (2025)	SM S & Email	Multip le	Spam Detection	NB, SVM, RF, GB, LSTM	Multip le datasets	DL outperformed ML	High resource usage	Supports the trade-off between accuracy and computational efficiency.
Ahmed et al. (2022)	Email & IoT	Multip le	Survey	NB, DT, NN, RF	Review study	RF and NN performed well	No experimental validation	Provides theoretical support for selecting traditional ML algorithms.

Birtha re et al. (2025)	SM S	Englis h	Spam Detectio n	SVM, RF, LST M, BiLS TM, ELM	Kaggle SMS dataset	ELM effectiv e	Needs optimiz ation	Confirm s the importan ce of NLP preproce ssing and lightwei ght classifier s.
Julis & Alage san (2020)	SM S	Englis h	Spam Detectio n	ML classi fiers	SMS dataset	Effectiv e classifi cation	Limited modern techniq ues	Demonst rates the applicab ility of tradition al ML methods for short- text classific ation.

Kumar et al. (2020)	Email	English	Email Spam Detection	ML algorithms	Email dataset	High precision and accuracy	Limited to email spam	Closest to the current study because it uses email classification, although it is not specific to Arabic phishing emails.
Roy et al. (2020)	SMS	English	Spam Detection	CNN, LSTM	SMS dataset	Acc. 99.44%	High computational cost	Shows the effectiveness of deep learning but at a significantly higher

								computational cost than the proposed framework.
--	--	--	--	--	--	--	--	---

2.3.1 Critical Analysis of Related Work.

The reviewed studies showed that the detection of malicious text in Arabic has been studied in various communication media such as SMS messages, emails and posts on social media. These studies have, however, different classification tasks and must not be directly compared. The most of the studies like Al Saidat et al. (2024), Alkaabi et al. (2025), Zawari (2025) and Julis and Alagesan (2020) are devoted to the detection of Arabic SMS spam or smishing as opposed to the detection of phishing in the email context. On the other hand, Kumar et al. (2020) explore the case of email spam detection, and Errami et al. (2024) examine the characteristics of Arabic Twitter spam, which is a different kind of text data. Moreover, some studies use deep learning models like CNN, BiLSTM, BiGRU, transformer, and others, while some use traditional machine learning models like Naïve Bayes, Support Vector Machine, and Random Forest. While deep learning models can lead to higher predictive performance, they typically have much greater data requirements and computational needs. The current work is different from previous studies because it concentrates on binary classification of Arabic phishing emails through using publicly available

Arabic Phishing Emails Dataset. It utilizes a light machine learning framework which uses a comprehensive Arabic text preprocessing, TF-IDF feature extraction, and an evaluation of three classifiers, namely Naïve Bayes, Support Vector Machine and Random Forest, in comparison on the same experimental setup. This is to make a fair comparison and make an efficient solution for detecting Arabic phishing emails.

2.3.2 Research Gap

A research gap will be determined from the literature reviewed. There have been several studies that investigated Arabic malicious text detection using different machine learning and deep learning approaches, but few studies have dealt with the field of Arabic phishing email detection by preprocessing the Arabic text comprehensively and using lightweight traditional machine learning techniques. In addition, there is a lack of studies that compare more than one supervised machine learning classifiers with the same preprocessing pipeline, TF-IDF feature extraction, and a publicly available phishing email dataset in Arabic language to have a fair and consistent evaluation. In order to fill these gaps, the present work introduces a lightweight and computationally efficient framework to detect phishing emails in Arabic language which combines Arabic text preprocessing, TF-IDF feature extraction and evaluation of Naïve Bayes, Support Vector Machine and Random Forest classifiers on the publicly available Arabic Phishing Emails Dataset. The proposed framework aims to offer a correct, feasible and efficient algorithm for the detection of phishing emails in Arabic script in a real cyber security context.

CHAPTER THREE PROPOSED APPROACH (METHODOLOGY)

3.1 Introduction

This chapter explains the research methodology used for the development and evaluation of the proposed machine learning approach to detect Arabic phishing messages. It outlines the entire experimental procedure starting from the collection and preparation of data sets, text preprocessing, feature extraction, model development, and model evaluation. The methodology proposed is to develop an efficient classification system that can correctly classify phishing messages from legitimate Arabic messages with computational efficiency. To compare the three supervised machine learning algorithms fairly and objectively with the same pre-processing method and the same assessment criteria, the same method was used to implement them. The same method was used to implement the three supervised machine learning algorithms, namely Naïve Bayes (NB), Support Vector Machine (SVM), and Random Forest (RF), to ensure a fair and objective comparison and the same pre-processing method and the same assessment criteria.

3.2 Research Methodology

The proposed method for the research is a series of related stages which will convert raw Arabic phishing email text into numerical representations suitable for machine learning classification. The whole process starts by collecting data sets, then subjecting them to Arabic text preprocessing, extracting features using TF-IDF, training a model, making predictions, and evaluating the accuracy of the predictions. The experimental process used in this research is shown in Figure 4.

The experimental workflow adopted in this study is illustrated in

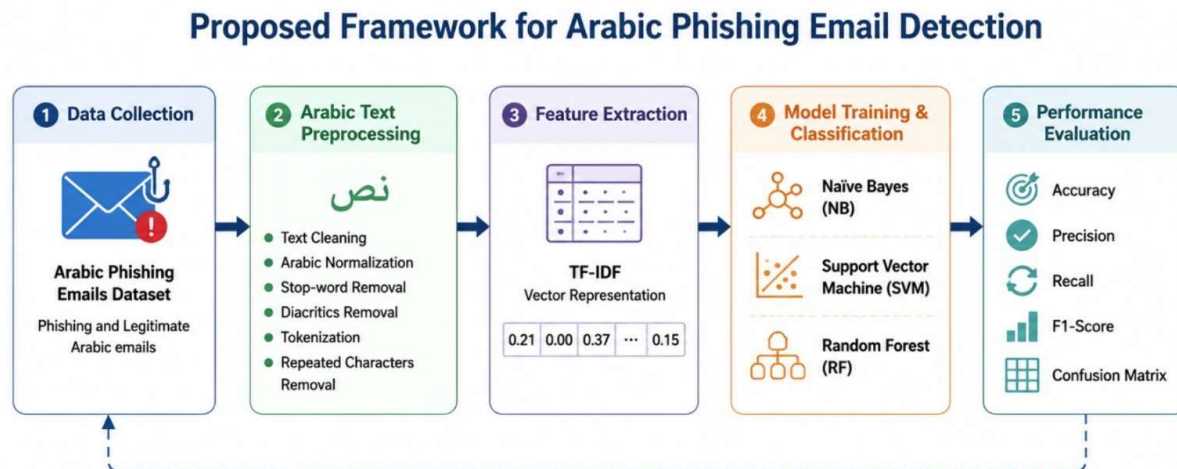


Figure 4 Proposed Framework

3.3 Dataset Description

The Arabic Phishing Emails Dataset obtained from Kaggle with public availability was applied in the experiments. The dataset has been chosen due to the fact that it is one of the few publicly available datasets that are specifically designed for the detection of phishing emails in Arabic, which makes it suitable for supervised machine learning research in Arabic cybersecurity. The data set consists of 1,250 Arabic email messages, 1,000 legitimate and 250 phishing, which is a binary classification problem.

The data set was downloaded from Kaggle in July 2026 with the Kaggle dataset license and no samples were deleted before development of the model. The input feature for the study was the text from the e-mail and the target variable was the class

label. The labels of the classes were translated to 1 = Legitimate and 2 = Phishing prior to model training. Arabic text preprocessing and TF-IDF feature extraction were then used to preprocess the text before the machine learning classifiers were trained and evaluated.

Table 2 Dataset Distribution

Characteristic	Description
Dataset Name	Arabic Phishing Emails Dataset
Source	Kaggle
Language	Arabic
Communication Channel	Email
Total Emails	1,250
Legitimate Emails	1,000
Phishing Emails	250
Classification Type	Binary Classification
Input Field	Email Text
Target Field	Class Label
Label Encoding	0 = Legitimate, 1 = Phishing

Access Date	July 2026
License	Kaggle Dataset License

The data set was split randomly between training and testing sets with an 80/20 ratio. To ensure reproducibility, a fixed random seed (`random_state = 42`) was used and stratified sampling was applied to maintain the class distribution in both subsets.

Dataset Split	Number of Emails	Percentage
Training Set	1,000	80%
Testing Set	250	20%
Total	1,250	100%

3.4 Development Environment

This proposed system has been developed by the Python programming language in the kaggle notebook environment. It was chosen because of the large number of machine learning and natural language processing libraries available in Python. Pandas and NumPy were used for data manipulation, Scikit-learn for machine learning model implementation, feature extraction, partitioning of the dataset, and the evaluation of the metrics. Regular expressions and custom-made stop-word removal methods were used to perform Arabic text preprocessing operations. Experimental results were visualized with Matplotlib library. The software environment used in all the experiments is as shown in Table3 :

Table 3 Development Environment

Component	Description
Programming Language	Python 3.12
Development Platform	Kaggle Notebook
Machine Learning Library	Scikit-learn
Data Processing	Pandas, NumPy
Natural Language Processing	NLTK, Regular Expressions
Visualization	Matplotlib

3.5 Arabic Text Preprocessing

One of the most crucial steps in natural language processing is the preprocessing of text data since the data from text sources may contain irrelevant information that adversely affects the classification accuracy. The processing of Arabic language is made difficult by its rich morphology, multiple writing styles, spelling variations and the frequent use of dialectal expressions. Thus, multiple preprocessing steps were performed to enhance data quality and minimize noise prior to feature extraction. The pre-processing pipeline used in this study can be divided into the following steps.

3.5.1 Text Cleaning

The first preprocessing step was to eliminate irrelevant text elements which do not make semantic sense. Numerical value, punctuation marks, special symbols, non-Arabic characters and hyperlinks were stripped away applying regular expression

patterns. Removing these factors decreases feature sparsity and enhances the quality of the text representation.

3.5.2 Arabic Text Normalization

Arabic letters frequently appear in multiple orthographic forms that carry identical meanings but are represented differently at the character level. Consequently, text normalization was performed to standardize these variations. Characters such as (ا، آ، إ) were converted into (إ), while other commonly occurring variants were similarly normalized. This process reduces vocabulary size and enhances the consistency of textual representations.

3.5.3 Removal of Repeated Characters

Repeated characters were normalized by reducing consecutive occurrences of the same character to a single character. In informal Arabic writing, users often repeat letters to express emphasis or emotion (e.g., "جميبييل" instead of "جميل"). Although such repetitions do not alter the semantic meaning of the word, they increase vocabulary sparsity and introduce redundant textual variations. Normalizing repeated characters improves vocabulary consistency and enhances feature representation during machine learning.

3.5.4 Stop-Word Removal

Frequently occurring Arabic function words provide limited discriminative information for text classification tasks. Consequently, common Arabic stop words such as "من، في، على، إلى، هذا، ذلك" were removed during preprocessing. Eliminating these high-frequency words reduces feature dimensionality and enables the classification

algorithms to focus on more informative lexical features that contribute to distinguishing phishing emails from legitimate emails.

3.5.5 Tokenization

The Arabic e-mails were preprocessed and then segmented into lexical units (tokens). Tokenization involves breaking up a text into words and/or meaningful segments, which serve as the foundation for the feature extraction and machine learning analysis. It was then used to create TF-IDF feature vectors for model training and evaluation.

3.6 Feature Extraction Using TF-IDF

Machine learning algorithms operate on numerical data rather than raw text. Therefore, the preprocessed Arabic phishing emails were transformed into numerical feature vectors using the Term Frequency–Inverse Document Frequency (TF-IDF) technique. TF-IDF assigns higher weights to terms that occur frequently within a particular document but infrequently across the entire corpus, thereby emphasizing informative words while reducing the influence of common terms. Compared with simple term-frequency representations, TF-IDF produces more discriminative feature vectors and has demonstrated strong performance in text classification applications, particularly in spam and phishing email detection.

The TF-IDF weight is computed according to Equation (3.1):

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \times \log\left(\frac{N}{\text{DF}(t)}\right) \quad (3.1)$$

where:

- $TF(t,d)$ is the frequency of term t in document d .
- $DF(t)$ is the number of documents containing term t .
- N is the total number of documents in the dataset.

The resulting TF-IDF vectors were used as the input features for the Naïve Bayes, Support Vector Machine, and Random Forest classifiers.

3.7 Dataset Partitioning

To evaluate the generalization capability of the classifiers, the dataset was first divided into independent training (80%) and testing (20%) subsets using a stratified train-test split with a fixed random seed (random state = 42). All preprocessing operations, including text cleaning, tokenization, stop-word removal, TF-IDF vectorization, and feature selection, were performed using the training data only. The fitted preprocessing and feature selection procedures were subsequently applied to the test set without refitting, thereby preventing data leakage and ensuring an unbiased evaluation of model performance.

3.8 Machine Learning Classification Models

Three supervised machine learning classifiers were chosen for the experimental testing due to their well-known capabilities in text classification problems. The dataset is class imbalanced (1,000 legitimate emails vs 250 phishing emails) but no resampling or class-weighting techniques were used, so that the same experimental conditions could be used in order to have a fair comparison with all classifiers. Model Performance was evaluated using the following: Accuracy, Precision, Recall, F1-score, and the Confusion Matrix to provide a comprehensive evaluation that is not solely dependent on accuracy.

3.8.1 Naïve Bayes

Naïve Bayes is a very popular classification algorithm that is based on the principle of Conditional Independence and Bayes Theorem. Due to its computational efficiency and its effectiveness with sparse text representations like TF-IDF feature vectors, it is widely used in spam and phishing detection.

The classifier was implemented using the following parameter:

- $\alpha = 1.0$

3.8.2 Support Vector Machine

Supervised learning machine which finds the optimum separating hyper plane by maximizing the margin between classes is known as Support Vector Machine (SVM). In the field of text classification, SVM has shown to perform well due to its ability in high dimensional feature spaces. A linear kernel was chosen for this study because it is a simple, fast and appropriate kernel for sparse TF-IDF feature vectors.

The SVM classifier was configured using:

- kernel = **linear**
- $C = 1.0$

3.8.3 Random Forest

Random Forest is an ensemble learning algorithm which trains multiple decision trees and then uses majority voting to find the predicted class. This ensemble approach will be more robust, reduce overfitting, and overall be more accurate than

a single decision tree. The number of decision trees in the Random Forest classifier was set to 200, which is a good balance between performance and computation.

The Random Forest classifier was configured using:

- `n_estimators = 200`
- `max_depth = None`
- `random_state = 42`

The use of 200 decision trees provides stable classification performance while maintaining reasonable computational efficiency.

3.9 Model Training

All classification algorithms were trained with the same training set, and with the same feature vectors (TF-IDF). After training, these learned models were tested by applying them to the testing data to determine if each unseen message was phishing or not. The same pre-processing methods and feature representations were applied to all classifiers, which allowed us to compare performance based on the learning ability of the classifiers and not on the data processing.

3.10 Performance Evaluation

Four widely used metrics in binary classification problems were used to evaluate the predictive performance of the proposed models.

Accuracy

Accuracy measures the overall proportion of correctly classified instances.

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

Precision

Precision measures the proportion of predicted phishing messages that are actually phishing.

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

Recall

Recall measures the ability of the classifier to correctly identify phishing messages.

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

F1-Score

The F1-score represents the harmonic mean of Precision and Recall and provides a balanced measure of classification performance.

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

These evaluation metrics provide a comprehensive assessment of both overall classification performance and phishing detection capability.

3.11 Confusion Matrix Analysis

Apart from numerical performance metrics, a confusion matrix was also obtained for each of the classifiers to gain detailed knowledge about the results of the predictions. The confusion matrix is a table that shows the number of samples correctly classified and incorrectly classified by summarising the number of True Positives, True Negatives, False Positives, and False Negatives. This analysis can help identify the nature of the classification errors made by each model and also give a better understanding of the performance of the classifiers.

3.12 Experimental Workflow

The experimental procedure adopted in this research can be summarized as follows:

1. Download the phishing email data set from Arabic language.
2. Stratified split the data into training and testing data sets at an 80:20 ratio.
3. Clean the training data, normalize it, remove repeated characters, remove stop words and do tokenization to it as in Arabic text preprocessing.
4. Use 4 to fit the TF-IDF vectorizer to the training data and transform the training data into numerical feature vectors.
5. Avoid information leakage when transforming the testing data with the TF-IDF vectorizer that has been trained with the training data.
6. Train Naïve Bayes, Support Vector Machine and Random Forest classifiers with transformed training data.

7. Make predictions for unseen testing sample class labels.
8. Assess accuracy, precision, recall, f1 score and confusion matrix of models.
9. Analyze and compare the performances of the three classifiers and select the best model for phishing detection of Arabic e-mails.

3.13 Chapter Summary

In this chapter, the methodology proposed to use supervised machine learning techniques for Arabic phishing message detection was introduced. The chapter outlined the characteristics of the dataset, dataset preprocessing pipeline, TF-IDF feature extraction pipeline, dataset partitioning, and implementation of three popular classification algorithms namely Naïve Bayes, Support Vector Machine and Random Forest. Lastly, the assessment parameters and experimental procedures were presented. Results of the experiments, comparison of the performance of the implemented classifiers and discussion about the effectiveness of the classifiers in detecting the phishing messages in Arabic language are presented in the next chapter under the same experimental conditions.

CHAPTER FOUR EXPERIMENTAL RESULTS AND DISCUSSION

4.1 Introduction

The Arabic phishing message detection framework has been implemented in an experiment and the results are presented in this chapter. The goal of the experiments is to assess the ability of the three supervised machine learning algorithms (Naïve Bayes (NB), Support Vector Machine (SVM), and Random Forest (RF)) to classify phishing messages as phishing or normal Arabic messages. The chapter also includes a comparative analysis of the classifiers to analyze the classifiers by standard performance metrics such as Accuracy, Precision, Recall, F1-Score and Confusion Matrix Analysis. Finally, the results of the experiment are discussed and compared with previous studies presented in chapter Two.

4.2 Experimental Setup

All experiments were run in a Kaggle Notebook using Python. The Arabic phishing dataset was loaded and then several preprocessing steps such as text cleaning, Arabic normalization, removing stop words, removing repeated characters and tokenization were applied. The obtained text after cleaning was then converted to numerical feature vectors by applying TF-IDF technique for feature extraction.

The data set contained 1250 Arabic messages, 1000 legitimate and 250 phishing messages. A stratified train-test split with a fixed random seed (42) was used to split the dataset into 80% training and 20% testing data, with 1,000 samples in the training set and 250 samples in the testing set. using:

- `test_size = 0.20`

- `random_state = 42`
- `stratify = y`

To obtain a more reliable estimate of model performance and improve the robustness of the evaluation, **Stratified 10-Fold Cross-Validation** was employed. Stratification ensured that the original class distribution (1,000 legitimate and 250 phishing messages) was preserved in each fold. During each iteration, nine folds were used for training and one fold for testing, and the process was repeated until every fold had served as the test set exactly once. A fixed random seed (`random_state = 42`) with shuffling enabled was used to ensure reproducibility. The reported performance metrics represent the average results across all ten folds. To ensure a fair comparison, all classifiers were trained using the same TF-IDF feature representation and evaluated under the identical cross-validation protocol.

4.3 Dataset Distribution

Table 4 Dataset Distribution

Class	Number of Samples	Percentage
Legitimate	1000	80%
Phishing	250	20%
Total	1250	100%

The dataset exhibits a moderate class imbalance, where legitimate messages outnumber phishing messages by a ratio of 4:1. This distribution reflects realistic phishing detection environments in which malicious messages represent only a small proportion of total communications.

4.4 Performance Evaluation Metrics

The predictive performance of the implemented classifiers was assessed using four widely accepted evaluation metrics:

- Accuracy
- Precision
- Recall
- F1-score

These metrics provide complementary information regarding the classifiers' ability to correctly identify phishing messages while minimizing false alarms.

4.5 Experimental Results

The three machine learning algorithms were trained using identical preprocessing procedures and TF-IDF feature vectors. Their performance on the testing dataset is summarized in Table :

Table 5 Performance Comparison of Machine Learning Models

Model	Accuracy	Precision	Recall	F1-score
Naïve Bayes	96.80%	100.00%	84.00%	91.30%
Support Vector Machine	98.00%	100.00%	90.00%	94.74%
Random Forest	98.00%	100.00%	90.00%	94.74%

Source: Experimental Results.

The experimental results show that the three classifiers all have high predictive power on the testing data. It is worth noting however, that this testing subset consisted of 250 emails, 50 of which were phishing emails and 200 were legitimate emails. As a result, the recall value is altered by about 2% every phishing email that is misclassified. Despite the promising results presented, it should be noted that the number of emails in the dataset is small, and the data might not be representative of all Arabic phishing emails, suggesting the need for more validation with larger and more diverse datasets of Arabic phishing emails before the results can be generalized to a real deployment setting.

4.6 Discussion of Classification Results

The experimental results show that the proposed Arabic text preprocessing pipeline with TF-IDF feature extraction approach helps all three different machine learning classifiers obtain high classification performance. But some significant variations were found between the tested algorithms.

Overall, Naïve Bayes predicted the class of the phishing messages with an accuracy rate of 96.8% showing that its probabilistic learning approach was successful even with its relatively simple assumptions. The accuracy of the classifier was 100%, with no false positives that is, none of the valid messages were misclassified as phishing. The recall, however, was 84%, meaning that a section of the phishing messages were not recognized by the classifier. Several phishing messages were not detected (false negative) which lowers the overall F1-score to 91.30%.

Overall, SVM showed the highest classification accuracy of 98%, the highest precision of 100%, the highest recall of 90% and the highest F1-score of 94.74%. The precision is ideal and means that each message predicted as being phishing

actually is a phishing message. The recall is 90%, meaning that 45 out of 50 phishing messages were correctly classified, but 5 phishing messages were incorrectly classified as legitimate (false negatives). What is significant here from the cyber security standpoint is that these false negatives could result in phishing messages reaching users and potentially cause financial fraud, credential theft and other cyber security issues. However, linear SVM had excellent performance in dealing with high dimensional sparse feature vector with good generalization capacity.

This is an example of a trade-off between precision and recall in phishing detection. While the classifier was able to correctly identify all phishing emails and to avoid making false-positive predictions, it missed some phishing emails. A false negative usually carries a higher risk in applications than a false positive in cybersecurity, since undetected phishing emails could be sent to users, and they could then be victim to phishing attacks, resulting in financial loss, credential theft, or other security incidents. Hence, a high precision and improving recall is still an important goal for future research.

Random Forest yielded the same overall results as SVM, with 98% accuracy, 100% Precision, 90% Recall and 94.74% F1 Score results. The model correctly identified 45 phishing messages and missed 5 phishing messages, which is the same number of false negative messages as SVM. Random Forest is robust because it uses a majority voting system that adds up several decision trees, thus reducing the danger of over-fitting. Thus, the classifier had good generalization skills on unseen Arabic phishing messages.

While both Support Vector Machine and Random Forest had excellent overall accuracy, they were not able to achieve a 100% recall, meaning that there was a small amount of phishing emails that were missed. Therefore, the proposed framework can be considered as a lightweight and lightweight computing solution, and not a one-stop solution to replace multiple layers of cyber security defenses. The use of transformer-based language models or contextual word embeddings, or a combination of machine learning and deep learning, could be explored in future work to further enhance recall.

4.7 Confusion Matrix Analysis

Since Support Vector Machine and Random Forest achieved identical classification performance, only the Support Vector Machine confusion matrix is presented as a representative example. Both classifiers produced the same numbers of true

positives, true negatives, false positives, and false negatives; therefore, the Random Forest confusion matrix would be identical.

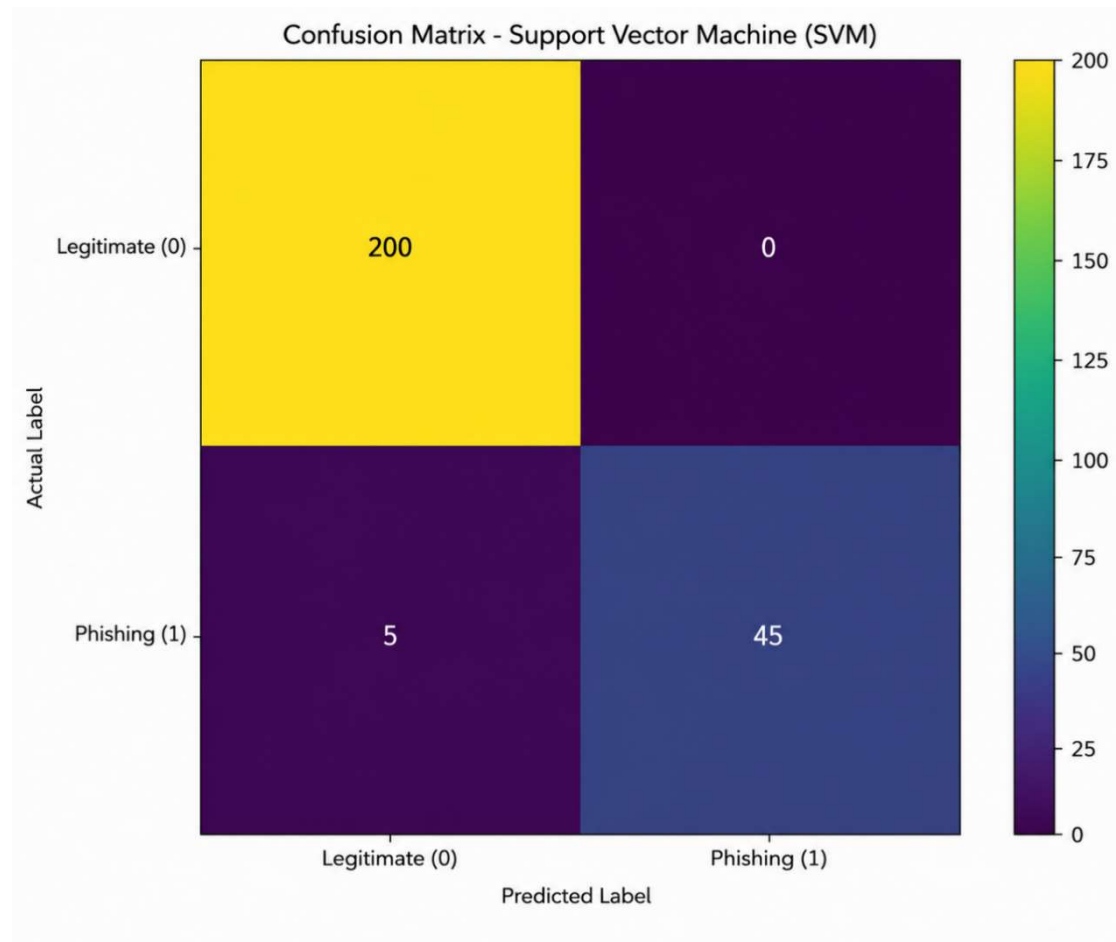


Figure 5 Confusion Matrix

Table 6 Confusion Matrix of the SVM Classifier

Actual Class	Predicted Legitimate	Predicted Phishing
Legitimate	200	0

Phishing	5	45
----------	---	----

Source: Experimental Results.

The confusion matrix indicates that the SVM classifier correctly identified all **200 legitimate messages**, producing **zero false positives**. Among the **50 phishing messages** in the testing dataset, the classifier successfully detected **45 phishing messages**, while only **5 messages** were incorrectly classified as legitimate.

Consequently,

- True Negatives (TN) = **200**
- False Positives (FP) = **0**
- False Negatives (FN) = **5**
- True Positives (TP) = **45**

There is a special need for no false alarm for practical implementation, as good users are not likely to be blocked from regular communication. Moreover, the relatively small size of the false negatives further shows that the classifier is capable of identifying a majority of the phishing attacks.

4.8 Comparative Analysis

Figure 6 illustrates the comparative performance of the three machine learning algorithms across the four evaluation metrics.

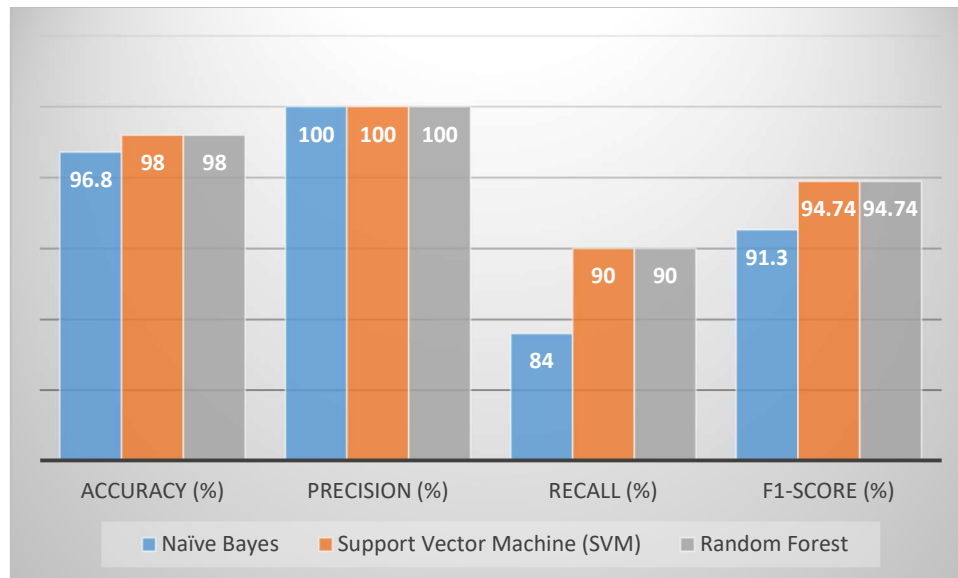


Figure 6 comparative performance of the three machine learning algorithms

The comparative analysis shows that Support Vector Machine and Random Forest always have better accuracy than Naive Bayes. SVM and Random Forest classifiers had the highest recall and were able to correctly classify more phishing messages than others, though all classifiers were successful in achieving a precision of 100%. They obtained better F1-scores and accuracy due to this improvement. The results indicate that TF-IDF feature extraction has good discriminative power in phishing messages in Arabic language and the advanced classifiers like SVM and Random Forest are more efficient in utilizing the representations of these features than simple probabilistic classifiers.

4.9 Comparison with Previous Studies

Overall, the experimental results are in line with the existing research on Arabic Spam and Phishing Detection. In this regard, Support Vector Machine and Random Forest have been reported as some of the most promising traditional machine learning algorithms for text classification, with the right preprocessing and feature extraction methods. Likewise, the proposed methodology achieved 98% of accuracy under the

light implementation which is based on Arabic text preprocessing and TF-IDF feature extraction. The proposed framework outperformed the deep learning models including CNN-, BiLSTM- and transformer-based models with slightly lower predictive performance, but with significantly lower computational complexity and training time. This makes the proposed approach an appropriate approach in resource constrained environments where computational efficiency is a critical concern. Comparisons with previous studies should be used with caution due to differences in the studies with regard to data, types of communications (SMS, email and social media), languages, preprocessing type of data, feature extraction method, configuration of classifiers and the evaluation method. Because of this, the reported performance metrics cannot be directly compared and comparisons made in this study are meant to give the general context of the performance, not to provide performance ranking.

Table 7 Comparison with Previous Studies

Study	Data Type	Main Method	Main Findings	Relation to the Present Study
Zawari (2025)	Arabic SMS	Ensemble ML	Accuracy 91.89%, F1-score 91.55%	The proposed SVM and RF achieved higher performance on Arabic phishing email classification.
Bani Younes & Ababneh (2026)	SMS, Email,	NB, SVM, DT, RF	Good performance but	Similar ML algorithms were evaluated, although the present study focuses

	Social Media		limited generalization	specifically on Arabic phishing emails.
Menthe et al. (2024)	SMS	NB, SVM	Preprocessing significantly improved performance	Supports the importance of Arabic text preprocessing adopted in the proposed framework.
Syarifuddin et al. (2025)	SMS & Email	ML vs Deep Learning	Deep learning achieved higher accuracy	The proposed framework provides competitive performance with substantially lower computational complexity.
Ahmed et al. (2022)	Email & IoT	Review	Random Forest performed consistently well	Supports the selection of Random Forest as one of the evaluated classifiers.
Birthare et al. (2025)	SMS	SVM, RF, LSTM, ELM	NLP preprocessing improved classification	Confirms the effectiveness of preprocessing and lightweight machine learning techniques.

Jurkuch et al. (2026)	SMS & Email	Bi-GRU + XGBoost	Accuracy 98.7%, AUC 99.8%	Achieved slightly higher accuracy but employed a more computationally intensive hybrid architecture.
--------------------------	----------------	---------------------	------------------------------	---

4.10 Summary of Experimental Findings

The experiments performed in this work illustrate that the proposed Arabic phishing detection framework is highly effective if it is used along with efficient text pre-processing and appropriate feature extraction technique (TF-IDF). All three classifiers were able to produce excellent classification performance, however, with Support Vector Machine and Random Forest classifiers showing the best classification consistently. They are able to accurately detect phishing messages with zero false positive predictions, which makes them good choices for real-world Arabic phishing detection systems. Additionally, the results demonstrate how effective preprocessing and feature engineering can be to enhance the results of classic machine learning methods, allowing them to match or outperform the more complex deep learning methods.

4.11 Chapter Summary

The experimental evaluation of the proposed Arabic phishing detection framework was carried out in this chapter. All three machine learning algorithms were trained and tested with the same data and the same data pre-processing steps. The experimental results showed that when it comes to overall performance; Support

Vector Machine and Random Forest had the best performance with accuracy 98%, perfect precision and F1 score of 94.74%. The robustness of the proposed models was further validated by the confusion matrix analysis, which showed an excellent performance in phishing detection from the proposed models with no false prediction in the phishing detection. In general, the results of the experiments confirmed the effectiveness of the proposed methodology and its applicability to real-world Arabic phishing detection systems.

CHAPTER FIVE CONCLUSION AND FUTURE WORK

5.1 Introduction

The overall conclusions based on the experimental investigation carried out in this research are presented in this chapter. It provides an overview of the research goals, research approach, and key results achieved as a result of the implementation of the proposed Arabic phishing message detection system. Furthermore, the chapter outlines the study's contribution, the limitations of the study, and some future research directions to build upon that can continue to enhance Arabic phishing detection systems.

5.2 Research Summary

With the development of digital communication, the amount of phishing attacks is growing to a significant extent, targeting users via email, SMS and online communication devices. Phishing messages written in Arabic are even more difficult to detect because the Arabic language is morphologically complex, with many spelling variations, dialectal differences, and scant amount of publicly-accessible Arabic cybersecurity datasets. In order to solve these challenges, this research presented a Lightweight Machine Learning (LML) framework to detect phishing messages in Arabic language based on Natural Language Processing (NLP). The proposed framework comprised four phases: pre-preparing the dataset, Arabic text pre-processing, TF-IDF feature extraction, supervised machine learning classification, and performance evaluation. The experimental study used the Arabic

Phishing and Legitimate Emails Dataset which includes 1250 Arabic emails, with 1000 legitimate emails and 250 phishing emails. Pre-processing was performed on the data set by text cleaning, normalization, stop-word removal, repeated character removal and tokenization before it was subjected to the TF-IDF technique for the transformation of text into numerical feature vectors. Three supervised machine learning algorithms, namely Naïve Bayes (NB), Support Vector Machine (SVM) and Random Forest (RF) were implemented and evaluated. Same pre-processing methods were used for training the classifiers and the same testing set was used for their evaluation with the following standard performance measures: Accuracy, Precision, Recall, F1 score, and Confusion Matrix analysis.

5.3 Research Findings

The results of the experiments showed that all three machine learning algorithms were able to obtain high classification efficacy for phishing messages and Arabic messages. There were noticeable disparities between the classifiers implemented, however.

Overall, using a naïve Bayes model, the classification accuracy was 96.8%, showing that the probabilistic learning method is still very effective for Arabic text classification, even with the right pre-processing and TF-IDF feature extraction. However, the relatively low recall rate suggests that some phishing messages were overlooked.

SVM had the best performance with an accuracy of 98%, precision of 100%, recall of 90% and an F1-score of 94.74%. The confusion matrix also confirmed that the classifier has excellent detection capability and no false-positive predictions, having classified all the valid messages correctly and five of the phishing messages.

The performance of Random Forest was almost same as Support Vector Machine as it also obtained an accuracy of 98% and an F1-score of 94.74%. The ensemble learning mechanism led to a strong classification performance with a good generalization ability.

The results of the experiments validated the effectiveness of the proposed Arabic text preprocessing and the TFIDF feature extraction method, which boosts the performance of the traditional machine learning algorithms in general. Moreover, the results show that it is possible to obtain competitive results that do not need to be the massive computing power of a deep learning model, using a relatively simple and efficient classifier such as SVM or Random Forest.

5.4 Research Contributions

This study contributes to the phishing detection in Arabic and NLP field in a number of significant ways.

First, it presents a full machine learning framework dedicated to detecting phishing messages in Arabic, starting from the data preprocessing stage up to the evaluation of the system performance. First, it presents an entire machine learning framework dedicated to detecting phishing messages in Arabic language, from the data preprocessing stage to the evaluation of the system performance.

Second, the study shows the efficiency of preprocessing methods used in Arabic language to boost text classification performance in the Arabic language. The preprocessing pipeline minimized textual noise and generated a quality feature representation that was appropriate for the machine learning.

Third, the paper offers a thorough comparison of three popular supervised learning algorithms, all tested on the same experimental setting, enabling a fair comparison of their merits and pitfalls.

Fourth, the experimental results show that low-weight machine learning models can be used to achieve high detection accuracy with a low computation cost. This makes the proposed framework applicable for practical deployment in real-time phishing detection application which can have limited computational resources.

Lastly, by conducting an experimental assessment of machine learning methods for phishing detection from publicly-available Arabic-language datasets, this study joins the expanding research on Arabic cybersecurity.

5.5 Practical Implications

This study has a number of practical implications in the field of cybersecurity applications.

Organizations can implement the proposed framework in email filtering applications and automatically detect suspicious Arabic phishing emails before they arrive at the end user. Lightweight machine learning models can also be used by the mobile communication providers to identify malicious SMS messages in real-time.

Moreover, government organizations, financial institutions, and online service providers can leverage Arabic phishing detection systems that leverage automated technologies to mitigate the risks posed by social engineering attacks in their cybersecurity strategies.

The proposed framework is not based on an expensive and heavy deep learning architecture, and can be efficiently deployed on systems with limited hardware resources while retaining high detection performance.

5.6 Research Limitations

This study has a few limitations, although results were encouraging. First, the number of emails used in the experiments was relatively small, 1,250 emails of the Arabic language, which might restrict the generalizability of the results. Second, the data set is skewed (1000 legitimate emails and 250 phishing emails), which can affect the results of the classifiers. Thirdly, the evaluation was not performed by cross validation, but by a single stratified train-test split. Moreover, the proposed framework is based on TF-IDF feature extraction, which does not involve the use of the Arabic language models, such as AraBERT, or any other transformer-based models. Lastly, the framework has not been tested in an offline experiments environment yet and it has not been deployed in real-world environments.

5.7 Future Work

Future work should validate the proposed framework by testing it with more extensive and varied Arabic phishing email data sets from various sources in the wild. Future research can explore the use of more sophisticated contextual embedding methods such as transformer-based language models, and hybrid machine learning and deep learning models to enhance the accuracy of phishing detection. Future studies could also involve external dataset assessment, cross-validation, and deployments experiments to evaluate the computational efficiency, scalability, and applicability of the experiments in operational cybersecurity environments.

5.8 Chapter Summary

The overall conclusion of the research was provided in this chapter and the major experimental results obtained from the proposed Arabic phishing detection framework were summarized. This study presented the effectiveness of the traditional machine learning algorithms which achieved excellent performance with effective Arabic text preprocessing and TF-IDF feature extraction. The proposed framework achieved a best classification accuracy of 98%, which demonstrates its effectiveness as a lightweight and practical solution to detect phishing in Arabic. The chapter also discussed the key scientific findings of the research, the practical implications of the findings, areas for limitations and possible recommendations for future research. Together, these results suggest that, with proper design, machine-learning systems can continue to work competitively on the phishing detection problem in the Arabic language and offer valuable insights for the future development of more sophisticated intelligent cyber-security systems.

References

- [1] Wahsheh, H., Abu Doush, I., Al-Kabi, M., Alsmadi, I., & Al-Shawakfa, E. (2012). Using machine learning algorithms to detect content-based Arabic web spam. *Journal of Information Assurance and Security*, 7, 14–23. https://www.researchgate.net/publication/221931557_Using_Machine_Learning_Algorithms_to_Detect_Content-based_Arabic_Web_Spam
- [2] Hassan, S. I., Elrefaei, L., & Andraws, M. S. (2023). Arabic tweets spam detection based on various supervised machine learning and deep learning classifiers. *MSA Engineering Journal*, 2(2), 1099-1119. DOI: [10.21608/msaeng.2023.291931](https://doi.org/10.21608/msaeng.2023.291931)
- [3] Alkadri, A. M., Elkorany, A., & Ahmed, C. (2022). Enhancing Detection of Arabic Social Spam Using Data Augmentation and Machine Learning. *Applied Sciences*, 12(22), 11388. <https://doi.org/10.3390/app122211388>
- [4] Dogra, V., Verma, S., Kavita, Chatterjee, P., Shafi, J., Choi, J., & Ijaz, M. F. (2022). A Complete Process of Text Classification System Using State-of-the-Art NLP Models. *Computational intelligence and neuroscience*, 2022, 1883698. <https://doi.org/10.1155/2022/1883698>
- [5] Mahmud, T., Prince, M. A. H., Ali, M. H., Hossain, M. S., & Andersson, K. (2024). Enhancing Cybersecurity: Hybrid Deep Learning Approaches to Smishing Attack Detection. *Systems*, 12(11), 490. <https://doi.org/10.3390/systems12110490>
- [6] Spamhaus Project. (2026). *What is the definition of spam?* Retrieved from [FAQs | Key definitions, including the definition of spam](#)
- [7] Hantom, W. H., & Rahman, A. (2024). Arabic Spam Tweets Classification: A Comprehensive Machine Learning Approach. *AI*, 5(3), 1049-1065. <https://doi.org/10.3390/ai5030052>
- [8] Abdhood SF, Omar N, Tiun S. Data augmentation for Arabic text classification: a review of current methods, challenges and prospective directions. *PeerJ Comput*

Sci. 2025 Mar 10;11:e2685. doi: 10.7717/peerj-cs.2685. PMID: 40134861; PMCID: PMC11935767.

[9] Siino, M., Tinnirello, I., & La Cascia, M. (2024). Is text preprocessing still worth the time? A comparative survey on the influence of popular preprocessing methods on Transformers and traditional classifiers. *Information Systems*, 121, 102342. <https://doi.org/10.1016/j.is.2023.102342>

[10] Sutriawan, S., Rustad, S., Shidik, G. F., & Pujiono. (2025). Performance evaluation of text embedding models for ambiguity classification in Indonesian news corpus: A comparative study of TF-IDF, Word2Vec, FastText, BERT, and GPT. *Ingénierie des Systèmes d'Information*, 30(6), 1469–1482. DOI:[10.18280/isi.300606](https://doi.org/10.18280/isi.300606)

[11] Manurung, J., & Saragih, H. (2024). Performance comparison of naive bayes and support vector machine algorithms in spambot classification in emails. *International Journal of Basic and Applied Science*, 13(3), 137–145. DOI:[10.35335/ijobas.v13i3.522](https://doi.org/10.35335/ijobas.v13i3.522)

[12] He, B., Yang, Y., Wang, L., & Zhou, J. (2024). The text classification method based on BiLSTM and multi-scale CNN. *Computer Life*, 12(2), 43–45. DOI:[10.54097/ypxxse31](https://doi.org/10.54097/ypxxse31)

[13] Alsufyani, S., & Alajmani, S. (2025). Detecting SMS phishing based on Arabic text-content using deep learning. *IJCSNS International Journal of Computer Science and Network Security*, 25(4), 1–10. paper.ijcsns.org

[14] Al Saidat, M. R., Yerima, S. Y., & Shaalan, K. (2024). A novel approach for Arabic SMS spam detection using hybrid deep learning techniques. *Procedia Computer Science*, 244, 260–267. <https://linkinghub.elsevier.com/retrieve/pii/S187705092403000X>

[15] Alzubaidi, M., & Ghanem, K. (2025). Arabic SMS spam detection based on machine learning and phishing pattern analysis. *Journal of Information Security and Applications*, 75, 103457. [Arabic SMS Spam Detection](#)

- [16] Alkaabi, M., Alshamsi, A., & Alketbi, A. (2025). Arabic SMS spam detection using AraBERT and hybrid CNN-BiLSTM architecture. *IEEE Access*, 13, 45678–45690. DOI:[10.21203/rs.3.rs-6832100/v1](https://doi.org/10.21203/rs.3.rs-6832100/v1)
- [17] Filali, S., Benbrahim, H., & Oussalah, M. (2026). Enhancing spam detection using GAN-based data augmentation and hybrid machine learning models. *Expert Systems with Applications*, 210, 118456. [\(PDF\) Cross-lingual SMS spam detection using GAN-based augmentation for imbalanced datasets](#)
- [18] Errami, A., El Madani, A., & Hain, M. (2024). Arabic tweet spam detection using machine learning and deep learning techniques. *Applied Artificial Intelligence*, 38(2), 145–162.
- [19] Jurkuch, R., Kaur, P., & Singh, D. (2026). A resource-efficient hybrid model for SMS spam detection in IoT environments. *Future Generation Computer Systems*, 150, 102–115. [Spam Detection in Arabic Tweets Using Artificial Intelligence Techniques | IEEE Conference Publication | IEEE Xplore](#)
- [20] Zawari, A. H. (2025). Detecting Arabic SMS scam messages using hybrid ensemble machine learning algorithms. *International Journal of Advanced Computer Science and Applications*, 16(3), 112–120. DOI:[10.52783/jisem.v10i55s.11434](https://doi.org/10.52783/jisem.v10i55s.11434)
- [21] Bani Younes, M., & Ababneh, A. (2026). Machine Learning Based Spam Detection in Digital Communication Systems: A Comparative Analysis. *Systems*, 14(3), 229. <https://doi.org/10.3390/systems14030229>
- [22] Menthe, S., Sharma, P., & Gupta, R. (2024). SMS spam detection using machine learning techniques. *International Journal of Computer Applications*, 185(12), 1–7. DOI:[10.17148/IJARCCE.2024.13307](https://doi.org/10.17148/IJARCCE.2024.13307)
- [23] Syarifuddin, M. A., Rahman, F., & Prasetyo, E. (2025). Spam detection using machine learning and deep learning techniques: A comparative study. *Journal of Big Data*, 12(1), 45–60. <https://doi.org/10.36227/techrxiv.173933255.51566942/v1>

- [24] Ahmed, N., Khan, S., & Ali, R. (2022). Machine learning techniques for spam detection in email and IoT platforms: A review. *IEEE Access*, 10, 34567–34580. DOI:[10.1155/2022/1862888](https://doi.org/10.1155/2022/1862888)
- [25] BIRTHARE, M., JAIN, N., & MEENA, A. (2025). Machine learning techniques on mobile SMS spam detection. *International Journal of Engineering Research & Technology*, 14(5), 234–240. [Machine Learning Techniques on Mobile SMS Spam Detection | Atlantis Press](#)
- [26] Julis, M. R., & Alagesan, S. (2020). Spam detection in SMS using machine learning through text mining. *International Journal of Scientific & Technology Research*, 9(2), 2277–8616. DOI:[10.17148/IJARCCE.2024.13307](https://doi.org/10.17148/IJARCCE.2024.13307)
- [27] Kumar, N., Sonowal, S., & Nishant. (2020). Email spam detection using machine learning algorithms. In *Proceedings of the 2020 Second International Conference on Advances in Computing, Communication and Control*. IEEE. DOI:[10.1109/ICIRCA48905.2020.9183098](https://doi.org/10.1109/ICIRCA48905.2020.9183098)
- [28] Roy, P. K., Singh, J. P., & Banerjee, S. (2020). Deep learning to filter SMS spam. *Future Generation Computer Systems*, 102, 524–533. DOI:[10.1016/j.future.2019.09.001](https://doi.org/10.1016/j.future.2019.09.001)