A Comprehensive Review of Machine and Deep Learning Approaches for Cyber Security Phishing Email Detection

Sarmad Rashed[*], Caner Ozcan[**]

[*]Department of Computer Engineering, Faculty of Engineering, Karabük University, Türkiye
Email: sermet.resit@gmail.com
https://orcid.org/0009-0009-3834-0522

[**]Department of Software Engineering, Faculty of Engineering, Karabük University, Türkiye
Email: canerozcan@karabuk.edu.tr
https://orcid.org/0000-0002-2854-4005

Abstract

Over the past fifteen years, phishing has emerged as the leading cybercriminal activity, resulting in the unauthorized acquisition of substantial financial resources amounting to billions of dollars. This phenomenon arises due to using novel (zero-day) and complicated tactics by phishing attackers to deceive internet users. Email is the primary approach utilized to initiate phishing attacks. This study comprehensively analyzes popular methods used in email spam tests. The present analysis comprehensively examines the key concepts, techniques, and research trends relative to spam filtering. The topic of discussion involved a general email spam filtering mechanism and the attempts of various scholars to counter spam by employing machine-learning methodologies. Our review examines the advantages and disadvantages of several machine learning methods within the context of spam filtering while addressing some of the biggest research inquiries in this domain.

Keywords- Spam filtering, Phishing email detection, Malware, Machine Learning, Deep learning.

I. INTRODUCTION

Emails have emerged as the primary, effective, and inexpensive means of academic communication, leading to an enormous rise in the overall quantity of emails sent. As a result, there has been an increased demand for more accurate spam filters. To provide optimal security and efficiency, an email filter must possess the capability to identify and classify phishing emails promptly in near real-time. Phishing is classified as a form of social engineering [49]. Phishing is a malicious activity that aims to fool individuals into revealing confidential and sensitive data, including personal identification and financial details. Phishing attacks sometimes involve emails with harmful attachments or URLs designed to reroute users. Hence, the identification of this potential danger presents a significant security obstacle. Phishing consists of various forms, such as bulk phishing, spear phishing, clone phishing, and whaling. Bulk phishing involves non-targeted attacks lacking customization for recipients. In contrast, spear-phishing specifically targets individuals or organizations. Clone-phishing entails sending a genuine email that is later modified with malicious content. Lastly, whaling focuses on senior-level employees as the primary targets [50]. Spam filters can be implemented at various levels, including email firewalls, mail transfer servers, and email servers. Anti-spam utilities offer network-level safeguards for email communication [52]. At the stage of the mail delivery agent, the implementation of spam filters is achievable. At the user's experience level, individuals can employ personalized spam filters that can automatically detect and eliminate such unwanted emails. Internet service providers, such as Yahoo and Google, have utilized machine learning techniques to implement email filtering systems. These providers also periodically update their blocklists to mitigate the risks posed by unsolicited bulk email (UBE) threats [53]. Phishing attacks involve the intentional transmission of malware through unauthorized messages by individuals seeking to exploit vulnerabilities in computer systems. Individuals become drawn towards fake websites by using fake URLs included in the content of unwanted messages. In this malicious action, fake URLs replicate widely recognized websites, diminishing their suspicion level. Several methodologies have been employed in detecting spam, involving tactics based on blocklists, heuristic techniques, metaheuristic approaches, and knowledge discovery procedures such as data mining and machine learning techniques [51].

II. LITERATURE REVIEW

Various scholarly inquiries concerning cybersecurity have been undertaken, including those related to intrusion detection and prevention. Email phishing is a confusing technique that involves illegally getting confidential and sensitive data from individuals and organizations. Researchers identified techniques for identifying fake email assaults. Detection of zero-day phishing attacks and phishing attacks, in general, was more accurate than previous approaches using actual phishing possibilities. According to three classifiers, NN had 95.18% accuracy, SVM 85.45%, and RF 78.89% in phishing recognition. In this study [1], the investigation detected unlawful activity. A dataset with 14 attributes was trained using three machine-learning architectures. This study [2] employs three machine-learning frameworks to identify fraudulent activity using a rule-based methodology. The models had been taught on a dataset with 14 attributes. K-nearest neighbors (KNN), Random Forest, and Support Vector Machines (SVM) are extensively used machine learning approaches.

Another study [3] describes the URL-based phishing detection algorithm PDGAN. An LSTM generated possible URLs, and a convolutional neural network (CNN) assessed phishing attacks to identify their maliciousness. A study [4] proposes a machine-learning phishing assault detection attack. Over 4,000 University of North Dakota email phishing emails were examined. After selecting the ten most critical hazards, a big dataset was utilized to build a model. Intelligent techniques such as Machine Learning (ML) and Deep Learning (DL) are the rising popularity of machine learning in cybersecurity. This may be attributed to its capacity to acquire knowledge from available data, enabling the extraction of pertinent information and anticipating forthcoming occurrences. In [5], it explores new phishing website detection techniques. Two datasets were analyzed for the strongest correlations. Domain-based and content-based URL-lexical functions were offered.

The researchers in the study [6] used deep learning to screen spam and fake emails. An optimized deep-learning system creates a 3-fold classifier that outperforms other cutting-edge investigations. A study [7] proposes a new feature selection method that blends scores from multiple current methodologies to remove character selection disparities and improve preprocessing reliability. In [8], researchers present neural network-based phishing categorization. Design risk minimization makes this detection approach accurate and generalizable. The primary aim of this research [9] is to examine many popular trendy spoofing prevention systems thoroughly. Researchers also present a high-level review of various phishing prevention tactics, such as detection, offensive defense, correction, and prevention, since they believe it's crucial to demonstrate how the different phishing detection methods fit into the greater scheme.

Using a mix of Ensemble Learning methods and hybrid features, the authors of the paper [10] propose a system named HELPFED for identifying phishing emails. By combining the content and linguistic properties of email communications, hybrid features accurately represent the messages. Using machine learning (ML) algorithms with several features, a study [11] detected phishing assaults using blacklist-based, online content-based, and heuristic-based tactics. In real-world circumstances, academic researchers provide empirical evidence to prove their accuracy and can quickly detect genuine websites and adjust to changing phishing methods. The authors [12] use LSTM to detect phishing in large email collections. The new technique requires sample growth and appropriate sample testing.

In [13], the authors propose a new phishing email classification method. The model analyzes email text using graph convolutional network (GCN) and natural language processing (NLP) methods. This approach [14] uses two screening modules before applying algorithms on web pages. The first module, preapproved site identification, uses the user's private allowlist to determine website safety. The second module, Login Form Finder, verifies website validity by checking for login forms. In [15], authors say most email classification approaches employ supervised learning algorithms, which require lots of labeled data for training. The study [16] proposes a novel framework for categorizing phishing emails that integrates deep learning techniques, a graph convolutional network (GCN), and natural language processing. These methodologies are employed to examine the content of an email and identify potential signs of harmful intentions.

The authors [17] present a complete study of the most recent machine-learning techniques for identifying and filtering spam emails. As part of their research, the authors examine the evolution of spam filtering, the most recent developments, and the subject's future. Another study [18] designed a real-time phishing detection system using neural networks and reinforcement learning. Reinforcement learning improves the system's performance by adapting to changing phishing email detection patterns and accounting for current behavior changes. In this proposal [19], researchers have devised an innovative method to improve the efficiency of the Naive Bayes Spam Filter in identifying textual modifications and accurately classifying emails as either spam or legitimate (ham). The authors propose an understanding-and-use-based quality categorization approach in the study [20]; the authors provide "PhishBench," a reliable experimental platform for phishing assault detection.

By reviewing all the state-of-the-art studies, we found that most of the studies are limited to extracting the features from the subsection of the email, like from the header, email body, or attached URLs. Numerous studies tried different techniques to extract the features for phishing detection. However, these studies also extract the features differently but from the same subsection of email. There are few studies [14,16] that tried to extract the features from different subsections of email. However, these studies have several issues, such as low robustness or high error rates. Collectively, there is a need to consider all the subsections of the email for phishing detection and improve the performance of the detection model.

III. DATASET

Various benchmark datasets are widely employed for email phishing detection. However, these datasets might not fulfill all the specific requirements, such as complete details of headers, bodies, and attachments for many samples.

3.1 Enron Email Dataset

The Enron-Spam dataset is a fantastic resource collected by V. Metsis, I. Androutsopoulos, and G. Paliouras [25]. The dataset consists of 33,716 email messages, with 17,171 classified as spam and 16,545 classified as non-spam, sometimes known as "ham". The ideal source for authentic emails is the Spam Assassin website, renowned for its role in data mining competitions. The original dataset is organized so that each email is stored in a separate text file spread over many folders. The dataset must be converted into a single CSV file for further processing. The Python script will convert the selected dataset into a format that best fits the model for phishing detection. The dataset contains the following columns, as shown in Table 1.

TABLE 1. CONTENTS OF ENRON EMAIL DATASET.

Column	Explanation
Subject	The e-mail's subject line
Message	The body of the dialog. This section may be left empty if the message contains a subject line without any accompanying body. Including the initial email subject line, "from:", "to:", and other relevant information is common practice in the case of misdirected emails or responses.
Spam/Ham	Includes the parameters "spam" and "ham". Whether or not the message was identified as spam.
Date	The date the e-mail arrived. Has a YYYY-MM-DD format.

3.2 Phishing Email Dataset (PhishTank)

PhishTank is a community-driven initiative that collects and shares information about phishing websites [26]. They provide a dataset containing URLs and associated metadata, including the email content used in phishing attacks. This is relevant for training machine learning techniques in the context of phishing detection. After searching for all the well-known datasets from the literature, we summarize them in Table 2.

TABLE 2. PUBLICLY AVAILABLE EMAIL DATASET.

Dataset Name	Number of Massages		Year	Challenges	References
	Spam	non-spam			
Spam email	1378	2949	2010	Email Variability	Csmininggroup [27]
Hunter	928	810	2008	limited amount of email	Gao et al [28]
Trec 2007	50,199	25,220	2007	Multilingual Content	Debarr and Wechsler [29]
Dredze image	3297	2021	2007	Scalability	Dredze, gevaryahu, and Elias-bachrach [30]
Zh1	1205	428	2004	limited amount of email	Zhang et al [31]
Trec 2005	52,790	39,399	2005	Outdated Content	Androutsopoulos et al. [32]
Enron-spam	20,170	16,545	2006	Data Variance, Lack of Diversity	Koprinska et al [25]

Table 3 below explains the literature study, used datasets, used machine learning methods, and the gap of each dataset in detail.

TABLE 3. DATASETS OVERVIEW IN THE LITERATURE.

Study	Datasets	Machine Learning Methods	Gap
Yadav et al. (2019) [53]	Enron Email Dataset, Phishing Email Dataset (PhishTank)	Deep Learning (LSTM, CNN)	Lack of comparison with traditional machine learning algorithms
Chandrasekaran et al. (2020) [54]	Enron Email Dataset, Phishing Email Dataset (PhishTank)	Random Forest, Support Vector Machines (SVM)	Limited exploration of deep learning techniques and feature engineering
Alshahrani et al. (2021) [55]	Enron Email Dataset, Phishing Email Dataset (PhishTank)	Ensemble Learning (Random Forest, XGBoost)	Lack of comparison with other ensemble methods
Islam et al. (2022) [56]	Enron Email Dataset, Phishing Email Dataset (PhishTank)	Deep Learning (BERT)	Limited exploration of other deep learning architectures
Nguyen et al. (2022) [57]	Enron Email Dataset, Phishing Email Dataset (PhishTank)	Gradient Boosting (XGBoost, LightGBM)	Limited evaluation of different datasets and feature extraction techniques
Dey et al. (2016) [58]	Enron Email Dataset	Naive Bayes, Decision Trees	Limited feature representation
Al Marzouqi et al. (2018) [59]	Phishing Email Dataset (PhishTank)	Random Forest, Logistic Regression	Lack of diversity in datasets and classifiers
Salah et al. (2019) [60]	Enron Email Dataset	Support Vector Machines (SVM)	Limited evaluation of different classifiers
Gharib et al. (2020) [61]	Phishing Email Dataset (PhishTank)	Deep Learning (Convolutional Neural Networks)	Lack of comparison with traditional machine learning algorithms
Li et al. (2021) [62]	Enron Email Dataset, Phishing Email Dataset (PhishTank)	Ensemble Learning (Random Forest, XGBoost)	Lack of focus on feature selection and extraction

Table 4 below explains in detail the summary and results of studies using machine learning techniques for phishing email classification (literature study, used machine learning architecture, used datasets, and the results of each dataset).

TABLE 4. GENERAL SUMMARY AND RESULTS OF STUDIES USING MACHINE LEARNING TECHNIQUES FOR PHISHING EMAIL CLASSIFICATION.

Author (Year)	Machine Learning Architecture	Datasets	Results
(Saad et al., 2022)	Random Forest, Support Vector Machines (SVM)	Enron Email Dataset, Phishing Email Dataset (PhishTank)	Accuracy: 97.58%, Precision: 98.02%
(Sakkis et al., 2001)	Ensemble Learning (Random Forest, XGBoost)	Enron Email Dataset, Phishing Email Dataset (PhishTank)	Classification of holdout stacking per usage: SR: 91.7%, SP: 96.5%, TCR: 8.44 Classification of cross-validation stacking per usage: SR: 89.6%, SP: 98.7%, TCR: 8.6.
(Yaseen et al., 2021)	Deep Learning (BERT)	Enron Email Dataset, Phishing Email Dataset (PhishTank)	Accuracy: 98.67%, F1 score: 98.66%
(Sankhwar et al., 2019)	Gradient Boosting (XGBoost, LightGBM)	Enron Email Dataset, Phishing Email Dataset (PhishTank)	Accuracy: 93.01%, Precision: 91.26%
(Palanichamy et al., 2023)	Naive Bayes, Decision Trees	Enron Email Dataset	Utilising TF-IDF achieved an Accuracy of 87.5%.
(Abdulraheem et al., 2022)	Random Forest, Logistic Regression	Phishing Email Dataset (PhishTank)	Using (PCA+LMT) achieved Accuracy: 96.92%, Recall: 96.9%.
Salah et al. (2019)	Support Vector Machines (SVM)	Enron Email Dataset	Using ANN (100,100) Relu function: Pd: 90.3%, Pfa: 1.5%, Pmd: 9.7%, Accuracy: 94.5%. Using SVM Gaussian Radial basis function: Pd: 82.3%, Pfa: 27.7%, Pmd: 17.7%, Accuracy: 77.3%. Using LR regularization parameter=0.7: Pd: 87.1%, Pfa: 1.4%, Pmd: 12.9%, Accuracy: 92.9%.
(Alshingiti, et al., 2023)	Deep Learning (Convolutional Neural Networks)	Phishing Email Dataset (PhishTank)	Accuracy of using Three distinct DL techniques: CNN: 99.2%, LSTM–CNN: 97.6%, LSTM: 96.8%.
Li et al. (2019)	Ensemble Learning (Random Forest, XGBoost)	Enron Email Dataset, Phishing Email Dataset (PhishTank)	ROC: 93.2%, AUC: 95.3%
Nguyen et al. (2022)	Enron Email Dataset, Phishing Email Dataset (PhishTank)	Gradient Boosting (XGBoost, LightGBM)	Recall: 91.7%, Precision: 93.3%

IV. METHODS

4.1 Naïve Bayes Classifier

Based on Bayes' Theorem [24], it is a classification technique that employs the idea of predictor independence. A Naive Bayes classifier operates on the premise that the presence of one feature in a class is independent of the presence of other features for the sake of simplicity. Implementing the Naive Bayes model needs less data and is advantageous when dealing with enormous datasets. Because of its simplicity and reputation for outperforming more sophisticated methods, Naive Bayes is a commonly used classification technique. The Naive Bayes classifier is an uncomplicated statistical classifier that applies Bayes's theorem and assumes independence and robustness. The name "autonomous attribute model" is a more suitable designation for the probability model [33,34].

$$\text{Bayes Theorem: Prob (B given A)= Prob (A and B)/Prob (A)} \quad (1)$$

Because of this apparent benefit, it is extensively utilized in spam filtration (to detect unsolicited emails) and sentiment mapping (to identify positive and negative consumer sentiments in social media studies). The application of the Naive Bayes classifier that is most commonly recognized is spam filtration. Currently, the majority of messaging programs employ Bayesian spam filtering techniques. Although consumers can install email-filtering apps, server-side email filters that utilize Bayesian spam filtering techniques are included in the software that facilitates email services for optimal functionality [35].

4.2 Support Vector Machine Classifier

Support Vector Machines (SVM) is a collection of techniques developed by [36] to address classification and regression tasks. Support Vector Machines (SVM) Utilize scientifically proven supervised learning methodologies that have demonstrated superior performance to other contemporaneous learning approaches [38]. While the SVM may not exhibit the same level of speed as other classification methods, its primary advantage resides in its superior accuracy. This is attributed to the algorithm's capability to effectively model complex multidimensional parameters that are neither sequential nor simplistic. Support Vector Machines (SVM) exhibit limited vulnerability to situations when the model is intricate and characterized by a high parameter count relative to the dataset size. The previously mentioned attributes render Support Vector Machines (SVM) the most suitable method for application in domains including digital handwriting recognition, text categorization, and voice recognition, among other areas.

4.3 *Ensemble* Classifiers

Ensemble learning is an emerging methodology that involves training and combining many classifiers to enhance the overall classification accuracy of a system when applied to a specific task, such as spam filtering. Ensemble learning algorithms are a classification methodology that integrates multiple machine learning algorithms to enhance the overall accuracy of the system's classification performance. In [43], the authors proposed the integration of many filters as a compelling strategy for efficiently addressing the proliferation of spam, which now manifests in diverse manifestations. Bagging and boosting are widely acknowledged as prominent ensemble classifiers [44]. The algorithms employ different subsets of the complete data set to train classification instances. Bagging refers to combining predictions given by many trained classifiers using subsets of data sampled from a larger dataset. Random forests represent a notable application of the ensemble approach, wherein multiple decision trees and various models are created to address the same difficulty, and their outcomes are collected to ascertain the most suitable classification conclusion overall [45]. Bagging, also known as the collection of bootstrapping, is a meta-learning approach that is frequently employed in decision tree systems. The random forest algorithm has become recognized as an ensemble technique for decision trees and is renowned for its ability to attain high levels of classification accuracy. Boosting is a productive strategy that amalgamates multiple "weak" trainees into a singular learner with greater potency than the standalone learners [46]. The AdaBoost system is a notable demonstration of the boosting learning method, founded on the principle of amalgamating several weak hypotheses. The objective of boosting is to provide a classification rule that exhibits high precision by aggregating several weak processes or poor estimations, each of which may possess only moderate accuracy. The student is systematically instructed through each stage of the categorization technique, and the outcomes of each activity are utilized to enhance the dependability of the data in the following phases [47].

4.3.1 Decision Tree

A Decision Tree (DT) is a type of classifier characterized by its hierarchical structure, which closely mimics a tree's. Decision tree induction is a distinct technique that aids in developing categorization information, as stated by references [38]. In a decision tree, every node can be classified as a leaf or non-leaf node. A leaf node represents the outcome or value of the goal feature, also known as the class. Additionally, the decision node has the potential to serve as an indicator for a test that will be done to evaluate the significance of a certain attribute. In the present scenario, each potential test result is shown through a subsection and a cluster of trees, representing a subset of the overall tree structure. Utilizing a decision tree is a viable approach for resolving classification problems. This involves starting at the tree's root and progressing towards the leaf node, ultimately yielding the classification outcome. The utilization of the decision tree learning technique has been observed in the

Context of spam filtering. This study aims to construct a decision tree model and subsequently train it to make predictions on the value of a target variable using a given set of input variables. Each leaf in the decision tree corresponds to a specific value of the desired variable, determined by the input variables along the path from the root to that leaf. Insight into a tree can be obtained by breaking down the fundamental set according to the value of the given characteristic. The above process is iteratively performed on every resulting subset, providing a potential rationale for its designation as recursive partitioning. The recursion process terminates when all subsets associated with a particular node possess target variables that can be compared.

4.3.2 Random Forests

The Random Forest (RF) algorithm is a widely recognized ensemble learning strategy and regression technique employed to categorize data into discrete groups [39]. It is presented for the first time by [40]. The system utilizes decision trees to make predictions. During the training phase, the programmer generates decision trees. Subsequently, the decision trees are used to estimate the group by considering the chosen groups from each tree and determining the group that receives the highest number of votes. The RF approach is increasingly being recognized and utilized across various academic areas. Within the realm of literature, it is frequently employed to address analogous issues, as evidenced by [25,41,42]. Random forests are more advantageous than decision trees for minimizing classification error and maximizing f-scores. Furthermore, the performance of the method above is generally on par with or surpasses that of support vector machines (SVMs), considering that SVMs are considerably easier for humans to understand. The model performs well when dealing with diverse data sets that exhibit missing variables. This approach offers a reliable means of quantifying missing data and maintaining data integrity when a significant amount is unavailable. Random Forest (RF) allows the user to construct the highest possible quantity of trees.

4.3.3 AdaBoost

AdaBoost, introduced in 1995 by [63], uses weak learners, such as small decision trees, using iteratively modified data to slightly enhance random guessing. Individual forecasts are mixed using a weighted average voting procedure to determine the final estimates. Each boosting iteration involves weight assignment $\omega 1, \omega 2, \omega 3, \ldots, \omega N$ for every training sample. First, a weak learner is trained on the original dataset with uniform weights $\omega i = 1/N$, adjusting sample weights after each iteration, and the learning algorithm is reapplied to the updated dataset. A phase in the boosted algorithm from the previous step raises the weights of erroneously estimated training data and decreases the weights of well-predicted training samples. With more repetitions, uncertain cases grow more prominent. Thus, every academically challenged student feels obligated to focus on the examples that previous generations ignored.

4.3.4 Gradient Boosting Machines (GBM)

Merging weak learners' shallow decision tree predictions into a robust model. Many short decision trees construct adaptive predictive models by aggregating weak learners' predictions. Many experts appreciate this technology's predictive modeling accuracy and versatility. GBMs start with a variable's median or average value for regression analysis or a category breakdown for classification tasks. It collects difficult pupils. Each iteration adds weak decision tree learners to the ensemble and teaches them to fix prior faults. We chose data segments miscategorized or had greater prediction errors in previous cycles. Newly trained weak learners anticipate loss functions. It finds leftover data mistakes. Lower learning rates lower risks but may need more iterations to converge. Higher learning rates may accelerate convergence but surpass the optimal solution. Weaker learners' estimates are added to the ensemble. By integrating the estimations of all weak learners, weighted by learning rate, the Gradient Boosting Machine (GBM) model produces the final forecast. Gradient Boosting Machines (GBMs) avoid overfitting via tree removal and maximum tree depth. GBM models may predict new data after training.

4.3.5 Stochastic Gradient Boosting (XGBoost)

XGBoost is known for its robustness against overfitting, automatic missing value management, and complex data linkages. This method is widely used in finance, healthcare, natural language processing, and others due to its excellent prediction abilities and ease of implementation. Formulating an objective function for optimization begins the process. The objective function has two parts: a loss function and a regularization term. The loss function measures the variation between goal values and model predictions. Variation is assessed using average squared error for regression and log loss for classification. Combining L1 (Lasso) and L2 (Ridge) regularization terms penalizes advanced models to reduce overfitting. The gradient of the objective function concerning the model's predictions is computed at each iteration to identify the mistake's size and direction. The model may prioritize data points with large prediction errors after training the weak learner (tree) to estimate the negative gradient. Unique to XGBoost, the CART (Classification and Regression Trees) feature solves classification and regression issues, including multiclass classification. XGBoost also controls model complexity via L1 and L2 regularization terms in the objective function. It also prunes decision trees to remove areas that reduce loss. Individual tree projections are generally summed to get the final forecast, with the learning rate defining their value.

4.4 Deep Learning Classifier

4.4.1 Convolutional Neural Networks (CNNs)

According to [64], Convolutional Neural Networks (CNNs) are made to extract characteristics and find patterns in grid-like data, especially image data. A hierarchical architecture with convolutional and pooling layers allows independent training and extraction of more complex data properties. CNNs process input data using convolutional layers and kernels, programmable algorithms. Additionally, filters traverse input to find and evaluate local patterns or characteristics. The early layer filtration commonly detects and extracts edges, corners, and textures. The filtration can capture increasingly abstract and complicated properties as we learn more about network topologies. During training, convolutional layers can acquire and extract these features separately from input data. Max-pooling, which retains the greatest value in a specific region of the feature map, is the most frequent pooling method. Pooling reduces computational complexity, overfitting, and translation-invariant characteristics by focusing on significant information. Common convolutional neural network (CNN) patterns in computer vision include LeNet-5, AlexNet, VGGNet, GoogLeNet (called Inception), and ResNet. The depth, filters, and layer structure of these models separate them. The fully linked layers recognize complex linkages between convolution and pooling layer data. CNNs use data labels for supervised learning, and optimizing model parameters like weights and biases minimizes a loss function. The loss function comes in two forms: cross-entropy for classification and mean squared error for regression. The pre-trained networks' characteristics can be used in new datasets or tasks with little information. Several studies have used CNNs, including [65,66].

4.4.2 Recurrent Neural Networks (RNNs)

RNNs process and predict sequential data based on the sequence and context of the input forms. Because they can capture temporal correlations and sequential patterns, recurrent neural networks (RNNs) are commonly used in machine learning. Recurrent connections distinguish recurrent neural networks (RNNs). These connections produce a loop that allows memory preservation by transmitting information from one time step to the next. This secret state efficiently stores previous time step data, and each time step updates the hidden state using the current input and prior hidden state. Each time step (t), a recurrent neural network (RNN) merges an input vector (such as a sentence word) with the previous hidden state to create a new hidden state. An activation function like the hyperbolic tangent (tanh) or corrected linear unit controls this approach. Sharing weights and biases across all temporal stages allows Recurrent Neural Networks (RNNs) to gather information and apply consistent changes across a sequence. Predicting the next word in a sentence is possible using the model, and it can also classify sequences, such as text sentiment. Additionally, the model may sequence written or audio outputs; these neural networks are intended to handle sequential input. Using recurrent connections and hidden states, these models can identify temporal relationships and patterns in sequential data, making them useful for machine-learning applications that rely on sequences. Many academics [65, 67, 68, 66] have trained and tested fake email detection models using recurrent neural networks (RNNs).

4.4.3 Bidirectional Encoder Representations from Transformers (BERT)

BERT uses Vaswani et al.'s transformer design from this study [69]. A transformer architecture uses a self-attention mechanism to handle sequential data simultaneously, making it an efficient NLP solution. Word order is considered by conventional language models like Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks. Also, improving bidirectional context comprehension is essential for comprehending statement significance. BERT is trained on a large corpus of web-sourced text to recognize missing sentence words. It distinguishes "real" statements from selected ones. This strategy helps BERT learn language, structure, terminology, and global knowledge. BERT contextualizes input tokens, and the representations operate dynamically, capturing token content within the phrase's context. The transformer-architecture-based BERT model pre-trains on a large text corpus to understand language and context. Chatbots, sentiment analysis, document classification, and language translation use the model, which performs well on many natural language processing (NLP) tasks. BERT's exceptional performance has prompted the NLP community to customize it for certain needs.

4.4.4 Deep Neural Networks (DNNs)

Deep Neural Networks (DNNs) use several connected layers to detect and represent data patterns [70] independently. These algorithms work well in image, natural language, and speech recognition. Input, embedding, and output layers make up the neural network architecture, and each layer of the mind has linked nodes. This is called feedforward propagation. All nodes in each layer receive weighted data from neurons in the preceding layer. Weighted inputs are collected and supplied as a function of activation. Supervised learning modifies Deep Neural Network (DNN) weights and biases using labeled examples as training data. Deep neural network evolution relies on the backpropagation technique. This technique optimizes the weights after computing the loss function's gradient about the model's variables. DNN depth depends on its hidden layers. In addition, deeper networks may be more conceptually complex and sophisticated. Utilizing such networks may require more information and processing resources. A DNN's capacity depends on the number of nodes in each layer. Growing the model can help it represent complicated linkages and patterns. Overfitting occurs if the model gets excessively suited to learning data and operates badly on unknown data. Following its training, a Deep Neural Network (DNN) may predict new data. In conclusion, neural networks are extensively used across disciplines because they automatically recognize characteristics, execute complicated tasks, and understand complex data patterns. They process visuals, text, and voice well because of this.

4.4.5 Gated Recurrent Unit (GRU) Networks

GRUs, like RNNs, use recurrent connections to transfer information between periodic steps [71]. These interactions create a covered state (memory) that stores information from previous phases. A permissions mechanism has been implemented for network information delivery, and this mechanism has two gates. The first is the update gate (z_t), which guides the relationship between keeping the prior hiding state and allowing new data from the present time step. The second one is the reset gate (r_t), which determines which elements of the earlier hidden state and current input should be merged to form a hidden state. Sigmoid activation functions compute the update gate (z_t) and reset gate (r_t), and return values range from 0 to 1. These gateways determine information flow: The update gate determines which parts of the earlier hidden state (h_(t-1)) should be retained and which ones should be updated to match the potential hidden state. The reset gate selects components of the prior hidden state (h_(t-1)) to exclude and any possible hidden state to replace. The hyperbolic tangent activation function may calculate the potential hidden state (h_tilde_t). After each iteration, the update gate adds the previously calculated hidden state (h_(t-1)) to the potential hidden state (h_tilde_t) to calculate the final hidden state (h_t). For training these methods, backpropagation through time (BPTT) is used, and every stage calculates weight and bias variations within the network and combines them across the sequence. Gradient descent uses slope to alter model variables. GRU networks address the constraints of ordinary RNNs. To capture persistent linkages in structured data, filtering approaches control network information flow.

V. Evaluation Measures

In general, evaluating spam filters involves utilizing publicly accessible large databases that consist of both legitimate (ham) and unauthorized (spam) messages. Classification accuracy (Acc) is a commonly employed performance statistic. The metric used to evaluate the filter's effectiveness is the relative number of correctly classified messages, which is determined by calculating the percentage of messages that are categorized correctly. Nevertheless, there has been a suggestion that relying solely on the Accuracy metric for performance measurement is deemed inappropriate. In addition to utilizing false positives and negatives in decision theory, it is crucial to include supplementary performance measures, such as recall and precision, which are widely applied in information retrieval. The significance of this matter is underscored by the financial implications linked to misclassification. When a spam message is erroneously classified as spam, the user has a low level of discomfort, as the only necessary action is to delete the message in the issue. Conversely, misclassifying a non-spam communication as spam can be vexing since it may lead to the accidental loss of vital information caused by the erroneous categorization of a filtering system. Equation (2) represents the formula for calculating classification accuracy:

$$\text{Accuracy} = (|TP| + |TN|) / (|TP| + |TN| + |FP| + |FN|) \quad (2)$$

The metrics of recall and precision, which have been acquired from the discipline of information retrieval, are employed to assess the usefulness and quality of spam filters [21], respectively. The accuracy meter, a quantitative indicator of the effectiveness and reliability of the spam filter, is determined by dividing the number of valid emails (ham) wrongly classified as spam by the overall amount of email messages [22,23]. Equation (3) represents spam precision:

$$\text{Precision} = (|TP|)/(|TP|+|FP|) \quad (3)$$

Recall, also known as effectiveness, describes the proportionate quantity of unauthorized messages the filter intercepted and prevented from infiltrating the mailbox. Equation (4) represents spam recall:

$$\text{Recall} = (|TP|)/(|TP|+|FN|) \quad (4)$$

The f1-score, often known as the F1 or F- measure, is a metric commonly used to assess the efficacy of a filter. The metric being evaluated measures the accuracy of a single measurement and is formally defined as the weighted harmonic average of precision and recall computed inside a unified equation. The F-measure utilizes a parameter that allows for the selection between recall and precision. Equation (5) represents the F1 measure:

$$\text{f1-score} = (2*\text{Precision}*\text{Recall})/(\text{Precision} + \text{Recall}) \quad (5)$$

VI. Discussion

Machine learning (ML) and deep learning (DL) have recently been recognized as effective anti-spam strategies. Email spam threatens privacy, security, and the user experience. Due to rule-based email filters' inability to combat developing spam methods, machine learning (ML) and deep learning (DL) technologies are needed. Such spam recognition methods have shown encouraging results in recent studies. Zhang et al. [72] tested a convolutional neural network (CNN) to identify and remove spam emails. The researchers used various categorized emails for modeling and assessment, including spam and legal ones. Their study showed that their Convolutional Neural Network (CNN) model beat rule-based and heuristic techniques in email spam categorization with high accuracy. Email spam detection systems using machine learning and deep learning algorithms are a breakthrough. The capacity of these algorithms to adapt to new spam methods makes them more flexible and robust than rule-based filters. People evaluate emails' contents, sender, attachments, and structure to make informed decisions. In addition, machine learning and deep

Learning models can detect fake emails that trick consumers into sharing private information. Email interactions are safer when these models identify spoofing movements. In recent years, ML and DL algorithms have improved email spam testing. Key advancements include:

- Text Classification: CNNs and RNNs are used to categorize email text, acquire contextual data, and analyze it efficiently.
- Ensemble Techniques: Multiple research projects have examined ensemble methods that use ML and DL models to improve spam detection.
- Defense Against Adversaries: Research has focused on techniques to resist spam filter-tricking campaigns.
- Real-Time Analysis: ML and DL models analyze emails in real-time to identify and quarantine spam.

Regardless of the notable progress achieved, the domain of email spam filtering still challenges several problems and unsolved matters. The next part outlines the ongoing research concerns related to email spam filtering:

- Improved image spam filtering; most spam filters detect only text spam. Many adept spammers hide their emails in stage images to avoid screening systems.
- Deep learning is necessary for spam filtering since it provides computational depth and hierarchical structures for data representation.
- Dynamic feature space filters' ineffectiveness. Many spam filters cannot add or remove features without rebuilding the model to keep up with email spam filtering advances.
- Due to the lack of a spam filter security approach, this attack might be destructive or exploratory, targeting specified targets or without discrimination.
- Improved image spam filtering. Spam filters typically identify text-based spam.

Machine learning (ML) and deep learning (DL) algorithms have improved email spam evaluation efficiency and flexibility. However, data privacy, adversarial assaults, and model generalization remain issues in the area. More research is needed to improve spam filters. This may be done via privacy-preserving methods, multimodal identification, autonomous behavioral analysis, and cross-domain transfer learning. Interdisciplinary collaboration and creative ideas are also needed to overcome environmental issues.

VII. CONCLUSION

The present study investigated several machine learning approaches and their implementation in the context of spam filtering. This study comprehensively assesses the currently employed algorithms for categorizing emails into spam or non-spam categories. It

examines the endeavors of many scholars in addressing the problem of spam through the utilization of machine learning classifiers. It aimed to explore the evolution of spam communications over time, specifically focusing on the strategies employed to bypass filtering mechanisms. The analysis focused on the underlying architecture of email filtering systems and the methodologies used to identify and detect spam emails. The study examined a selection of publicly accessible datasets and performance indicators that can be employed to assess the efficacy of spam filters. The research emphasized the difficulties encountered by machine-learning approaches in effectively addressing the spam problem and conducted a comparative analysis of several machine-learning methodologies. Furthermore, we have presented certain outstanding matters about research on spam filters. Overall, the quantity and size of the published work we analyzed suggest that significant advancements have been made and will continue to be made in this field of study. There are an extensive number of different possibilities for further research and exploration. The phishing datasets may potentially acquire further characteristics. One potential future upgrade is the implementation of real-time retraining and evaluation capabilities. Furthermore, it would be of great interest if additional studies in the domain of malicious email detection were to explore sophisticated approaches such as Reinforcement Learning and Hidden Markov Model-based techniques.

REFERENCES

[1] Mohamed, G., Visumathi, J., Mahdal, M., Anand, J., & Elangovan, M. (2022). An Effective and Secure Mechanism for Phishing Attacks Using a Machine Learning Approach. Processes, 10(7), 1356. https://doi.org/10.3390/pr10071356.

[2] Ojewumi, T. O., Ogunleye, G. O., Oguntunde, B. O., Folorunsho, O., Fashoto, S. G., & Ogbu, N. (2022). Performance evaluation of machine learning tools for detection of phishing attacks on web pages. Scientific African, 16, e01165. https://doi.org/10.1016/j.sciaf.2022.e01165

[3] Al-Ahmadi, S., Alotaibi, A., & Alsaleh, O. (2022). PDGAN: Phishing Detection With Generative Adversarial Networks. IEEE Access, 10, 42459–42468. https://doi.org/10.1109/access.2022.3168235

[4] Salahdine, F., El Mrabet, Z., & Kaabouch, N. (2021). Phishing Attacks Detection A Machine Learning-Based Approach. 2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON). https://doi.org/10.1109/uemcon53757.2021.9666627

[5] Aljabri, M., & Mirza, S. (2022). Phishing Attacks Detection using Machine Learning and Deep Learning Models. 2022 7th International Conference on Data Science and Machine Learning Applications (CDMA). https://doi.org/10.1109/cdma54072.2022.00034.

[6] Magdy, S., Abouelseoud, Y., & Mikhail, M. (2022). Efficient spam and phishing emails filtering based on deep learning. Computer Networks, 206, 108826. https://doi.org/10.1016/j.comnet.2022.108826

[7] Rajab, K. D. (2017). New Hybrid Features Selection Method: A Case Study on Websites Phishing. Security and Communication Networks, 2017, 1–10. https://doi.org/10.1155/2017/9838169

[8] Feng, F., Zhou, Q., Shen, Z., Yang, X., Han, L., & Wang, J. (2018). The application of a novel neural network in the detection of phishing websites. Journal of Ambient Intelligence and Humanized Computing. https://doi.org/10.1007/s12652-018-0786-3

[9] Khonji, M., Iraqi, Y., & Jones, A. (2013). Phishing Detection: A Literature Survey. IEEE Communications Surveys & Tutorials, 15(4), 2091–2121. https://doi.org/10.1109/surv.2013.032213.00009

[10] Bountakas, P., & Xenakis, C. (2023). HELPHED: Hybrid Ensemble Learning PHishing Email Detection. Journal of Network and Computer Applications, 210, 103545. https://doi.org/10.1016/j.jnca.2022.103545

[11] Barraclough, P. A., Fehringer, G., & Woodward, J. (2021). Intelligent cyber-phishing detection for online. Computers & Security, 104, 102123. https://doi.org/10.1016/j.cose.2020.102123

[12] Li, Q., Cheng, M., Wang, J., & Sun, B. (2022). LSTM Based Phishing Detection for Big Email Data. IEEE Transactions on Big Data, 8(1), 278–288. https://doi.org/10.1109/tbdata.2020.2978915

[13] Alhogail, A., & Alsabih, A. (2021). Applying machine learning and natural language processing to detect phishing email. Computers & Security, 110, 102414. https://doi.org/10.1016/j.cose.2021.102414

[14] Gowtham, R., & Krishnamurthi, I. (2014). A comprehensive and efficacious architecture for detecting phishing webpages. Computers & Security, 40, 23–37. https://doi.org/10.1016/j.cose.2013.10.004

[15] Li, W., Meng, W., Tan, Z., & Xiang, Y. (2019). Design of multi-view based email classification for IoT systems via semi-supervised learning. Journal of Network and Computer Applications, 128, 56–63. https://doi.org/10.1016/j.jnca.2018.12.002

[16] Alhogail, A., & Alsabih, A. (2021). Applying machine learning and natural language processing to detect phishing email. Computers & Security, 110, 102414. https://doi.org/10.1016/j.cose.2021.102414

[17] Dada, E. G., Bassi, J. S., Chiroma, H., Abdulhamid, S. M., Adetunmbi, A. O., & Ajibuwa, O. E. (2019). Machine learning for email spam filtering: review, approaches and open research problems. Heliyon, 5(6), e01802. https://doi.org/10.1016/j.heliyon.2019.e01802

[18] Smadi, S., Aslam, N., & Zhang, L. (2018). Detection of online phishing email using dynamic evolving neural network based on reinforcement learning. Decision Support Systems, 107, 88–102. https://doi.org/10.1016/j.dss.2018.01.001

[19] Mageshkumar, N., Vijayaraj, A., Arunpriya, N., & Sangeetha, A. (2022). Efficient spam filtering through intelligent text modification detection using machine learning. Materials Today: Proceedings, 64, 848–858. https://doi.org/10.1016/j.matpr.2022.05.364
[20] El Aassal, A., Baki, S., Das, A., & Verma, R. M. (2020). An In-Depth Benchmarking and Evaluation of Phishing Detection Research for Security Needs. IEEE Access, 8, 22170–22192. https://doi.org/10.1109/access.2020.2969780
[21] Androutsopoulos, I., Koutsias, J., Chandrinos, K. V., & Spyropoulos, C. D. (2000). An experimental comparison of naive Bayesian and keyword-based anti-spam filtering with personal e-mail messages. Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. https://doi.org/10.1145/345508.345569
[22] Sakkis, G., Androutsopoulos, I., Paliouras, G., Karkaletsis, V., Spyropoulos, C. D., & Stamatopoulos, P. (2001, June 19). Stacking classifiers for anti-spam filtering of e-mail. arXiv.org. https://arxiv.org/abs/cs/0106040
[23] Androutsopoulos I. Paliouras G. Michelakis E. Learning to filter unsolicited commercial e-mail (Vol. 2004). National Center for Scientific Research, 2004.
[24] Bayes Theorem. (2021). Encyclopedia of Evolutionary Psychological Science, 522–522. https://doi.org/10.1007/978-3-319-19650-3_300417
[25] Koprinska, I., Poon, J., Clark, J., & Chan, J. (2007). Learning to classify e-mail. Information Sciences, 177(10), 2167–2187. https://doi.org/10.1016/j.ins.2006.12.005
[26] Abu-Nimeh, S., Nappa, D., Wang, X., & Nair, S. (2007). A comparison of machine learning techniques for phishing detection. Proceedings of the Anti-Phishing Working Groups 2nd Annual ECrime Researchers Summit. https://doi.org/10.1145/1299015.1299021
[27] Available at, CS Mining Group, 2010, http://www.csmining.org/index.php/malicious-software-datasets-.html.
[28] Yan Gao, Ming Yang, Xiaonan Zhao, Bryan Pardo, Ying Wu, Pappas, T. N., & Alok Choudhary. (2008). Image spam hunter. 2008 IEEE International Conference on Acoustics, Speech and Signal Processing. https://doi.org/10.1109/icassp.2008.4517972
[29] Tastemir, B. B., Malikova, F. U., & Aitbayeva, R. B. (2022). RANDOM FORESTS MACHINE LEARNING TECHNIQUE FOR EMAIL SPAM FILTERING. SERIES PHYSICO-MATHEMATICAL, 2(342), 130–141. https://doi.org/10.32014/2022.2518-1726.134
[30] Rapacz, S., Chołda, P., & Natkaniec, M. (2021). A Method for Fast Selection of Machine-Learning Classifiers for Spam Filtering. Electronics, 10(17), 2083. https://doi.org/10.3390/electronics10172083
[31] Biggio, B., Fumera, G., Pillai, I., & Roli, F. (2008, August). Improving image spam filtering using image text features. In Proc of the fifth conf on email and anti-spam.
[32] Seewald, A. K. (2007). An evaluation of Naive Bayes variants in content-based learning for spam filtering. Intelligent Data Analysis, 11(5), 497–524. https://doi.org/10.3233/ida-2007-11505
[33] Utari, M. I., & Medyawati, H. (2020). CLASSIFICATION OF NEWS TYPES BY IMPLEMENTING ENHANCED CONFIX STRIPPING STEMMER. International Journal of Engineering Technologies and Management Research, 6(5), 135–141. https://doi.org/10.29121/ijetmr.v6.i5.2019.380
[34] DENG, W., & HONG, Z. (2010). Double-stage spam filtering method based on rough set. Journal of Computer Applications, 30(8), 2006–2009. https://doi.org/10.3724/sp.j.1087.2010.02006
[35] Ray, S. (6). Easy Steps to Learn Naive Bayes Algorithm with codes in Python and R," 2017.
[36] Vapnik, V. N. (1995). The Nature of Statistical Learning Theory. Springer New York. https://doi.org/10.1007/978-1-4757-2440-0
[37] Liu, W., & Wang, T. (2010). Index-based Online Text Classification for SMS Spam Filtering. Journal of Computers, 5(6). https://doi.org/10.4304/jcp.5.6.844-851
[38] Gabriel, A. O., & Joy, A. A. (2022). An Email Spam Filtering Model Using Ensemble of Machine Learning Techniques. International Journal of Computer Applications Technology and Research, 11(03), 66–71. https://doi.org/10.7753/ijcatr1103.1003
[39] Akinyelu, A. A., & Adewumi, A. O. (2014). Classification of Phishing Email Using Random Forest Machine Learning Technique. Journal of Applied Mathematics, 2014, 1–6. https://doi.org/10.1155/2014/425731
[40] L. Breiman, A. Cutler, Random Forests-Classification Description, Department of Statistics Homepage, 2007. http://www.stat.berkeley.edu/~breiman/RandomFo rests/cchome.htm.
[41] Fette, I., Sadeh, N., & Tomasic, A. (2006). Learning to Detect Phishing Emails. Defense Technical Information Center. https://doi.org/10.21236/ada456046
[42] Stowell, D., & Plumbley, M. D. (2014). Automatic large-scale classification of bird sounds is strongly improved by unsupervised feature learning. PeerJ, 2, e488. https://doi.org/10.7717/peerj.488
[43] P Guerra, P. H. C., Guedes, D., Meira, J. W., Hoepers, C., Chaves, M. H. P. C., & Steding-Jessen, K. (2010, July). Exploring the spam arms race to characterize spam evolution. In Proceedings of the 7th Collaboration, Electronic messaging, Anti-Abuse and Spam Conference (CEAS), Redmond, WA.
[44] Breiman, L. (1996). Bagging predictors. Machine Learning, 24(2), 123–140. https://doi.org/10.1007/bf00058655

[45] Random Forests. (2017). Encyclopedia of Machine Learning and Data Mining, 1054–1054. https://doi.org/10.1007/978-1-4899-7687-1_695
[46] Qiu, Y.-Z. (2023). Universal adversarial perturbations for multiple classification tasks with quantum classifiers. Machine Learning: Science and Technology, 4(4), 045009. https://doi.org/10.1088/2632-2153/acffa3
[47] Friedman, J., Hastie, T., & Tibshirani, R. (2000). Additive logistic regression: a statistical view of boosting (With discussion and a rejoinder by the authors). The Annals of Statistics, 28(2). https://doi.org/10.1214/aos/1016218223
[48] El Aassal, A., Baki, S., Das, A., & Verma, R. M. (2020). An In-Depth Benchmarking and Evaluation of Phishing Detection Research for Security Needs. IEEE Access, 8, 22170–22192. https://doi.org/10.1109/access.2020.2969780
[49] Sankhwar, S., Pandey, D., & Khan, R. A. (2019). Email Phishing: An Enhanced Classification Model to Detect Malicious URLs. ICST Transactions on Scalable Information Systems, 6(21), 158529. https://doi.org/10.4108/eai.13-7-2018.158529
[50] Talaei Pashiri, R., Rostami, Y., & Mahrami, M. (2020). Spam detection through feature selection using artificial neural network and sine–cosine algorithm. Mathematical Sciences, 14(3), 193–199. https://doi.org/10.1007/s40096-020-00327-8
[51] Deepika Mallampati. (2018). An Efficient Spam Filtering using Supervised Machine Learning Techniques. International Journal of Scientific Research in Computer Science and Engineering, 6(2), 33–37. https://doi.org/10.26438/ijsrcse/v6i2.3337
[52] Gangavarapu, T., Jaidhar, C. D., & Chanduka, B. (2020). Applicability of machine learning in spam and phishing email filtering: review and approaches. Artificial Intelligence Review, 53(7), 5019–5081. https://doi.org/10.1007/s10462-020-09814-9
[53] Singh, M. C., Sumanth, P., Sathyanarayana, S. B., & Rithika, G. (2022). Phishing email detection using deep learning algorithms. International Journal of Health Sciences, 8130–8139. https://doi.org/10.53730/ijhs.v6ns3.7944
[54] Alghamdi, J., Lin, Y., & Luo, S. (2022). A Comparative Study of Machine Learning and Deep Learning Techniques for Fake News Detection. Information, 13(12), 576. https://doi.org/10.3390/info13120576
[55] Singh, M. C., Sumanth, P., Sathyanarayana, S. B., & Rithika, G. (2022). Phishing email detection using deep learning algorithms. International Journal of Health Sciences, 8130–8139. https://doi.org/10.53730/ijhs.v6ns3.7944
[56] AbdulNabi, I., & Yaseen, Q. (2021). Spam Email Detection Using Deep Learning Techniques. Procedia Computer Science, 184, 853–858. https://doi.org/10.1016/j.procs.2021.03.107
[57] Sakunthala Jenni, R., & Shankar, S. (2022). Semantic Based Greedy Levy Gradient Boosting Algorithm for Phishing Detection. Computer Systems Science and Engineering, 41(2), 525–538. https://doi.org/10.32604/csse.2022.019300
[58] Palanichamy, N., & Murti, Y. S. (2023). Improving Phishing Email Detection Using the Hybrid Machine Learning Approach. Journal of Telecommunications and the Digital Economy, 11(3), 120–142. https://doi.org/10.18080/jtde.v11n3.778
[59] Abdulraheem, R., Odeh, A., Al Fayoumi, M., & Keshta, I. (2022). Efficient Email phishing detection using Machine learning. 2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC). https://doi.org/10.1109/ccwc54503.2022.9720818
[60] M, M., & Godara, S. (2019). Analysis of various Machine Learning Techniques to Detect Phishing Email. International Journal of Computer Applications, 178(38), 4–12. https://doi.org/10.5120/ijca2019919251
[61] Alshingiti, Z., Alaqel, R., Al-Muhtadi, J., Haq, Q. E. U., Saleem, K., & Faheem, M. H. (2023). A Deep Learning-Based Phishing Detection System Using CNN, LSTM, and LSTM-CNN. Electronics, 12(1), 232. https://doi.org/10.3390/electronics12010232
[62] Venkat, D. S. S. P. (2023). Improved Phishing Detection using Ensemble Models in Machine Learning. International Journal for Research in Applied Science and Engineering Technology, 11(6), 3616–3620. https://doi.org/10.22214/ijraset.2023.54359
[63] Freund, Y., & Schapire, R. E. (1997). A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. Journal of Computer and System Sciences, 55(1), 119–139. https://doi.org/10.1006/jcss.1997.1504
[64] Zhao, S., Xu, Z., Liu, L., Guo, M., & Yun, J. (2018). Towards Accurate Deceptive Opinions Detection Based on Word Order-Preserving CNN. Mathematical Problems in Engineering, 2018, 1–9. https://doi.org/10.1155/2018/2410206
[65] Vinayakumar, R., Soman, K. P., Poornachandran, P., Mohan, V. S., & Kumar, A. D. (2018). ScaleNet: Scalable and Hybrid Framework for Cyber Threat Situational Awareness Based on DNS, URL, and Email Data Analysis. Journal of Cyber Security and Mobility. https://doi.org/10.13052/2245-1439.823
[66] Castillo, E., Dhaduvai, S., Liu, P., Thakur, K. S., Dalton, A., & Strzalkowski, T. (2020, May). Email threat detection using distinct neural network approaches. In Proceedings for the First International Workshop on Social Threats in Online Conversations: Understanding and Management (pp. 48-55).
[67] Lee, Y., Saxe, J., & Harang, R. (2020). Catbert: Context-aware tiny bert for detecting social engineering emails. arXiv preprint arXiv:2010.03484.
[68] Vinayakumar, R., Soman, K. P., Prabaharan Poornachandran, Akarsh, S., & Elhoseny, M. (2019). Deep Learning Framework for Cyber Threat Situational Awareness Based on Email and URL Data Analysis. Cybersecurity and Secure Information Systems, 87–124. https://doi.org/10.1007/978-3-030-16837-7_6
[69] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.
[70] Jen Tzung. C. (2019). Chapter 7. Deep Neural Network, Source Separation and Machine Learning, Academic Press. Pages 259-320, ISBN 9780128177969.

[71] Cho, K., Van Merriënboer, B., Bahdanau, D., & Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259.
[72] Zhang, L., Xu, Z., Lu, Z., & Wang, S. (2020). An efficient deep convolutional network for email spam classification. IEEE Access, 8, 131617-131626.