Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.057279

ARTICLE

Comparative Analysis of Machine Learning Algorithms for Email Phishing Detection Using TF-IDF, Word2Vec, and BERT

Arar Al Tawil[1,*], Laiali Almazaydeh[2], Doaa Qawasmeh[3], Baraah Qawasmeh[4], Mohammad Alshinwan[1,5] and Khaled Elleithy[6]

[1]Faculty of Information Technology, Applied Science Private University, Amman, 11931, Jordan

[2]College of Engineering, Abu Dhabi University, Abu Dhabi, Al Ain, P.O. Box 1790, United Arab Emirates

[3]Faculty of Artificial Intelligence, Al-Balqa Applied University, Salt, 19117, Jordan

[4]Department of Civil and Construction Engineering, Western Michigan University, Kalamazoo, MI 49008, USA

[5]MEU Research Unit, Middle East University, Amman, 11831, Jordan

[6]Department of Computer Science and Engineering, University of Bridgeport, Bridgeport, CT 06604, USA

*Corresponding Author: Arar Al Tawil. Email: ar_altawil@asu.edu.jo

Received: 14 August 2024 Accepted: 15 October 2024 Published: 18 November 2024

ABSTRACT

Cybercriminals often use fraudulent emails and fictitious email accounts to deceive individuals into disclosing confidential information, a practice known as phishing. This study utilizes three distinct methodologies, Term Frequency-Inverse Document Frequency, Word2Vec, and Bidirectional Encoder Representations from Transformers, to evaluate the effectiveness of various machine learning algorithms in detecting phishing attacks. The study uses feature extraction methods to assess the performance of Logistic Regression, Decision Tree, Random Forest, and Multilayer Perceptron algorithms. The best results for each classifier using Term Frequency-Inverse Document Frequency were Multilayer Perceptron (Precision: 0.98, Recall: 0.98, F1-score: 0.98, Accuracy: 0.98). Word2Vec's best results were Multilayer Perceptron (Precision: 0.98, Recall: 0.98, F1-score: 0.98, Accuracy: 0.98). The highest performance was achieved using the Bidirectional Encoder Representations from the Transformers model, with Precision, Recall, F1-score, and Accuracy all reaching 0.99. This study highlights how advanced pre-trained models, such as Bidirectional Encoder Representations from Transformers, can significantly enhance the accuracy and reliability of fraud detection systems.

KEYWORDS

Attacks; email phishing; machine learning; security; representations from transformers (BERT); text classifeir; natural language processing (NLP)

1 Introduction

The Internet is used all the time by people from everywhere. In 2014, over 40% of people worldwide (primarily in rich countries) used the Internet. NATO sees the Internet as a valuable tool for nations, an essential part of their infrastructure, and a significant driver of progress and economic growth [1]. The

proliferation of the Internet has led to the emergence of malevolent software and malware designed to penetrate computer systems via aggressive attacks and data destruction. The objective of these assaults is to illicitly acquire users' information, such as credit card numbers and passwords, and then surreptitiously disseminate it without their awareness. Malware is computer software that can damage files and networks [2,3]. The risk isn't just for people; it also affects businesses, institutions, and even states, putting citizen and military assets at risk. When these groups lose important information or have their identities hurt, they are vulnerable. In the past few years, there have been many cases of stolen Google intellectual property, user data, and credit and debit card information bought illegally from online payment systems [4,5]. Kaspersky Labs offers one of the many definitions of cybersecurity: Cybersecurity refers to the practice of protecting data, electronic systems, networks, servers, mobile devices, and computers against malicious assaults [6,7]. It is sometimes referred to as electronic information security or information technology security. This notion spans several fields, including end-user education, disaster recovery, and computer security. Cybersecurity safeguards personal, governmental, and corporate data from unwanted access or alteration. There are three critical tasks involved in this context: (a) continuously carrying out and improving these actions, (b) ensuring the quality of protection against different threats, and (c) implementing steps to safeguard hardware, software, and the information they hold [8,9].

In order to enhance the digital security of cybersecurity, it is imperative to continuously adapt and mitigate increasingly intricate attacks through the use of machine learning (ML). In order to analyze substantial quantities of real-time data, identify deviations from conventional patterns, and anticipate potential security hazards, cybersecurity professionals utilize machine learning. This state-of-the-art technology enhances the precision and efficacy of identifying potential hazards and aids in the development of proactive strategies to reduce the extent of future cyber attacks [10,11]. Conventional security methods encounter difficulties in adapting to the ever-changing nature of cyber threats. Performing research on the vast quantities and diverse types of data produced by networks, systems, and individuals is not feasible. A significant advantage of machine learning systems is their capacity to analyze immense quantities of data, adapt to novel attack methods, and identify patterns. Machine learning is essential for constantly adapting to the ever-changing nature of threats [12].

Machine learning is employed in numerous cybersecurity applications to automate incident response, detect abnormal user behavior, foretell potential vulnerabilities, and identify malicious software. Advancements in algorithms may offer cybersecurity professionals a proactive advantage in safeguarding critical digital assets. However, there are certain obstacles associated with integrating these elements. For machine learning models to attain precision and minimize the probability of inaccurate results, it is essential to implement rigorous training and validation procedures. The possibility of adversarial attacks on machine learning models and ethical concerns further complicate the use of automated systems in security scenarios [13,14].

Pernicious individuals employ phony emails to deceive recipients into disclosing sensitive information or installing pernicious malware in an email impersonation attack, a highly sophisticated cyber threat. These emails imitate legitimate sources, including banks, social networking sites, and established businesses, which can make it challenging for users to identify the threat. The primary objectives of email phishing are the acquisition of personal data, including credit card numbers, authentication passwords, and other confidential information.

This information will be exploited by cybercriminals to gain illicit access to accounts, commit identity theft, or generate financial gains. It is imperative to authenticate the sender's email address, avoid clicking on suspicious links, and implement robust security measures such as multi-factor

authentication and antivirus software to prevent email spoofing. But education and awareness are the most effective forms of protection. Effective detection and prevention of email spoofing attacks necessitate a sufficient level of awareness and knowledge.

In the following sections of this work, the organization is as follows: Section 2 offers a comprehensive analysis of the existing literature. The methodology, with respect to the dataset, machine learning algorithm, and performance measurements, is comprehensively described in Section 3. The proposed ensemble learning methodology is the subject of a comprehensive explication in Section 4. Section 5 conducts a comprehensive analysis and evaluation of the experimental results obtained from the two datasets. The discoveries are further elaborated and examined in Section 6. A concise summary and suggestions for additional research are provided in the paper's conclusion.

2 Literature Review

Web-based and email-based phishing are the two primary categories of phishing detection research broadly categorized in this section.

2.1 Web-Based

In Table 1, the exhaustive summary of the prior research on the detection of fraudulent URLs using machine learning algorithms is provided. The phishing datasets, evaluation metrics, performance outcomes, preprocessing stages, and machine learning algorithms that have been implemented are the primary focus. Abutaha et al. [12] meticulously integrated four machine learning methodologies: Support Vector Machine, Random Forest, Gradient Boosting, and Neural Networks. In order to produce 22 distinctive features, we meticulously processed the dataset, which comprises 1,056,937 URLs. Subsequently, various feature reduction methodologies were implemented to enhance the features above. The data preprocessing procedure necessitated the removal of 14,786 duplicate records and the resolution of absent values. F1-score, precision, recall, false positive rate, and accuracy were the five metrics that were implemented to evaluate the algorithms' performance. The Support Vector Machine demonstrated the maximum accuracy rate in identifying the URLs, with a 97.3% accuracy rate, as indicated by the results. This method could be integrated with add-on or auxiliary features in Internet browsers to notify users when they attempt to access a fraudulent website exclusively based on its URL.

Table 1: Previous papers summarization

Ref.	Algorithms	Results
[12]	SVM, Random forest, Gradient boosting, Neural networks	SVM achieved the highest accuracy of 97.3%
[14]	SVM, RF	Accuracy of 97%
[15]	24 classifiers from six groups: Bayes, Functions, Lazy, Meta, Rules, Trees	RF and J48 Has the best result
[16]	Various ML and DL techniques	Accuracy: 0.94
[17]	Various ML algorithms	Accuracy: 0.98
[18]	Various ML algorithms	Accuracy = 0.977
[19]	BiGRU	Accuracy = 0.9739

The spear-phishing assaults are particularly persuasive due to the social and psychological vulnerabilities they exploit. The repercussions of these attacks were mitigated by a multilayered strategy proposed by Samad et al. [14]. This approach utilizes both the content and attachments of an email to safeguard against deceptive attacks. By employing sentiment analysis techniques, such as Random Forest (RF) and Support Vector Machine (SVM) classifiers, the researchers were able to accurately classify web pages as either spam or non-spam, resulting in an exceptional level of precision. The dataset they employed was 3000 websites that were classified as either spam or non-spam on the Kaggle platform. The principal themes in the dataset were identified using Latent Dirichlet Allocation (LDA) for topic modeling. Results indicated that the RF algorithm outperformed others, attaining a 97% accuracy rate during detection.

2.2 Email Based

Rathee et al. [15] employed a variety of machine learning (ML) and deep learning (DL) methodologies to detect fraudulent emails. They trained and tested a variety of classification models using datasets that contained legitimate and fraudulent emails. The performance of these models was evaluated using the F1-score, precision, recall, and accuracy. The most effective model in their investigation achieved an F1-score of 0.975, a precision of 0.983, a recall of 0.968, and an accuracy of 0.975. This discovery emphasizes the critical role of these reliable models in distinguishing between legitimate and fraudulent emails, a crucial aspect of cybersecurity.

Conventional machine learning techniques, including Naive Bayes, Logistic Regression (LR), and Support Vector Machine (SVM), were employed by Unnithan et al. [16] to investigate the detection of fraudulent emails. To evaluate the accuracy of these classifiers, they implemented a dataset that comprised both legitimate and fraudulent emails. The SVM exhibited the highest accuracy of 0.943 among the evaluated models, with Naive Bayes and Logistic Regression following closely behind with 0.934 and 0.922, respectively.

Study [17] implemented an assortment of machine learning algorithms, including Random Forest (RF) and Support Vector Machine (SVM), to categorize emails as either legitimate or fraudulent. The accuracy of their classifications was also improved by the implementation of feature extraction techniques. The training set yielded an exceptional accuracy of 0.98 for the Random Forest model. However, the SVM model significantly outperformed other models on the test set, achieving an F1-score of 0.99, a precision of 0.98, and an accuracy of 0.98.

Study [18] executed an investigation into the detection of fraudulent emails by employing an assortment of machine-learning algorithms. Their primary goal was to improve detection capabilities by implementing more effective preprocessing and feature selection methods. The most effective model was determined to be the one that demonstrated the highest level of accuracy, precision, recall, and F1-score in identifying fraudulent communications.

The article [19] introduced a model for detecting fraud attacks that employ four deep learning algorithms: LSTM, BiLSTM, GRU, and BiGRU, to analyze the text of web pages. The study's innovative approach led to the discovery that GRU and BiGRU outperformed the other models, with BiGRU achieving the highest mean accuracy of 97.39% and GRU closely following at 97.29%. By effectively capturing the sequential and contextual features of text, these models are highly suitable for detecting fraud attacks in web content, inspiring further research and development in the field of cybersecurity and fraud detection.

Table 1 summarizes all the previous papers about phishing in web phishing and email.

3 Methodology

This section outlines the methodology utilized in this paper, which consists of three primary stages: the datasets used, the machine learning algorithms implemented to build the models, and the performance metrics implemented to evaluate the algorithms' efficacy. The first concerns the dataset, the number of instances, and the distribution of emails into two categories: safe and phishing. Then, we will discuss the machine learning model we used in this methodology. Finally, we discuss evaluating the approach using different performance metrics, such as accuracy, recall, precision, and F1-score.

3.1 Phishing Datasets

We utilized a dataset of fraudulent emails obtained from Kaggle [20], encompassing various email-related attributes. This dataset consists of 18,650 instances, each characterized by two attributes: Safe Email and Fraudulent Email. These capabilities enable the classification of emails according to their content and structure. The dataset offers a comprehensive foundation for the training and evaluation of our models, providing a balanced distribution of safe and fraudulent emails. Fig. 1 illustrates the frequency of the email categories, while Fig. 2 presents a word cloud that displays the dataset's most frequently used terms.

3.2 Building Models

Five classification machine learning techniques were employed to analyze each dataset to verify the authenticity of an email. Machine learning algorithms are computational models that enable computers to identify patterns and generate predictions or evaluations based on data without explicit programming. These algorithms are the fundamental building blocks of modern artificial intelligence.

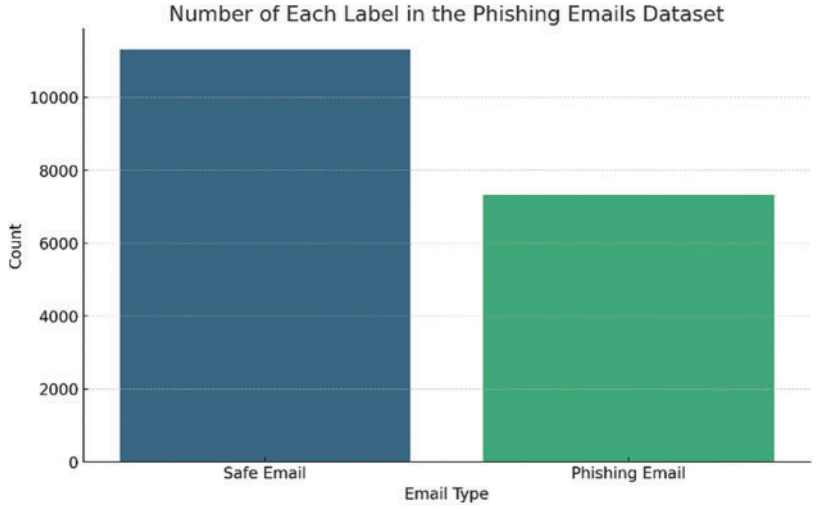


Figure 1: Dataset frequency dataset

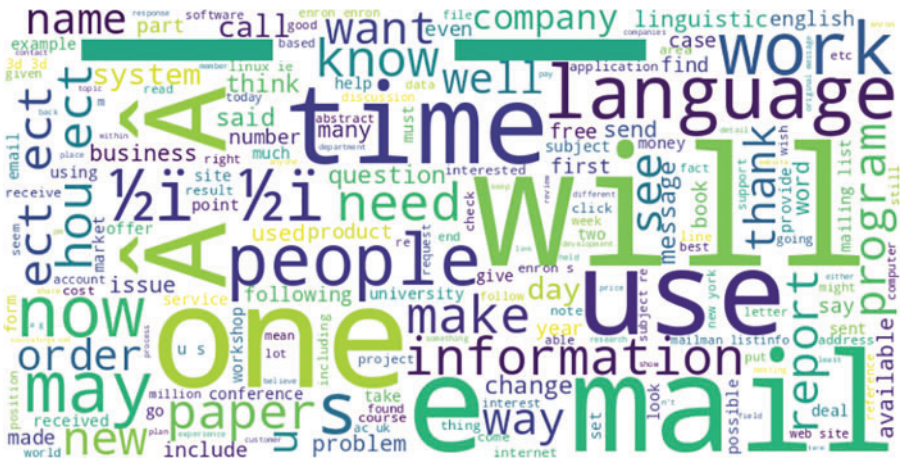


Figure 2: Dataset word cloud

3.2.1 The Random Forest (RF) Algorithm

A technique known as ensemble learning, which incorporates multiple decision trees, is employed by random forests to produce predictions for both classification and regression tasks. In machine learning, ensemble learning provides various benefits, including the ability to address complex problems, resilience, and improved performance. To enhance their predictive capabilities, random forests implement ensemble learning methodologies. The mean for the regression task is computed by the random forest (RF), which also calculates the average prediction for all trees in the classification task [21]. Decision trees are the fundamental components that form the core of random forests. The projections are produced by hierarchical models known as decision trees, which employ binary divisions on characteristics. This is accomplished by partitioning the data into smaller subsets by predetermined criteria [22]. The target variable prediction is ultimately determined. The projections of multiple decision trees are combined in random forests to improve accuracy, mitigate overfitting, and resolve intricate problems. Random forests encompass a variety of data elements by aggregating these decision trees, resulting in more dependable predictions. The RF calculates the final prediction (y) in Eq. (1), where $h_i(x)$ denotes the prediction from each decision tree and N is the number of trees.

$$y(x) = \frac{1}{N} \sum_{i=1}^{N} h_i(x) \tag{1}$$

3.2.2 Decision Tree Algorithm

It's easy to understand decision tree models, which are used for both classification and regression jobs. In decision support systems, a decision tree is a hierarchical model showing the possible results of actions while considering resource costs, utility, and random events. The tree layout starts with a root node in the middle linked to other nodes by branches [23]. Internal and leaf nodes can form a tree-like structure organized in a hierarchy. Data is split into subgroups and groups as part of the algorithm's forecast process. The Gini index, information gain, gain ratio, and entropy are some of the tools used to find the root nodes of these groups. The primary method used in this study for the division procedure is entropy.

3.2.3 Multilayer Perceptron Algorithm (MLP)

Neurons and perceptrons are two words that are often used to describe the parts of a neural network. A neural network is usually built with a feedforward design, which means that where the data moves from the input layer to the output layer in a one direction, going through the hidden levels. Some of the things that Multilayer Perceptrons (MLPs) are used for are regression, classification, and pattern recognition [24]. An MLP has three kinds of layers: output, input, and secret [25].

3.2.4 Logistic Regression

Logistic Regression is a widely used statistical method for binary classification tasks. This technique predicts the probability that a given input belongs to a specific category, typically encoded as 0 or 1. Unlike linear regression, which predicts continuous values, logistic regression employs a logistic function (sigmoid) to map predicted values to probabilities between 0 and 1. This transformation allows it to effectively handle binary outcomes [26].

Mathematically, the logistic regression model is expressed as present in Eq. (2):

$$P(Y = 1 \mid X) = \frac{1}{1 + e^{-\left(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n\right)}} \tag{2}$$

where $P(Y = 1 \mid X)$ is the probability of the outcome being 1 given the input variables $X_1, X_2, \ldots, X_n$, and $\beta_0, \beta_1, \ldots, \beta_n$ are the model coefficients estimated from the data.

3.2.5 Bidirectional Encoder Representations

The 2019 paper by Devlin et al. [27] introduced a new way to think about Natural Language Processing (NLP) that they called Bidirectional Encoder Representations from Transformers (BERT). BERT teaches deep two-way representations on both the left and right sides at all levels. This is done so that it can learn how language works and how things depend on each other. When using the Transformer design, BERT handles long-range dependencies by using self-attention to process input text in parallel. Some language models only work in one way, but the BERT model does two. An MLM task helps it figure out what parts are hidden by looking at what else is nearby. An NSP task helps it figure out how words fit together. BERT made it possible for some of the most cutting edge NLP standards to be made, such as GLUE and SQuAD. Some of the changes that have been made to make it work better are RoBERTa [28], DistilBERT [29], and ALBERT [30]. A lot has changed in the world of NLP because of BERT. It's now the standard way to work in a call center, translate text, figure out how people feel about something, and make systems that answer questions.

BERT, with its sophisticated NLP capabilities, particularly excels in detecting phishing communications. Unlike conventional methods such as TF-IDF and Word2Vec, which often miss nuanced phishing attempts due to their lack of contextual awareness, BERT stands out by accurately interpreting the meaning of words through context analysis. This bidirectional approach equips BERT to spot subtle manipulation tactics and sophisticated wording commonly used in phishing attacks. Its ability to comprehend intricate language patterns and long-term dependencies makes it a perfect fit for detecting email fraud. Our thorough research, which demonstrates the exceptional superiority of BERT over traditional machine learning models, should inform you and instill confidence in its enhanced accuracy and reliability in identifying fraudulent emails through its deep contextual understanding.

3.3 Performance Metrics

Precision, recall, F1-score, and accuracy are the four evaluation metrics used to determine machine learning's experimental results. The formula for each metric is as follows: FN represents False Negative, TP represents True Positive, TN represents True Negative, and FP represents False Positive [31].

- The Accuracy metric quantifies the frequency with which the model's predictions are accurate in every class. The metric is determined through division of the sum of the accurate predictions (including True Positives and True Negatives) by the overall count of predictions generated [31].

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Predictions}} \tag{3}$$

- The Precision metric assesses the accuracy of the model's optimistic predictions. The calculation entails dividing the number of True Positives by the sum of the number of False Positives and True Positives. This metric is highly advantageous when the goal is to minimize the number of false optimistic predictions [31].

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \tag{4}$$

- Also known as the True Positive Rate or Sensitivity, this metric quantifies the model's ability to identify all relevant instances reliably. The calculation entails dividing the number of True Positives by the sum of False Negatives and True Positives. As recall increases, the model's capacity to identify and retain all positive instances is enhanced [31].

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{5}$$

- The balanced average of recall and precision is used to calculate the F1-score. It balances precision and recall by integrating false positives and false negatives into a single metric. The formula can be determined by dividing the product of Precision and Recall by their sum twice [31].

$$F1 \text{ Score } = 2 \times \frac{\text{recall} \times \text{precision}}{\text{recall} + \text{precision}} \tag{6}$$

- Training Time: The time it took for each model to train on the dataset. This metric is not just a measure of computational efficacy, but a key factor in determining the feasibility of deploying the model, especially in large-scale applications where rapid training is a necessity.
- Evaluation Time: The duration necessary for the model to generate predictions based on the test data. This is not just a technical detail, it's a critical factor that determines the feasibility of our model's deployment. When real-time or near-real-time predictions are required, this time becomes even more significant, underlining the relevance of our work.

4 Proposed Approach

4.1 Proposed Approach Overview

We have divided our classification methodology into two components. In the initial phase, conventional machine learning classifiers, such as Random Forests (RF), Multilayer Perceptrons (MLP), and Logistic Regression (LR), are implemented, in addition to feature extraction methodologies such as Word2Vec (W2V) and TF-IDE. BERT (Bidirectional Encoder Representations from Transformers)

is employed in the second section, which employs sophisticated techniques that utilize Large Language Models (LLMs). By conditioning on both left and right contexts in all layers, BERT pre-trains deep bidirectional representations, allowing it to capture intricate relationships and dependencies within the text. Consequently, it is highly effective for classification tasks, the Fig. 3 explains the approach.

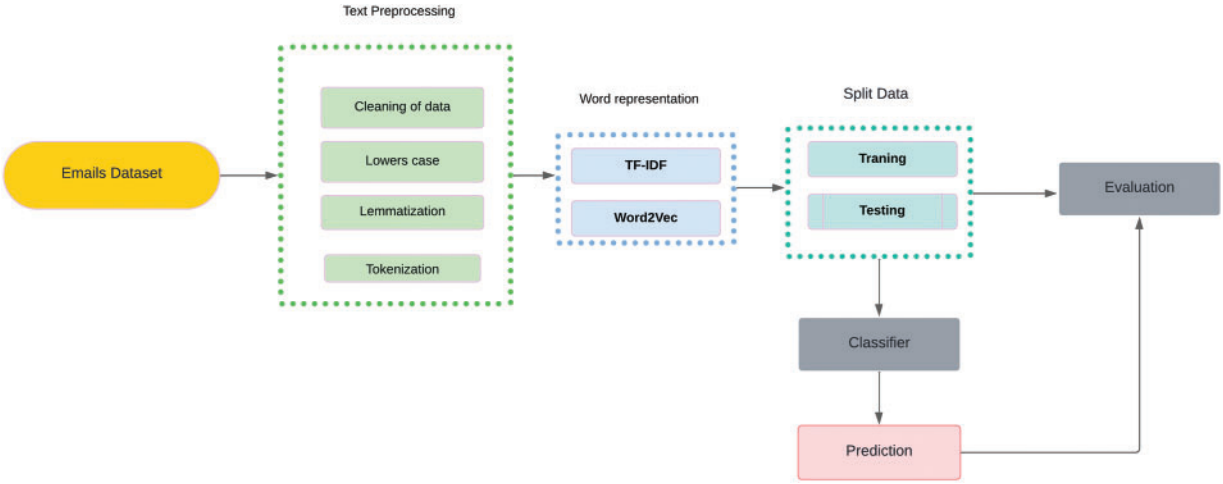


Figure 3: Proposed approach

4.2 Text Preprocessing

The text pre-processing phase consists of data cleansing, stop word removal, lemmatization, and tokenization.

- Cleaning of data. A collection of functions is established in this phase to purify the transcription data. These functions are employed to eliminate the majority of punctuation, special characters, numerals, URLs, stop words, and irrelevant text [32].
- It lowers the text. Each capital character is transformed into its corresponding minor character at this stage [32].
- Metaphor. In this stage, the morphological analysis is deemed to reduce the variability in the words, resulting in mapping words with similar meanings to the exact words [32]. One of the characteristics of lemmatization is its endeavor to determine the appropriate lemma based on the context.
- The tokenization process involves dividing text into smaller components, each referred to as a token. The token size most frequently encountered is a word [32].
- We employed the tokenizer integrated into BERT (BertTokenizer from 'bert-base-uncased') [33]. This tokenizer effectively manages out-of-vocabulary words, captures subtle nuances, and converts words into subword units, efficiently handling raw text. Its role in preserving intricate patterns found in phishing emails is crucial, as it instills confidence in the system's ability to handle complex data. This is essential for detecting fraud, enabling a more comprehensive text representation [34]. The model's capacity to detect sophisticated phishing attempts is improved by using BERT's tokenizer, which enhances its ability to capture contextual meanings.

4.3 Word Representation

During the word representation phase, text data is transformed into a vector representation to facilitate the algorithms' automatic comprehension of analogies and the generalization of the word.

Word representation models exhibited variations, ranging from classical to representation learning models [35]. Our research has implemented three predominant models for the classification of clinical text. These models are examined in the subsequent section.

4.3.1 The Term Frequency-Inverse Document Frequency (TF-IDF)

The TF-IDF was first introduced by [36] as a weighting factor for feature extraction. This numerical statistic is intended to demonstrate the importance of a word in a document within a corpus or collection. TF denotes the term's frequency of occurrence in a document, whereas IDF denotes the frequency of documents that contain a specific word.

4.3.2 Word2Vec

In 2013, the development of Word2Vec was initiated by Mikolov and his team at Google [37]. It employs a neural network architecture that adjusts its weights through backpropagation and gradient descent to transform words into vectors. The continuous skip-gram model and the continuous bag-of-words (CBOW) model are the two learning models used to generate word representations. The CBOW model predicts the target word using context words, whereas the continuous skip-gram model predicts the context of words using the target word [38]. Consequently, these models differ. To guarantee that words are embedded in vector space alongside related words according to their contextual meaning.

4.4 Classifier

The classifiers should be implemented after the completion of text preprocessing, word representation, and feature reduction. We evaluated four supervised classifiers, LR, RF, MLP, DT, and BERT, to ascertain which yielded the most favorable outcomes. 75% of the dataset was allocated to the training set, while 25% was dedicated to the testing set.

Bidirectional Encoder Representations Configuration

Utilizing its pre-trained weights for fine-tuning, we implemented the 'bert-base-uncased' variant of BERT. The model is composed of 12 layers, each of which contains 12 attention centers. In order to mitigate overfitting, we implemented the AdamW optimizer with a learning rate schedule during the training process. Using an early halting strategy to prevent overfitting, the model was trained over three epochs with a sample size 16. A 75%–25% divide between training and validation was established, and we reassured the model's efficiency by employing hardware accelerators (GPUs) to expedite the training process. In order to guarantee the comprehensive detection of phishing emails, we implemented hyperparameter optimization to enhance model performance.

5 Evaluation and Results

This section elucidates our findings by applying conventional machine learning techniques and LLM to the dataset and assessing the metrics above: precision, recall, accuracy, and F1-score. Subsequently, we evaluate the experimental configurations of each algorithm, which include the parameters and their respective values.

5.1 Experimental Setup

This section presents the experimental machine learning parameters for each algorithm used in this research to detect phishing attacks in phishing email datasets, as illustrated in Table 2.

Table 2: Machine learning parameters

Algorithm	Parameters	Value
Logistic regression	Solver	lbfgs
	Max iterations	100
Decision tree	Criterion	gini
	Max_depth	None
	Random _state	None
Random forest	Estimators	100
	Criterion	gini
	Max depth	None
	Random state	None
MLP	Optimize	adam
	H_L	100
	Activate function	relu
	Max iterations	300
	Random state	None
BERT	Pre-trained model	Bert-base-uncased
	Number of labels	2
	Tokenizer	Bert-base-uncased

Before machine learning algorithms can be applied to the dataset, they must be divided into testing and training. Models are developed according to these algorithms using the training dataset, while their efficacy is assessed using the testing dataset. This research ratio of training to examination is as follows: The assessment process processes the remaining portion of the dataset. In contrast, the training process utilizes 0.75 of the entire dataset.

5.2 Results and Discussion

The email fraud datasets were subjected to the application of the machine learning algorithms RF, DT, MLP, LR, and BERT to produce the results of this paper, which are presented in this section. We implemented these investigations in the Kaggle Notebook to address the binary classification assignment. We then compared the results using four evaluation metrics: precision, recall, F1-score, and accuracy.

Fig. 4 and Table 3 illustrate the outcomes of machine learning algorithms following the application of TF-IDF (Term Frequency-Inverse Document Frequency).

Fig. 5 and Table 4 illustrate the outcomes of machine learning algorithms following the application of W2V (Word two Vector).

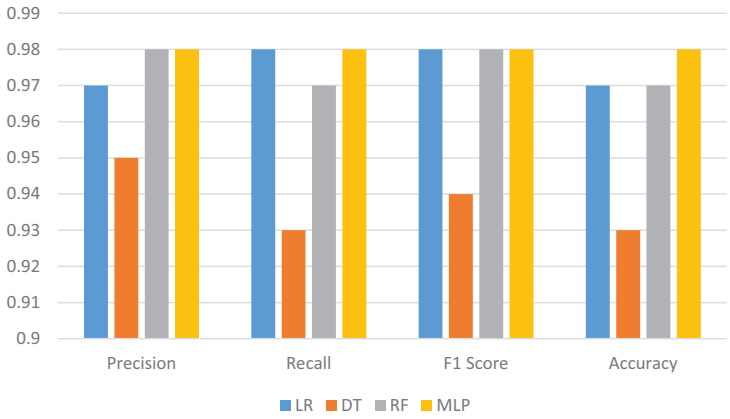


Figure 4: ML With TF-IDE

Table 3: Result ML with TF-IDE

	Precision	Recall	F1-score	Accuracy
LR	0.97	0.98	0.98	0.97
DT	0.95	0.93	0.94	0.93
RF	0.98	0.97	0.98	0.97
MLP	0.98	0.98	0.98	0.98

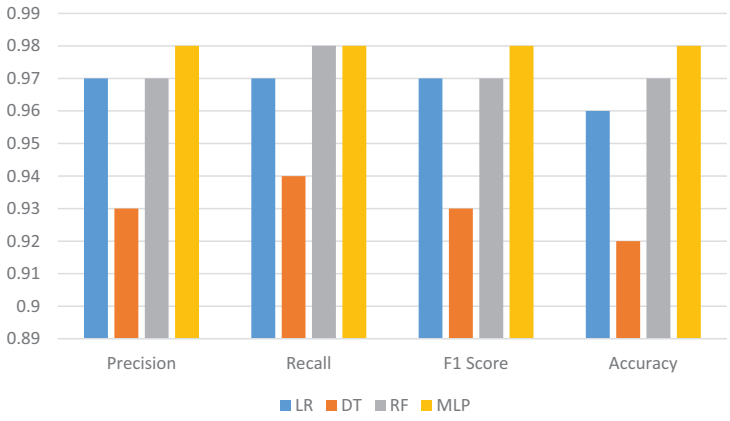


Figure 5: ML With W2V

Table 4: Result ML with W2V

	Precision	Recall	F1-score	Accuracy
LR	0.97	0.97	0.97	0.96
DT	0.93	0.94	0.93	0.92
RF	0.97	0.98	0.97	0.97
MLP	0.98	0.98	0.98	0.98

The results of the BERT large language model (LLM) are depicted in Fig. 6 and Table 5. The results demonstrate the efficacy of BERT in addressing text classification tasks, as evidenced by its performance on the provided dataset.

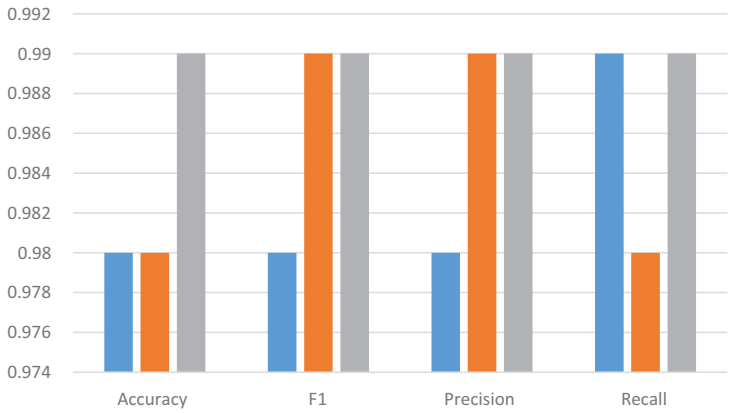


Figure 6: Result of BERT

Table 5: Result of BERT

Epoch	Accuracy	F1	Precision	Recall
1	0.98	0.98	0.98	0.99
2	0.98	0.99	0.99	0.98
3	0.99	0.99	0.99	0.99
Final	0.99	0.99	0.98	0.99

As Table 6 and Fig. 7, the training and evaluation periods for the various models used in our study were extracted using three distinct feature extraction methods: TF-IDF, Word2Vec (W2V), and BERT. What's intriguing is the variability in these durations. For instance, Logistic Regression (LR) demonstrated the quickest training and evaluation durations, with TF-IDF requiring approximately 10.94 s for training and 0.09 s for evaluation, outperforming all other methods. The training durations of the Decision Tree (DT) and Random Forest (RF) models were moderate, with the Random Forest (RF) model taking longer, especially in the case of TF-IDF and Word2Vec. The Multilayer Perceptron (MLP) model, however, required a greater amount of time, particularly for training, with TF-IDF needing approximately 107.58 s. It's worth noting that BERT, a Large Language Model (LLM), had the longest training and evaluation periods. The training process lasted approximately 1012.66 s, while the evaluation process took approximately 13.09 s. This variability in performance across models adds an intriguing dimension to our study.

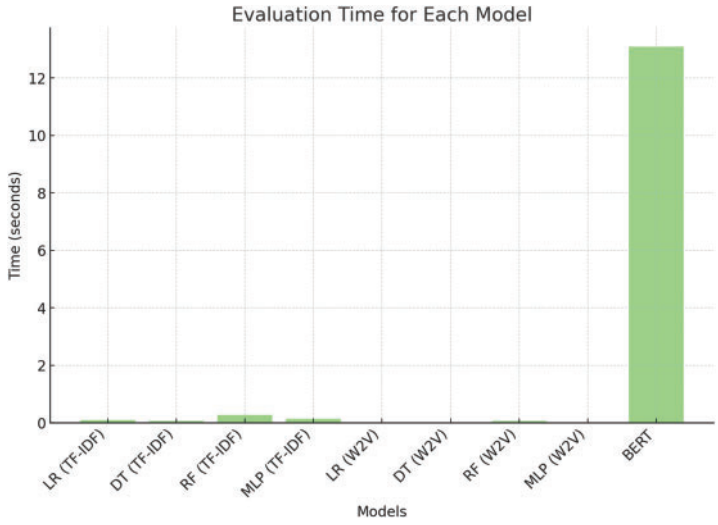
Table 6: Training and evaluation periods

	Model	Training time	Evaluation time
TF-IDF	LR	10.94446	0.085882902
	DT	30.78671	0.067041159
	RF	40.38932	0.266058207
	MLP	107.5756	0.1283288

(Continued)

Table 6 (continued)

	Model	Training time	Evaluation time
W2v	LR	0.254068	0.000999212
	DT	2.361782	0.00115037
	RF	12.97686	0.054642916
	MLP	34.43397	0.005062342
LLM	BERT	1012.66	13.09



(a) Evaltion time

(b) Traning time

Figure 7: Time

5.3 Discussion

Using models such as Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), and Multilayer Perceptron (MLP), we conducted a comprehensive evaluation of the performance of a variety of machine learning methods, including TF-IDF, Word2Vec (W2V), and BERT, in our experiment. Logistic Regression and Random Forest achieved an F1-score of 0.98 and an accuracy of 0.97 when using TF-IDF, while traditional methods such as Word2Vec and TF-IDF demonstrated strong performance. The MLP also demonstrated satisfactory performance with both TF-IDF and Word2Vec, achieving an F1-score and an accuracy of 0.98. Word2Vec's results, while marginally inferior to those of TF-IDF, still demonstrated robust performance, with F1-scores of 0.97 for Random Forest and 0.96 for Logistic Regression.

Nevertheless, BERT's sophisticated NLP capabilities considerably outperformed these conventional methods. BERT is particularly adept at interpreting the meaning of words through context analysis, in contrast to TF-IDF and Word2Vec, which frequently overlook nuanced phishing attempts as a result of their lack of contextual awareness. Due to its bidirectional approach, it is capable of identifying sophisticated terminology and subtle manipulation tactics that are frequently employed in phishing attacks. The findings of our investigation illustrated BERT's remarkable capacity to understand complex language patterns and long-term dependencies, rendering it particularly effective in the detection of email fraud. With BERT, we obtained an F1-score and accuracy of 0.99, which suggests that it is highly reliable and accurate in identifying fraudulent emails. BERT's potential as a potent instrument for email phishing detection is affirmed by this exhaustive analysis, which clearly demonstrates its superior performance in comparison to traditional machine learning models. Table 7 compares our project to others that are similar.

Table 7: Comparison between our work and previous works

Ref.	Algorithms	Accuracy
Approach (TF-IDF)	LR, DT, RF, MLP	0.98
Approach (W2V)	LR, DT, RF, MLP	0.98
Approach (BERT)	BERT	0.99
GRU, LSTM, BiGRU and BiLSTM [19]	GRU	0.9722
	LSTM	0.96
	BiGRU	0.973
	BiLSTM	0.972
Various ML and DL [19]	Best model: Various techniques	0.975
LR, NB, SVM [20]	SVM: Best model	0.943
	Naive Bayes	0.934
	Logistic regression	0.922
Various ML algorithms [21]	Best model: Various techniques	0.977

6 Conclusion

In summary, the comparison of machine learning methods, including TF-IDF, Word2Vec, and BERT, has provided us with a multitude of valuable information regarding the efficacy of these feature extraction strategies. The TF-IDF method was highly effective with all of the algorithms, achieving an F1-score of 0.98 and an accuracy of 0.97 for Random Forest and Logistic Regression. Word2Vec operated satisfactorily; nevertheless, it was inferior to TF-IDF in terms of efficiency. The Random Forest and Logistic Regression models have consistently demonstrated exceptional performance; however, their F1-scores and accuracy have experienced substantial declines. BERT enhanced the precision and dependability of text categorization tasks, rendering them more advantageous than conventional methodologies. During the third epoch, the BERT model exhibited its ability to manage complex language tasks with increased precision by achieving an F1-score and an accuracy of 0.99.It is recommended that future research focus on a number of critical areas in order to enhance the findings of this investigation. At the outset, the generalizability of the findings can be improved by incorporating a broader array of diverse and larger datasets. Furthermore, by incorporating additional sophisticated language models such as GPT-3 or T5, it is possible to develop a comparative understanding of BERT's performance. The quality and accuracy of classification can be enhanced by utilizing ensemble learning methods with language models that have already been trained. In conclusion, it is feasible to assess the model's ability to adapt to novel circumstances, and real-time fraud detection systems that employ these models may be implemented in the real world. In order to guarantee that models continue to function as phishing tactics evolve, it is essential to emphasize the importance of consistently revising and modifying them with new data.

In the future, the model's adaptability to new and evolving phishing threats could be improved by investigating multi-task and zero-shot learning to enhance BERT for phishing detection. This exciting research direction has the potential to inspire new ideas and approaches in the field of cybersecurity. Furthermore, the efficacy of BERT can be enhanced by tailoring it to the distinctive characteristics of fraudulent emails through domain-specific fine-tuning. In addition, future research could entail a comparative analysis of BERT and GPT-based models, which would examine their respective strengths in this field. Another promising direction is integrating BERT with other techniques, such as Graph Neural Networks (GNNs), to capture email metadata and network-based features. This hybrid approach can enhance the detection systems by offering a more thorough comprehension of fraudulent activities.

Acknowledgement: The authors would like to express their gratitude for the valuable feedback and suggestions provided by all the anonymous reviewers and the editorial team.

Funding Statement: This research was supported by Abu Dhabi University.

Author Contributions: Arar Al Tawil contributed to the conceptualization, methodology, software development, writing of the original draft, and project administration. Laiali Almazaydeh was responsible for data analysis and supervision. Doaa Qawasmeh contributed through investigation, providing resources, and reviewing and editing the manuscript. Baraah Qawasmeh handled validation and resource management. Mohammad Alshinwan contributed to visualization and investigation. Khaled Elleithy handled the final draft proofreading and supervision. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The dataset used in this study was obtained from the Kaggle website and is publicly available online.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] S. Villamil, C. Hernández, and G. Tarazona, "An overview of Internet of Things," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 18, no. 5, pp. 2320–2327, 2020. doi: 10.12928/telkomnika.v18i5.15911.

[2] B. B. Gupta and M. Quamara, "An overview of Internet of Things (IoT): Architectural aspects, challenges, and protocols," *Concurr. Comput.*, vol. 32, no. 21, 2020, Art. no. e4946. doi: 10.1002/cpe.4946.

[3] Y. Li and Q. Liu, "A comprehensive review study of cyber-attacks and cyber security; Emerging trends and recent developments," *Energy Rep.*, vol. 7, no. 8, pp. 8176–8186, 2021. doi: 10.1016/j.egyr.2021.08.126.

[4] K. T. Smith, L. M. Smith, M. Burger, and E. S. Boyle, "Cyber terrorism cases and stock market valuation effects," *Inf Comput. Secur.*, vol. 31, no. 4, pp. 385–403, 2023. doi: 10.1108/ICS-09-2022-0147.

[5] I. H. Sarker, "Deep cybersecurity: A comprehensive overview from neural network and deep learning perspective," *SN Comput. Sci.*, vol. 2, no. 3, p. 154, 2021. doi: 10.1007/s42979-021-00535-6.

[6] D. Bhamare, M. Zolanvari, A. Erbad, R. Jain, K. Khan and N. Meskin, "Cybersecurity for industrial control systems: A survey," *Comput. Secur.*, vol. 89, 2020, Art. no. 101677. doi: 10.1016/j.cose.2019.101677.

[7] H. Kavak, J. J. Padilla, D. Vernon-Bido, S. Y. Diallo, R. Gore and S. Shetty, "Simulation for cybersecurity: State of the art and future directions," *J. Cybersecur.*, vol. 7, no. 1, 2021, Art. no. tyab005. doi: 10.1093/cybsec/tyab005.

[8] D. Dasgupta, Z. Akhtar, and S. Sen, "Machine learning in cybersecurity: A comprehensive survey," *The J. Def. Model. Simul.*, vol. 19, no. 1, pp. 57–106, 2022. doi: 10.1177/1548512920951275.

[9] K. Shaukat *et al.*, "Performance comparison and current challenges of using machine learning techniques in cybersecurity," *Energies*, vol. 13, no. 10, 2020, Art. no. 2509. doi: 10.3390/en13102509.

[10] T. Berghout, M. Benbouzid, and S. M. Muyeen, "Machine learning for cybersecurity in smart grids: A comprehensive review-based study on methods, solutions, and prospects," *Int. J. Crit. Infrastruct. Prot.*, vol. 38, no. 19, 2022, Art. no. 100547. doi: 10.1016/j.ijcip.2022.100547.

[11] I. D. Aiyanyo, H. Samuel, and H. Lim, "A systematic review of defensive and offensive cybersecurity with machine learning," *Appl. Sci.*, vol. 10, no. 17, 2020, Art. no. 5811. doi: 10.3390/app10175811.

[12] M. Abutaha, M. Ababneh, K. Mahmoud, and S. A. H. Baddar, "URL phishing detection using machine learning techniques based on URLs lexical analysis," in *2021 12th Int. Conf. Inf. Commun. Syst. (ICICS)*, Valencia, Spain, IEEE, May 2021, pp. 147–152.

[13] S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nair, "A comparison of machine learning techniques for phishing detection," in *Proc. Anti-Phish. Work. Groups 2nd Annual eCrime Res. Summit.*, Oct. 2007, pp. 60–69.

[14] D. Samad and G. A. Gani, "Analyzing and predicting spear-phishing using machine learning methods," *Multidiszciplináris Tudományok*, vol. 10, no. 4, pp. 262–273, 2020. doi: 10.35925/j.multi.2020.4.30.

[15] V. Rathee and R. Mann, "Detection of phishing emails using machine learning, deep learning," *Int. J. Comput. Appl.*, vol. 183, no. 47, pp. 1–7, 2022. doi: 10.5120/ijca2022918687.

[16] N. A. Unnithan, N. B. Harikrishnan, S. Akarsh, R. Vinayakumar, and K. P. Soman, "Machine learning based phishing E-mail detection," in *Proc. 1st Anti-Phish. Shared Pilot 4th ACM Int. Workshop Secur. Priv. Analy. (IWSPA 2018)*, Tempe, AZ, USA, 2018, pp. 1–7.

[17] N. B. Harikrishnan, R. Vinayakumar, and K. Soman, "A machine learning approach towards phishing email detection CEN-Security@IWSPA 2018," in *Proc. 1st Anti-Phishing Shared Task Pilot 4th ACM IWSPA Co-Located 8th ACM Conf. Data Appl. Secur. Priv. (CODASPY 2018)*, Tempe, AZ, USA, Mar. 2018.

[18] S. Rawal, B. Rawal, A. Shaheen, and S. Malik, "Phishing detection in E-mails using machine learning," *Int. J. Appl. Inf. Syst.*, vol. 12, no. 7, pp. 21–24, Oct. 2017. doi: 10.5120/ijais2017451713.

[19] E. Benavides-Astudillo, W. Fuertes, S. Sanchez-Gordon, D. Nuñez-Agurto, and G. Rodríguez-Galán, "A phishing-attack-detection model using natural language processing and deep learning," *Appl. Sci.*, vol. 13, no. 9, Apr. 2023, Art. no. 5275. doi: 10.3390/app13095275.

[20] Phishing_Mail, "Kaggle," Accessed: Jul. 13, 2023. [Online]. Available: https://www.kaggle.com/datasets/somumourya/fishing-mail

[21] S. Sindhu, S. P. Patil, A. Sreevalsan, F. Rahman, and M. S. AN, "Phishing detection using random forest, SVM and neural network with backpropagation," in *Proc. Int. Conf. Smart Technol. Comput., Electr. Electron. (ICSTCEE)*, 2020, pp. 391–394.

[22] A. Pandey, N. Gill, K. S. P. Nadendla, and I. S. Thaseen, "Identification of phishing attack in websites using random forest-SVM hybrid model," in *Proc. 18th Int. Conf. Intell. Syst. Des. Appl. (ISDA 2018)*, Vellore, India, 2020, vol. 941, pp. 120–128.

[23] O. Kayode-Ajala, "Applying machine learning algorithms for detecting phishing websites: Applications of SVM, KNN, decision trees, and random forests," *Int. J. Inf. Cybersecur.*, vol. 6, no. 1, pp. 43–61, 2022.

[24] S. Al-Ahmadi, "PDMLP: Phishing detection using multilayer perceptron," *Int. J. Netw. Secur. Appl. (IJNSA)*, vol. 12, pp. 59–72, 2020.

[25] A. Odeh, I. Keshta, and E. Abdelfattah, "Efficient prediction of phishing websites using multilayer perceptron (MLP)," *J. Theor. Appl. Inf. Technol.*, vol. 98, no. 16, pp. 3353–3363, 2020.

[26] K. Joshi, M. Deshmukh, A. Kharat, and R. Kulkarni, "Performance evaluation of logistic regression and neural network for credit risk assessment," *Int. J. Eng. Adv. Technol. (IJEAT)*, vol. 9, no. 3, pp. 1245–1250, Feb. 2021. doi: 10.35940/ijeat.C6152.029320.

[27] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2019, *arXiv:1810.04805*.

[28] Y. Liu *et al.*, "RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*.

[29] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter," 2019, *arXiv:1910.01108*.

[30] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma and R. Soricut, "ALBERT: A lite BERT for self-supervised learning of language representations," 2020, *arXiv:1909.11942*.

[31] M. Hossin and M. N. Sulaiman, "A review on evaluation metrics for data classification evaluations," *Int. J. Data Min. Knowl. Manag. Process*, vol. 5, no. 2, pp. 1–11, 2015. doi: 10.5121/ijdkp.2015.5201.

[32] L. Almazaydeh, M. Abuhelaleh, A. Al Tawil, and K. Elleithy, "Clinical text classification with word representation features and machine learning algorithms," *Int. J. Online Biomed. Eng.*, vol. 19, no. 4, pp. 65–76, Apr. 2023. doi: 10.3991/ijoe.v19i04.36099.

[33] A. Nayak, H. Timmapathini, K. Ponnalagu, and V. Gopalan Venkoparao, "Domain adaptation challenges of BERT in tokenization and sub-word representations of out-of-vocabulary words," presented at the 1st Workshop Insights Negat. Results NLP, Nov. 2020, pp. 1–5.

[34] F. Trad and A. Chehab, "Prompt engineering or fine-tuning? A case study on phishing detection with large language models," *Mach. Learn. Knowl. Extract.*, vol. 6, no. 1, pp. 367–384, 2024. doi: 10.3390/make6010018.

[35] U. Naseem, I. Razzak, K. Khan, and M. Prasad, "A comprehensive survey on word representation models: From classical to state-of-the-art word representation language models," *Trans. Asian Low-Res. Lang. Inf. Process.*, vol. 20, no. 5, pp. 1–35, 2021. doi: 10.1145/3434237.

[36] K. S. Jones, "Document retrieval systems," in *A Statistical Interpretation of Term Specificity and Its Application in Retrieval*, London, UK: Taylor Graham Publishing, 1988, pp. 123–142.

[37] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proc. 1st Int. Conf. Learn. Represent. (ICLR 2013)*, Scottsdale, AZ, USA, 2013.

[38] W. S. Liu, Z. W. Cao, J. Wang, and X. Y. Wang, "Short text classification based on Wikipedia and Word2vec," in *Proc. 2nd IEEE Int. Conf. Comput. Commun. (ICCC)*, Chengdu, China, 2016, pp. 1195–1200. doi: 10.1109/CompComm.2016.7924894.