International Journal of Informatics Engineering and Computing (IJIMATIC)
Vol.3, No.1, 2026
E-ISSN: 3090-9112

Comparing Deep Learning and Machine Learning Approaches for Spam Email Detection

Hamzah[1], Niko Irsyad Maulana[2], Akhmad Wakhid[3]
[1,2] Department of Informatics, Universitas Respati Yogyakarta, Indonesia
[3] Harbin University of Science and Technology, China

Abstract
Spam messages remain one of the most prevalent cybersecurity threats because they facilitate phishing, fraud, and malware distribution through digital communication platforms. This paper evaluates the effectiveness of six machine learning algorithms for spam message detection, namely Convolutional Neural Network (CNN), Support Vector Machine (SVM), Decision Tree, K-Nearest Neighbor (KNN), Gaussian Naïve Bayes (GNB), and Gradient Boosting (GBoost). The proposed approach compares the performance of these classifiers using accuracy, precision, recall, and F1-score to identify the most reliable model for distinguishing spam and legitimate messages. Experimental results demonstrate that the CNN achieves the highest overall accuracy of 99%, with precision, recall, and F1-score of 0.99, 1.00, and 0.99 for normal messages, and 0.99, 0.92, and 0.95 for spam messages, respectively. Gradient Boosting provides the second-best performance with an accuracy of 92%, while SVM, Decision Tree, and KNN achieve accuracies ranging from 88% to 90% but exhibit lower recall for the spam class. Gaussian Naïve Bayes produces the weakest performance with an accuracy of only 16%, indicating that its feature independence assumption is insufficient for modeling the complex relationships within textual spam data. These findings demonstrate that deep learning, particularly CNN, provides more accurate and balanced spam detection than conventional machine learning approaches. The proposed model offers an effective solution for practical spam filtering systems by minimizing both false positives and false negatives while maintaining high classification reliability. Future work will investigate transformer-based language models, hybrid deep learning architectures, explainable artificial intelligence techniques, and larger multilingual datasets to further improve detection accuracy, interpretability, and robustness against evolving spam patterns.

Keywords:
Spam Detection, Machine Learning, CNN, Cybersecurity

1. Introduction

Email spam detection is an important concern in the world of cybersecurity and digital communications due to the increasing number of spam emails and their potential threats to individuals and organizations. Unsolicited deceptive emails, commonly known as spam, can potentially contain malicious software and gain access to sensitive information without the recipient's consent [1]. They have the potential to occupy significant space and consume a considerable amount of bandwidth, which can impact the effectiveness of email communication [2]. In addition, the effectiveness of email communication is hindered by the limitations of current spam filtering techniques, which often misclassify legitimate emails as spam (resulting in false positives) [3]. Furthermore, the adaptive nature of spam emails makes it necessary to continuously improve spam detection models to effectively identify and filter out spam emails [4]. Overall, the development of accurate and efficient spam detection models using machine learning algorithms is crucial for protecting users'

Corresponding Author: Hamzah, Universitas Respati Yogyakarta, Indonesia (hamzahi@respati.ac.id)
1. Hamzah, Universitas Respati Yogyakarta, Indonesia
2. Niko Irsyad Maulana, Universitas Respati Yogyakarta, Indonesia
3. Akhmad Wakhid, Harbin University of Science and Technology, China

confidential information and ensuring the smooth functioning of email communication systems [5].

The primary difficulties in identifying email spam involve the incorrect categorization of legitimate emails as spam, commonly referred to as false positives, the increasing volume of spam emails, and the need for effective filtering methods [6]. Identifying and separating spam emails from legitimate emails can be hindered by the use of spoofed email addresses, where the sender's email address is manipulated to appear genuine [2]. Machine learning methods that rely on the content of emails have demonstrated their effectiveness in identifying and screening out spam emails, but the large increase in email spamming requires continuous study and classification of emails [7]. Additionally, the use of artificial intelligence-enabled Deepfakes has exacerbated the issue of social spam, where coordinated automated accounts spread spam, rumors, and fake news on social networking sites [8]. These difficulties underscore the importance of developing strong detection systems, dynamic datasets in real-time, and effective approaches to combat malicious efforts aimed at undermining machine learning-based spam filters [9].

Effective and fast detection techniques are needed because they allow for the accurate and timely identification of various events or conditions. In the context of voltage sags in power systems, a fast and accurate algorithm is proposed for detecting voltage sags, which can affect the operation of equipment connected to the power grid [10]. In the field of neuroimaging, the presence of extensive datasets requires the creation of rapid approaches for detecting changes in brain networks, enabling neuroscientists to hypothesize about potential mechanisms [11]. Similarly, in the domain of salient object detection, a novel framework is introduced, which distinguishes between semantic context, spatial intricacies, and boundary data, addressing each aspect individually, leading to improved accuracy and efficiency [12]. In the context of photovoltaic array fault detection, the importance of rapid and precise methods cannot be overstated, as they play a pivotal role in improving the effectiveness, dependability, and security of PV systems [13].

CNNs were selected as a method for email spam detection because of their capacity to efficiently examine and categorize text-based information. CNNs have demonstrated their effectiveness in various natural language processing applications, such as sentiment analysis and text categorization. They excel at capturing localized patterns and relationships within textual data, a vital capability for detecting spam emails. Moreover, CNNs are well-suited to processing extensive datasets and autonomously acquiring intricate features, reducing the need for manual feature engineering [14]. The use of CNNs in spam detection has been demonstrated in multiple papers, such as Samarthrao et al [15] and Huang [2]. These studies highlight the effectiveness of CNNs in detecting spam emails by leveraging word embeddings and sequential properties of the text.

Comparing CNN to conventional methods in machine learning like SVM, KNN, Random Forest, Decision Tree, and XGBoost algorithms. Relevance in this context stems from its capacity to offer insights into the benefits of employing CNNs in various applications. CNN models have shown remarkable performance in tasks such as sentiment analysis [16], breast tumor segmentation [17], fake news classification [18], and mitosis detection [19]. These models have the ability to extract higher-level features and capture long-term dependencies, which can lead to improved accuracy and performance [20]. In contrast, traditional machine learning models may not be able to capture complex patterns and relationships in the data as effectively as deep learning models. Therefore, comparing CNN to conventional machine learning techniques allows researchers to understand the advantages and limitations of each approach and make informed decisions about which model to use for specific tasks.

Performance measurement of a model involves evaluating how well the model learns the given data and applies that learning in practice. Different types of performance evaluations exist, and the selection of evaluation metrics is contingent upon the particular

use case and machine learning technique being used. For example, in the context of machine learning models for health conditions, average performance over an entire population of interest is assessed, and subpopulation performance metrics are computed to evaluate fairness and equity [21]. In the case of botnet detection using deep learning models, Performance measures like accuracy, sensitivity, specificity, precision, and F1 score are utilized for the evaluation of performance and how well models can classify both recognized and unfamiliar botnet traffic patterns [22]. Similarly, in the evaluation of intrusion detection systems, performance metrics like precision and recall are computed to assess the effectiveness of selected classifiers [23]. In the realm of recognizing handwritten digits, the assessment involves a comparison of the accuracy and error rates among various classifiers, and the suggested model (SIFT+PCA+CNN) demonstrates superior accuracy and reduced errors [24].

The potential implications of the studies include improving email spam detection and further development in cybersecurity. In the realm of spam detection research, machine learning algorithms are commonly employed like Naive Bayes, SVM, and Ensemble Voting classifiers to achieve precise categorization of spam emails [1] [25]. Additionally, the use of ensemble voting classifiers is proposed to improve accuracy and performance measures [2]. Another study focuses on developing a novel spam detection model using deep learning approaches like Recurrent Neural Network (RNN) and CNN [26]. Furthermore, the research investigates the potential for adversarial attacks against machine learning-based spam detectors and proposes text crafting methods to improve attack effectiveness [27]. The results of these studies can be applied to enhance existing spam detection systems, protect users from spam emails, and contribute to the development of more robust cybersecurity measures.

2. Related Works

Email spam is characterized as unsolicited and misleading electronic messages that are transmitted to individuals or organizations, often containing malicious content and seeking unauthorized access to sensitive information [1]. It is a significant problem in digital communication due to its potential to compromise user's confidential information and disrupt communication channels [2]. The literature emphasizes the need for effective spam detection methods to mitigate this problem [28]. Various algorithms and techniques have been proposed for spam detection, including machine learning algorithms such as Naive Bayes, Logistic Regression, SVM, and Artificial Neural Networks (ANN) [29]. These methods utilize different feature sets, such as stopwords and word count, to classify emails as spam or ham [3]. Identifying spam emails by analyzing their content, checking for malware, and assessing sender information is essential for minimizing the risk to users' sensitive data. In summary, effective email spam detection plays a vital role in upholding the security and trustworthiness of digital correspondence.

Different strategies have been utilized for identifying email spam, and one alternative approach is to leverage machine learning methods such as Multinomial Naive Bayes, Logistic Regression, Linear Support Vector Machines (SVM), and ANN models. These algorithms are used to differentiate between spam and legitimate (ham) emails by analyzing the text content and metadata present in an email message [1]. An alternative method involves utilizing an Artificial Neural Network (ANN) while employing the Term Frequency-Inverse Document Frequency (TFIDF) technique. This method compares the confusion matrix, accuracy, and precision of spam detection [30]. Traditional machine learning classifiers like Naive Bayes, SVM, k-Nearest Neighbor, Decision Trees, Random Forests, and AdaBoost have also been used for spam detection. In addition, an Ensemble Voting classifier has been used to boost accuracy by merging the predictions made by individual classifiers [25]. A novel technique for detecting spam emails has been suggested, which combines the strengths of the Naive Bayes and J48 decision tree

algorithms through a hybrid bagging approach. This approach considers features such as repetitive keywords, Cc or Bcc, domain, and header [3]. Supervised machine learning techniques, including Naive Bayes, SVM, and Random Forest Classifier, have also been used for spam detection with high accuracy rates [31].

Machine learning algorithms such as SVM, KNN, Random Forest, Decision Tree, and XGBoost have been implemented in email spam detection. These algorithms have been applied to various datasets to classify emails as spam or non-spam. The results of these approaches have shown high accuracy rates, ranging from 91.5% to 98.8% [1]. Multinomial Naive Bayes, Bernoulli Naive Bayes, GNB, Random Forest Classifier, and SVM have achieved accuracy rates of 98.8%, 97.6%, 91.5%, 97.8%, and 98.5% respectively [32]. However, there are limitations to these approaches. Some of the limitations include the need for pre-processing and feature extraction, the reliance on labeled datasets for training, and the potential for false alarms or missed detections [30] [31]. Additionally, the effectiveness of these algorithms can be influenced by factors such as general model precision, the probability of various model mistakes, and user confidence [33].

CNN are extensively used in image analysis for email spam detection. CNN models have been applied to detect image spams, which are unwanted content embedded inside images, and pose a threat to email-based communication systems [34]. CNN's ability to process grid-like data, such as images, makes it suitable for this task [35]. CNNs have demonstrated their ability to autonomously learn distinctive features and effectively classify diseases using medical images. They have been applied in various medical applications, such as automatically categorizing skin lesions, identifying diabetic retinopathy, and classifying X-rays for COVID detection. [36]. In the context of email spam detection, CNNs have been trained using various image features and have shown robustness in detecting image spams [4]. The use of CNNs in this domain offers a new tool for detecting image spams and has been compared against existing tools [37].

The general architecture of a CNN consists of multiple layers, Comprising an input layer, hidden layers, and an output layer, the hidden layers usually encompass convolutional layers, ReLU layers, pooling layers, and fully-connected layers are commonly found within the hidden layers of a neural network, which usually include convolutional layers, ReLU layers, and pooling layers [38]. The pooling layers reduce the spatial dimensions of the data, and the fully connected layers establish connections between all neurons in the network. These layers work together to automatically detect features in images and classify them accurately [39]. In the context of email spam detection, machine learning algorithms, including CNNs, are used to classify emails as spam or ham (valid). Two feature sets, specifically stopwords and word count, are employed to ascertain the characteristics of an email based on its textual content and fields. Multinomial Naive Bayes, Logistic Regression, Linear SVM, and ANN algorithms are applied to compare the performance of these feature sets. Through the examination of email content, malware presence, and sender details, the identification of spam emails can substantially mitigate the risk to users' sensitive data [34].

Email spam detection using CNN has shown significant advantages over traditional machine learning algorithms. Several papers have compared the performance of CNN with other algorithms and found that CNN outperforms them in terms of accuracy, precision, and recall. For example, Sethi et al. [1] After comparing the effectiveness of Multinomial Naive Bayes, Logistic Regression, Linear SVM, and ANN algorithms in contrast with CNN, it was determined that CNN proved to be a more dependable approach for spam detection. Similarly, Sheneamer compared deep learning methods like CNN and LSTM with traditional machine learning algorithms and found that deep learning algorithms, especially CNN with the GloVe model, achieved higher accuracy scores [40]. These results indicate that CNN is a promising approach for email spam detection, offering improved performance compared to traditional machine learning algorithms.

Several factors influence the performance of CNNs in email spam detection. One

important factor is the use of word embedding techniques, such as BERT, which enhance the accuracy of the models in classifying spam emails [1]. Another factor is the incorporation of sentiment analysis and sequential properties of the email body text using techniques like Word-Embeddings and Bidirectional LSTM networks [15]. Additionally, the use of CNNs to extract higher-level text features can improve the training time and overall performance of the model [30]. It is also worth noting that adversarial attacks on machine learning models, including spam detectors, can affect the effectiveness of CNNs in detecting spam emails [41]. These factors collectively contribute to the performance of CNNs in email spam detection.

Machine learning and CNN techniques have been implemented in the context of phishing detection on websites. Researchers have investigated computer vision methods and developed deep learning and machine learning classifiers for the identification of phishing websites and their associated brands. [42]. Various machine learning algorithms like Logistic Regression, Adaboost, and Gradient boost have been compared to identify the approach that provides maximum accuracy rate and time effectiveness [43]. The proposed Stacking Classifier achieved an accuracy of 85.6% in phishing URL detection [44]. A novel CNN with self-attention, called self-attention CNN, has been proposed for phishing URL identification, which achieved an accuracy of 95.6% and outperformed other models like CNN-LSTM, single CNN, and single LSTM [45]. A rule-based hybrid solution that incorporates various algorithmic models, including deep learning models like MLP and CNN, achieved high efficiency and accuracy in detecting and controlling phishing activities [46].

Email spam detection and phishing detection on websites have similarities and differences in terms of the approach and algorithms used. Both involve the use of machine learning techniques for classification. In email spam detection, features are extracted from emails and machine learning techniques like decision trees, random forests, and logistic regression are employed to forecast whether an email qualifies as spam or not [47]. Similarly, in phishing detection on websites, machine learning algorithms like C4.5 (J48), random forest, and logistic regression are used to classify websites as phishing or legitimate [48]. However, there are differences in the features used for detection. Email spam detection focuses on features extracted from the email content, while phishing detection on websites considers features related to the URL and website structure [43] [49]. Additionally, the datasets used for training and testing also differ, with email spam detection using the SPAMASSASIAN corpus and phishing detection on websites using datasets from the UCI repository [50].

The literature addresses class imbalances in email spam detection and phishing detection through various techniques. One approach is the use of ensemble strategies and Sampling techniques to address class imbalances in real-world applications [48]. Another approach is the comparison of different natural language processing (NLP) and machine learning methods, such as TF-IDF, Word2Vec, BERT, Random Forest, Decision Tree, Logistic Regression, Gradient Boosting Trees, and Naive Bayes, to detect phishing emails [51]. Additionally, deep learning and machine learning classifiers are used to detect phishing websites and their brands, with experiments conducted using off-the-shelf CNN architectures [42]. Furthermore, the use of email embeddings techniques has been found effective in classifying emails as phishing or legitimate, capturing indicators that contribute to variations in classifiers [52]. These approaches demonstrate the ongoing efforts to address class imbalances and improve the accuracy of email spam and phishing detection.

The assessment of email spam detection and phishing detection models in literature involves the utilization of different evaluation criteria. These criteria encompass metrics such as Accuracy, Precision, Recall, F1 score, and the Confusion matrix [53]. While Accuracy is commonly used, it does not provide a complete picture of the overall performance of the models. Precision and Recall are also important in understanding the working of these algorithms [54]. In addition, the F-score, which combines Precision and Recall, is used to assess the effectiveness of phishing email detection models [55]. Other

studies focus on the ecological validity of phishing detection measures, using tasks such as the Phishing Email Suspicion Test (PEST) to quantify behavior and compare suspicion scores with real-world efficacy [56]. Overall, the literature emphasizes the importance of considering multiple evaluation metrics to accurately assess the performance of email spam and phishing detection models.

Research results related to email spam detection and phishing detection can be applied in real-world scenarios to improve digital communication security and user experience. An alternative method is to create innovative spam detection models that encompass steps such as acquiring datasets, extracting features, selecting the most suitable features, and performing effective detection [2]. These models have the flexibility to utilize various approaches, which involve analyzing text characteristics using methods such as Term Frequency-Inverse Document Frequency (TF-IDF) to extract relevant features, as well as the extraction of visual characteristics through approaches like color correlogram and Gray-Level Co-occurrence Matrix (GLCM) [57]. Additionally, the use of deep learning approaches like RNN and CNN can enhance the performance of spam email classification [58]. Furthermore, benchmarking frameworks like PhishBench can be used to assess and contrast the available attributes for identifying phishing attempts, it is essential to consider the unique attributes of the datasets and the inherent imbalance in phishing attack occurrences. [59]. These research findings can contribute to the development of intelligent anti-spam email filters and improve the detection and prevention of phishing attacks, thereby enhancing digital communication security and user experience.

3. Proposed Method

The proposed model is a CNN for a classification task implemented using TensorFlow/Keras. Let's explain the mathematical concepts behind this model and write the equations. As an an input layer, denoted by the variable "i," is a crucial component of the neural network architecture. It represents the input data, typically of variable length denoted as "T." The input layer, in essence, consists of a vector "i" with a length equal to "T," and it serves as the initial input to the neural model, allowing the network to process and analyze the input data effectively during its computations.

The Embedding Layer is a critical component of natural language processing models. Its primary function is to transform words or tokens from the input data into D-dimensional vectors, where "D" represents the predetermined embedding dimension. The layer's operation is defined by the number of unique tokens in the dataset, denoted as "V," which corresponds to the vocabulary size. The Embedding Layer essentially maps each token to a continuous vector space, enabling the neural network to process and learn from the textual data efficiently. This layer's functionality is mathematically represented in Equation (1), illustrating its role in converting discrete tokens into continuous, numerical representations.

$$x = Embedding(V + 1, D)(i) \tag{1}$$

In this study, the model incorporates three convolutional layers, each sharing a common kernel size of 3. These layers have varying numbers of filters, The initial layer consists of 32 filters, followed by a second layer with 64 filters, and then a third layer with 128 filters. After each convolutional operation, a Rectified Linear Unit (ReLU) activation function is employed to introduce non-linearity to the network. The convolution operation can be represented mathematically represented in Equation (2), which underscores its role in feature extraction and transformation within the neural network architecture.

$$x = \text{Conv1D}(C, K, \text{activation='relu'})(x) \quad (2)$$

where C is the number of filters, K is the kernel size (3 in this case), and "activation='relu'" refers to the ReLU activation function.

Each convolutional layer within the neural network architecture, a MaxPooling1D layer is incorporated to reduce the feature dimension. This MaxPooling operation is performed using a window size of 3, which means it extracts the maximum value within a sliding window of three consecutive values along the feature axis. The MaxPooling calculated in Equation is used to mathematically represent in Equation (3), facilitating the downsampling of feature maps and helping the network capture essential features while reducing computational complexity.

$$x = \text{MaxPooling1D}(3)(x) \quad (3)$$

This study adopts the GlobalMaxPooling1D layer as an essential component in the neural network architecture, specifically designed to extract the maximum value across all the feature vectors generated by the preceding convolutional layers. It operates by selecting the maximum value from each feature map, effectively condensing the spatial information while preserving the most significant features. The GlobalMaxPooling operation is mathematically represented in Equation (4), Explaining its function in diminishing the spatial characteristics of feature maps while preserving the most crucial information for further neural network processing.

$$x = \text{GlobalMaxPooling1D}(x) \quad (4)$$

The Dense Layer, serving as the ultimate layer within the neural network architecture, comprises five neurons, each corresponding to one of the output classes. To generate class probabilities, the activation function applied to this layer is softmax, which normalizes the neuron outputs and generates a probability distribution across the existing classes. Mathematically, the Dense Layer's operation is depicted represented in Equation (5), highlighting its role in mapping the learned features to class probabilities and enabling the network to make predictions or classifications based on the input data.

$$x = \text{Dense}(5, \text{activation='softmax'})(x) \quad (5)$$

The model is prepared for training through a compilation process where it is configured with specific settings. In this case, the choice of loss function is "sparse_categorical_crossentropy," which is suitable for classification tasks with discrete class labels. The model utilizes the "adam" optimizer for effective weight updates throughout the training process, while the "accuracy" metric is employed to evaluate the model's performance. This compilation step ensures that the model is equipped with the appropriate tools for learning from the data, optimizing its parameters, and evaluating its accuracy during the training process. The model undergoes the training process, where it learns from the training dataset and evaluates its performance on validation data over 200 training epochs. A batch size of 100 is utilized for mini-batch gradient descent, facilitating more efficient weight updates.

This model describes the flow of data through a CNN architecture used for a classification task. The dataset utilized in this research is the Spam Email dataset obtained from Kaggle, comprising 5572 records with two features. The target variable in this dataset is binary, with "yes" denoting spam emails and "no" representing non-spam or ham emails. This dataset serves as the foundation for studying and developing models for the classification of spam and non-spam emails, a fundamental task in email filtering and cybersecurity.

4. Experimental Setup

A. Dataset

The dataset utilized in this research is the Spam Email dataset obtained from Kaggle, comprising 5572 records with two features. The target variable in this dataset is binary, with "yes" denoting spam emails and "no" representing non-spam or ham emails. This dataset serves as the foundation for studying and developing models for the classification of spam and non-spam emails, a fundamental task in email filtering and cybersecurity.

B. Preprocessing

In the provided code, the initial data from a CSV file is read using pandas, and then the category labels ('Normal' and 'Spam') are transformed into numerical values using label encoding. Subsequently, the dataset has been split into two subsets, with one designated for training purposes, and the other reserved for testing, and the text in the 'Message' column is tokenized and padded into sequences of integers with a fixed length to be used as input for the CNN model. The CNN model is constructed and trained with the training data, with the use of ModelCheckpoint to save the best-performing model. After training, The model's performance is assessed using the testing dataset, and the results are compared with several other classification methods such as SVM, Decision Tree, KNN, GNB, and GBoost. Finally, the classification results from all methods are evaluated and visualized for performance comparison. All these steps constitute the data preprocessing and analysis process for the spam classification task.

C. Comparison Results

This research employs a CNN as the primary algorithm for the classification task, aimed at Distinguishing between spam and legitimate emails. in the Kaggle-based Spam Email dataset. In addition to the CNN model, the study establishes a set of baseline algorithms, including SVM, Decision Trees, KNN, GNB, and GBoost, to assess CNN's performance against well-established classification techniques. These baseline algorithms provide a benchmark against which CNN's effectiveness and capability to tackle the email spam detection problem can be evaluated, Offering insightful viewpoints on the strengths and weaknesses of each approach in comparison.

D. Training and Evaluation

In this research, the model is trained using 200 epochs and a batch size of 64. The performance evaluation of the model is conducted through the utilization of a Confusion Matrix and a Classification Report. For the Confusion Matrix (CM), it can be represented in Equation (6):

$$Confusion\ Matrix = \begin{bmatrix} TP & FP \\ FN & TN \end{bmatrix} \tag{6}$$

Where:

In email classification, several key metrics are used to evaluate the performance of a model. True Positives (TP) represent the instances correctly predicted as positive, which in this context, are spam emails. False Positives (FP) occur when non-spam emails, or negative cases, are incorrectly classified as positive. On the other hand, False Negatives (FN) account for actual positive instances that are wrongly predicted as negatives. Lastly, True Negatives (TN) stand for accurately predicted negative instances. For the Classification Report, it provides various metrics such as precision, recall, F1-score, and support for both "spam" and "non-spam" classes. The equations for precision (7), recall(8), F1-score(9), and Accuracy (10) are as follows:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{7}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{8}$$

$$\text{F1-Score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{9}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{10}$$

These metrics are essential in assessing the model's performance in distinguishing between spam and legitimate emails, providing a comprehensive view of its classification capabilities.

5. Result and Analysis

A. Training Process

In the section of our experimental analysis, we present two crucial figures that depict our model's performance during the training process. Figure 1 showcases the training accuracy (depicted by the blue line) and validation accuracy (the orange line) of our model, offering insights into how effectively the model generalizes information from the training data to the validation data.

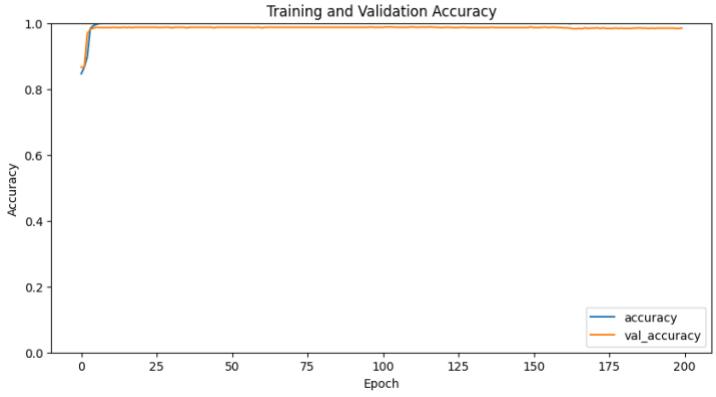

Figure 1. Training and Validation Accuracy

Meanwhile, in Figure 2, We create a visual representation of the fluctuations in training loss (depicted by the blue line) and validation loss (indicated by the orange line) during the training procedure. This figure aids in understanding how the model manages issues such as overfitting or underfitting during training. With these two figures, we can glean valuable insights into the model's performance and adaptation capabilities in distinguishing between spam and non-spam emails within our dataset.

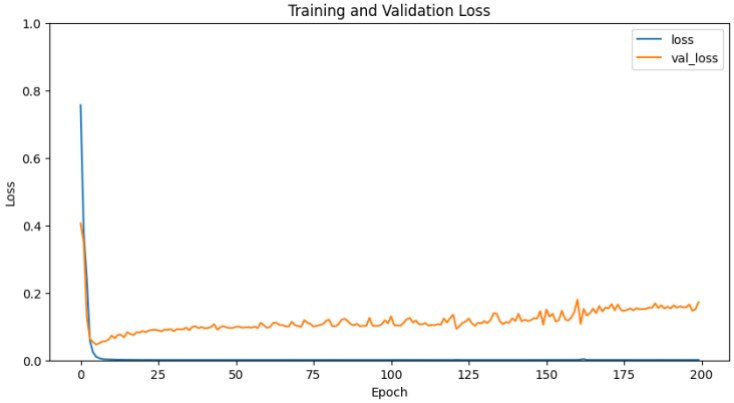

Fig 2. Training and Validation Loss

B. Model Performance

In the subsequent section of our analysis, we present two essential tables that provide a comprehensive overview of the performance metrics for various classification algorithms applied to our dataset. Table 1 encapsulates the Confusion Matrix, displaying key values such as True Positives (TP), False Positives (FP), False Negatives (FN), and True Negatives (TN) for each of the algorithms, including CNN, SVM, Decision Tree, KNN, GNB, and GBoost. This matrix allows us to assess the effectiveness of these algorithms in correctly classifying spam and non-spam emails.

Table 1. Confusion Matrix

Algorithm	TP	FP	FN	TN
CNN	137	1	12	965
SVM	49	7	100	959
Decision Tree	66	37	83	929
KNN	25	11	124	955
GNB	149	939	0	27
GBoost	73	13	76	953

Moving on to Table 2, we delve into the Classification Report, which offers a broader perspective on the model performance. This table includes metrics such as accuracy, precision, recall, and F1-score for each algorithm—CNN, SVM, Decision Tree, KNN, GNB, and GBoost. These metrics enable us to gauge the overall classification performance and the balance between precision and recall, providing insights into the algorithms' strengths and weaknesses in handling email spam detection within our dataset.

Table 2. Classification Report

Algorithm	Accuracy	Class	Precision	Recall	F1-Score
CNN	0.99	Normal	0.99	1.00	0.99
		Spam	0.99	0.92	0.95
SVM	0.90	Normal	0.91	0.99	0.95
		Spam	0.88	0.33	0.48
Decision Tree	0.89	Normal	0.92	0.96	0.94
		Spam	0.64	0.44	0.52
KNN	0.88	Normal	0.89	0.99	0.93
		Spam	0.69	0.17	0.27
GNB	0.16	Normal	1.00	0.03	0.05
		Spam	0.14	1.00	0.24
Gboost	0.92	Normal	0.93	0.99	0.96
		Spam	0.85	0.49	0.62

Table 2 presents the classification performance of six machine learning algorithms for spam message detection. Among all evaluated models, the Convolutional Neural Network (CNN) achieved the highest overall accuracy of 0.99. The CNN also produced excellent performance for both classes, obtaining a precision of 0.99, recall of 1.00, and F1-score of 0.99 for the normal class. For the spam class, the model achieved a precision of 0.99, recall of 0.92, and F1-score of 0.95, indicating that it successfully detected spam messages while maintaining a very low false-positive rate. These results demonstrate that CNN effectively learned complex textual patterns and provided the most reliable classification performance among all evaluated methods.

Gradient Boosting (GBoost) produced the second-best performance with an overall accuracy of 0.92. It achieved an F1-score of 0.96 for the normal class and 0.62 for the spam class, indicating a balanced but less effective performance than CNN in identifying spam messages. Similarly, Support Vector Machine (SVM), Decision Tree, and K-Nearest Neighbor (KNN) achieved accuracies ranging from 0.88 to 0.90. Although these algorithms maintained high recall values for the normal class (0.96–0.99), their recall for the spam class was considerably lower, reaching only 0.33 for SVM, 0.44 for Decision Tree, and 0.17 for KNN. Consequently, their F1-scores for spam detection remained between 0.27 and 0.52, indicating that these conventional machine learning algorithms tended to misclassify a substantial number of spam messages as normal.

Gaussian Naïve Bayes (GNB) demonstrated the weakest performance, achieving an overall accuracy of only 0.16. While the model correctly identified all spam messages with a recall of 1.00, it classified only 3% of normal messages correctly, resulting in an F1-score of 0.05 for the normal class and 0.24 for the spam class. This imbalance suggests that the feature independence assumption of GNB is not suitable for modeling the complex dependencies found in textual spam data. Overall, the experimental results indicate that deep learning, particularly CNN, provides the most accurate and balanced performance for spam message detection, whereas ensemble learning offers competitive results and conventional classifiers exhibit limitations in recognizing minority spam instances.

6. Conclusion

This paper evaluated the performance of six machine learning algorithms for spam message detection, namely Convolutional Neural Network (CNN), Support Vector Machine (SVM), Decision Tree, K-Nearest Neighbor (KNN), Gaussian Naïve Bayes (GNB), and Gradient Boosting (GBoost). We utilized classification metrics, including accuracy, precision, recall, and F1-score, to assess the effectiveness of each model. The experimental results demonstrated that the CNN achieved the highest overall accuracy of 99% and consistently delivered the best performance for both normal and spam classes. The model obtained precision, recall, and F1-score values of 0.99, 1.00, and 0.99 for

normal messages, while achieving 0.99 precision, 0.92 recall, and 0.95 F1-score for spam messages. These findings indicate that CNN effectively captured complex textual patterns and accurately distinguished spam from legitimate messages while maintaining a very low false-positive rate.

The comparative evaluation also showed that Gradient Boosting was the second-best performer with an accuracy of 92%, followed by SVM, Decision Tree, and KNN with accuracies ranging from 88% to 90%. Although these conventional machine learning models achieved high performance in recognizing normal messages, they exhibited considerably lower recall and F1-scores for the spam class, indicating difficulty in identifying minority spam instances. In contrast, Gaussian Naïve Bayes produced the weakest performance, achieving only 16% accuracy. Its extremely low recall for the normal class and poor F1-scores demonstrate that the feature independence assumption is inadequate for representing the complex dependencies among textual features in spam messages. These results confirm that deep learning and ensemble learning approaches provide more robust and balanced classification than conventional probabilistic methods.

Overall, this paper demonstrates that CNN is the most effective algorithm for spam message detection because it provides superior classification accuracy and balanced performance across both message categories. We show that deep learning can significantly improve spam filtering by reducing false positives and false negatives while maintaining reliable detection capability. These findings support the adoption of CNN-based models for practical email and messaging security systems. Future work may investigate transformer-based language models, such as BERT and RoBERTa, hybrid deep learning architectures, explainable artificial intelligence (XAI), and larger multilingual spam datasets to further improve detection accuracy, model interpretability, and robustness against evolving spam patterns.

Acknowledgments

This paper is conducted in the Department of Informatics, Universitas Respati Yogyakarta, Indonesia.

Declaration of Competing Interests

The authors declaire no conflict of interest.

Data Availability

The dataset comes from Kaggle website (https://www.kaggle.com/code/mfaisalqureshi/email-spam-detection-98-accuracy/input)

References

[1] M. Sethi, S. Chandra, V. Chaudhary, and Y. Dahiya, “Spam Email Detection Using Machine Learning and Neural Networks,” in Sentimental Analysis and Deep Learning, S. Shakya, V. E. Balas, S. Kamolphiwong, and K.-L. Du, Eds., Singapore: Springer Singapore, 2022, pp. 275–290.

[2] K. V. Samarthrao and V. M. Rohokale, “Enhancement of email spam detection using improved deep learning algorithms for cyber security,” Journal of Computer Security, vol. 30, no. 2, pp. 231–264, 2022, doi: 10.3233/JCS-200111.

[3] R. Nayak, S. Amirali Jiwani, and B. Rajitha, “WITHDRAWN: Spam email detection using machine learning algorithm,” Materials Today: Proceedings, Apr. 2021, doi: 10.1016/j.matpr.2021.03.147.
[4] A. Theodorus, T. K. Prasetyo, R. Hartono, and D. Suhartono, “Short Message Service (SMS) Spam Filtering using Machine Learning in Bahasa Indonesia,” in 2021 3rd East Indonesia Conference on Computer and Information Technology (EIConCIT), Apr. 2021, pp. 199–203. doi: 10.1109/EIConCIT50028.2021.9431859.
[5] D. Timbadia and N. Vesaokar, “Spam Text Detection,” IJSRCSEIT, pp. 698–704, Jun. 2021, doi: 10.32628/CSEIT2173151.
[6] S. Rao, A. K. Verma, and T. Bhatia, “A review on social spam detection: Challenges, open issues, and future directions,” Expert Systems with Applications, vol. 186, p. 115742, Dec. 2021, doi: 10.1016/j.eswa.2021.115742.
[7] S. P. Z and M. Mohan, “Detection and Prediction of Spam Emails Using Machine Learning Models,” in Handbook of Research on Cyber Crime and Information Privacy, IGI Global, 2020, pp. 201–218. doi: 10.4018/978-1-7998-5728-0.ch011.
[8] S. Shukla, M. Misra, and G. Varshney, “Identification of Spoofed Emails by applying Email Forensics and Memory Forensics,” in 2020 the 10th International Conference on Communication and Network Security, ACM, Nov. 2020. doi: 10.1145/3442520.3442527.
[9] H. Mohmmadzadeh, “Case Study Email Spam Detection of Two Metaheuristic Algorithm for Optimal Feature Selection,” Preprints, Nov. 2020, doi: 10.20944/preprints202001.0309.v3.
[10] F. Kleber de Araújo Lima, J. M. Guerrero, F. L. Tofoli, C. G. C. Branco, and J. L. Dantas, “Fast and accurate voltage sag detection algorithm,” International Journal of Electrical Power & Energy Systems, vol. 135, p. 107516, Feb. 2022, doi: 10.1016/j.ijepes.2021.107516.
[11] P. Bai, A. Safikhani, and G. Michailidis, “A Fast Detection Method of Break Points in Effective Connectivity Networks,” IEEE Transactions on Medical Imaging, vol. 41, no. 5, pp. 1017–1030, May 2022, doi: 10.1109/TMI.2021.3131142.
[12] Z. Zhao, C. Xia, C. Xie, and J. Li, “Complementary Trilateral Decoder for Fast and Accurate Salient Object Detection,” in Proceedings of the 29th ACM International Conference on Multimedia, in MM ’21. New York, NY, USA: Association for Computing Machinery, 2021, pp. 4967–4975. doi: 10.1145/3474085.3475494.
[13] C. Li, Y. Yang, K. Zhang, C. Zhu, and H. Wei, “A fast MPPT-based anomaly detection and accurate fault diagnosis technique for PV arrays,” Energy Conversion and Management, vol. 234, p. 113950, Apr. 2021, doi: 10.1016/j.enconman.2021.113950.
[14] A.-S. Khaled and P. Wanda, “How to Stepping up Characters Recognition using CNN Algorithm?,” International Journal of Informatics and Computation; Vol 4 No 2 (2022): International Journal of Informatics and ComputationDO - 10.35842/ijicom.v4i2.53, Dec. 2022, [Online]. Available: https://ijicom.respati.ac.id/index.php/ijicom/article/view/53
[15] I. AbdulNabi and Q. Yaseen, “Spam Email Detection Using Deep Learning Techniques,” Procedia Computer Science, vol. 184, pp. 853–858, Jan. 2021, doi: 10.1016/j.procs.2021.03.107.
[16] R. Othman, R. Faiz, Y. Abdelsadek, K. Chelghoum, and I. Kacem, “Deep Hybrid Neural Networks with Improved Weighted Word Embeddings for Sentiment Analysis,” in Advances in Intelligent Data Analysis XIX, P. H. Abreu, P. P. Rodrigues, A. Fernández, and J. Gama, Eds., Cham: Springer International Publishing, 2021, pp. 50–62.
[17] W. Gómez-Flores and W. Coelho de Albuquerque Pereira, “A comparative study of pre-trained convolutional neural networks for semantic segmentation of breast tumors in ultrasound,” Computers in Biology and Medicine, vol. 126, p. 104036, Nov. 2020, doi: 10.1016/j.compbiomed.2020.104036.
[18] S. A. Alameri and M. Mohd, “Comparison of Fake News Detection using Machine Learning and Deep Learning Techniques,” in 2021 3rd International Cyber Resilience Conference (CRC), Jan. 2021, pp. 1–6. doi: 10.1109/CRC50527.2021.9392458.
[19] S. Nofallah et al., “Machine learning techniques for mitoses classification,” Computerized Medical Imaging and Graphics, vol. 87, p. 101832, Jan. 2021, doi: 10.1016/j.compmedimag.2020.101832.
[20] R. L. Satria Putra and M. H. Wathan, “Shallots Classification using CNN,” International Journal of Informatics and Computation; Vol 3 No 1 (2021): International Journal of Informatics and Computation, 2022, doi: 10.35842/ijicom.v3i2.41.
[21] B. Nugraha, A. Nambiar, and T. Bauschert, “Performance Evaluation of Botnet Detection using Deep Learning Techniques,” in 2020 11th International Conference on Network of the Future (NoF), Oct. 2020, pp. 141–149. doi: 10.1109/NoF50125.2020.9249198.

[22] Z. Somogyi, “Performance Evaluation of Machine Learning Models,” in The Application of Artificial Intelligence: Step-by-Step Guide from Beginner to Expert, Z. Somogyi, Ed., Cham: Springer International Publishing, 2021, pp. 87–112. doi: 10.1007/978-3-030-60032-7_3.
[23] Andrew C. Miller, Leon A. Gatys, Joseph D. Futoma, and E. Fox, “Model-based metrics: Sample-efficient estimates of predictive model subpopulation performance.”
[24] S. Singhal and P. Yadav, “Evaluation of Model Using J-48 and Other Classifier on Kddcup99 Through Performance Metrics,” in Advanced Informatics for Computing Research, A. K. Luhach, D. S. Jat, K. B. G. Hawari, X.-Z. Gao, and P. Lingras, Eds., Singapore: Springer Singapore, 2019, pp. 12–19.
[25] N. Nisar, N. Rakesh, and M. Chhabra, “Voting-Ensemble Classification for Email Spam Detection,” in 2021 International Conference on Communication information and Computing Technology (ICCICT), Jun. 2021, pp. 1–6. doi: 10.1109/ICCICT50803.2021.9510066.
[26] C. Wang, D. Zhang, S. Huang, X. Li, and L. Ding, “Crafting Adversarial Email Content against Machine Learning Based Spam Email Detection,” in Proceedings of the 2021 International Symposium on Advanced Security on Software and Systems, ACM, May 2021. doi: 10.1145/3457340.3458302.
[27] Ismail B. Mustapha, S. Hasan, S. Olatunji, S. Shamsuddin, and Afolabi Kazeem, “Effective Email Spam Detection System using Extreme Gradient Boosting.”
[28] E. Ferrara, “The History of Digital Spam,” Commun. ACM, vol. 62, no. 8, pp. 82–91, Jul. 2019, doi: 10.1145/3299768.
[29] K. Agarwal and T. Kumar, “Email Spam Detection Using Integrated Approach of Na\" ive Bayes and Particle Swarm Optimization,” in 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), IEEE, Jun. 2018. doi: 10.1109/iccons.2018.8662957.
[30] C. Bansal and B. Sidhu, “Machine Learning based Hybrid Approach for Email Spam Detection,” in 2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Sep. 2021, pp. 1–4. doi: 10.1109/ICRITO51393.2021.9596149.
[31] T. Toma, S. Hassan, and M. Arifuzzaman, “An Analysis of Supervised Machine Learning Algorithms for Spam Email Detection,” in 2021 International Conference on Automation, Control and Mechatronics for Industry 4.0 (ACMI), Jul. 2021, pp. 1–5. doi: 10.1109/ACMI53878.2021.9528108.
[32] S. Kaur, M. Bansal, and A. K. Bathla, “A Comparitive Study of E-Mail Spam Detection using Various Machine Learning Techniques,” presented at the International Conference on Women Researchers in Electronics and Computing, 2021, pp. 426–435. doi: 10.21467/proceedings.114.56.
[33] M. C. Stites, M. Nyre-Yu, B. Moss, C. Smutz, and M. R. Smith, “Sage Advice? The Impacts of Explanations for Machine Learning Models on Human Decision-Making in Spam Detection,” in Artificial Intelligence in HCI, H. Degen and S. Ntoa, Eds., Cham: Springer International Publishing, 2021, pp. 269–284.
[34] K. Khalil, B. Dey, A. Kumar, and M. Bayoumi, “A Reversible-Logic based Architecture for Convolutional Neural Network (CNN),” in 2021 IEEE International Midwest Symposium on Circuits and Systems (MWSCAS), Aug. 2021, pp. 1070–1073. doi: 10.1109/MWSCAS47672.2021.9531842.
[35] “Analysis of Improved CNN for Wireless Network Intrusion Detection,” The Society of Convergence Knowledge Transactions, vol. 9, no. 3, pp. 147–154, Sep. 2021, doi: 10.22716/SCKT.2021.9.3.037.
[36] Z. Zhang, Y. Chen, H. Li, and Q. Zhang, “IA-CNN: A generalised interpretable convolutional neural network with attention mechanism,” in 2021 International Joint Conference on Neural Networks (IJCNN), IEEE, Jul. 2021. doi: 10.1109/ijcnn52387.2021.9533727.
[37] C. Xiao and J. Sun, “Convolutional Neural Networks (CNN),” in Introduction to Deep Learning for Healthcare, C. Xiao and J. Sun, Eds., Cham: Springer International Publishing, 2021, pp. 83–109. doi: 10.1007/978-3-030-82184-5_6.
[38] M. F. Muhdalifah, “Pooling Comparison in CNN Architecture for Javanese Script Classification,” International Journal of Informatics and Computation; Vol 3 No 2 (2021): International Journal of Informatics and Computation, 2022, doi: 10.35842/ijicom.v3i2.30.

[39] Y. Liu, L. Wang, T. Shi, and J. Li, “Detection of spam reviews through a hierarchical attention architecture with N-gram CNN and Bi-LSTM,” Information Systems, vol. 103, p. 101865, Jan. 2022, doi: 10.1016/j.is.2021.101865.
[40] Et. al. Rushali Deshmukh, “Performance Comparison for Spam Detection in Social Media Using Deep Learning Algorithms,” Turkish Journal of Computer and Mathematics Education (TURCOMAT), vol. 12, no. 1S, pp. 193–201, Apr. 2021, doi: 10.17762/turcomat.v12i1s.1609.
[41] S. E. Rahman and S. Ullah, “Email Spam Detection using Bidirectional Long Short Term Memory with Convolutional Neural Network,” in 2020 IEEE Region 10 Symposium (TENSYMP), Jun. 2020, pp. 1307–1311. doi: 10.1109/TENSYMP50017.2020.9230769.
[42] S. Khandelwal and R. Das, “Phishing Detection Using Computer Vision,” in Computer Networks and Inventive Communication Technologies, S. Smys, R. Bestak, R. Palanisamy, and I. Kotuliak, Eds., Singapore: Springer Singapore, 2022, pp. 113–130.
[43] V. K. Nadar, B. Patel, V. Devmane, and U. Bhave, “Detection of Phishing Websites Using Machine Learning Approach,” in 2021 2nd Global Conference for Advancement in Technology (GCAT), Oct. 2021, pp. 1–8. doi: 10.1109/GCAT52182.2021.9587682.
[44] M. Das, S. Saraswathi, R. Panda, A. K. Mishra, and A. K. Tripathy, “Exquisite Analysis of Popular Machine Learning–Based Phishing Detection Techniques for Cyber Systems,” Journal of Applied Security Research, vol. 16, no. 4, pp. 538–562, Oct. 2021, doi: 10.1080/19361610.2020.1816440.
[45] X. Xiao et al., “Phishing websites detection via CNN and multi-head self-attention on imbalanced datasets,” Computers & Security, vol. 108, p. 102372, Sep. 2021, doi: 10.1016/j.cose.2021.102372.
[46] Y. Mourtaji, M. Bouhorma, D. Alghazzawi, G. Aldabbagh, and A. Alghamdi, “Hybrid Rule-Based Solution for Phishing URL Detection Using Convolutional Neural Network,” Wireless Communications and Mobile Computing, vol. 2021, p. 8241104, Sep. 2021, doi: 10.1155/2021/8241104.
[47] S. Ganesan, “Detection of Phishing Websites Using Classification Algorithms,” in Cyber Security and Digital Forensics, K. Khanna, V. V. Estrela, and J. J. P. C. Rodrigues, Eds., Singapore: Springer Singapore, 2022, pp. 129–141.
[48] N. Yadav and S. P. Panda, “Feature Selection for Email Phishing Detection Using Machine Learning,” in International Conference on Innovative Computing and Communications, A. Khanna, D. Gupta, S. Bhattacharyya, A. E. Hassanien, S. Anand, and A. Jaiswal, Eds., Singapore: Springer Singapore, 2022, pp. 365–378.
[49] L. I. Sakri, P. S. Nikkam, M. Kulkarni, P. Kamath, S. S. Bhat, and S. Kamat, “Phishing Websites, Detection and Analysis: A Survey,” in Contemporary Issues in Communication, Cloud and Big Data Analytics, H. K. D. Sarma, V. E. Balas, B. Bhuyan, and N. Dutta, Eds., Singapore: Springer Singapore, 2022, pp. 19–26.
[50] N. Bhoj, R. Bawari, A. Tripathi, and N. Sahai, “Naive and Neighbour Approach for Phishing Detection,” in 2021 10th IEEE International Conference on Communication Systems and Network Technologies (CSNT), Jun. 2021, pp. 171–175. doi: 10.1109/CSNT51715.2021.9509566.
[51] W. Wong and G. Dobbie, “Pelican: Continual Adaptation for Phishing Detection,” in 2020 International Conference on Data Mining Workshops (ICDMW), Nov. 2020, pp. 440–447. doi: 10.1109/ICDMW51313.2020.00067.
[52] P. Bountakas, K. Koutroumpouchos, and C. Xenakis, “A Comparison of Natural Language Processing and Machine Learning Methods for Phishing Email Detection,” in The 16th International Conference on Availability, Reliability and Security, ACM, Aug. 2021. doi: 10.1145/3465481.3469205.
[53] R. Valecha, P. Mandaokar, and H. R. Rao, “Phishing Email Detection Using Persuasion Cues,” IEEE Transactions on Dependable and Secure Computing, vol. 19, no. 2, pp. 747–756, Apr. 2022, doi: 10.1109/TDSC.2021.3118931.
[54] S. B. Torane and Dr. N. Shekokar, “Performance Analysis of Machine Learning Algorithms Used for Web Based Phishing Detection,” Journal of University of Shanghai for Science and Technology, vol. 23, no. 05, pp. 650–656, May 2021, doi: 10.51201/jusst/21/05187.
[55] Z. M. Hakim et al., “The Phishing Email Suspicion Test (PEST) a lab-based task for evaluating the cognitive mechanisms of phishing detection,” Behavior Research Methods, vol. 53, no. 3, pp. 1342–1352, Jun. 2021, doi: 10.3758/s13428-020-01495-0.
[56] S. Salloum, T. Gaber, S. Vadera, and K. Shaalan, “Phishing Email Detection Using Natural Language Processing Techniques: A Literature Survey,” Procedia Computer Science, vol. 189, pp. 19–28, 2021, doi: 10.1016/j.procs.2021.05.077.

[57] M. Fendt et al., “Context and trade-offs characterize real-world threat detection systems: A review and comprehensive framework to improve research practice and resolve the translational crisis,” Neuroscience & Biobehavioral Reviews, vol. 115, pp. 25–33, Aug. 2020, doi: 10.1016/j.neubiorev.2020.05.002.
[58] A. El Aassal, S. Baki, A. Das, and R. M. Verma, “An In-Depth Benchmarking and Evaluation of Phishing Detection Research for Security Needs,” IEEE Access, vol. 8, pp. 22170–22192, 2020, doi: 10.1109/ACCESS.2020.2969780.
[59] Z. Yang, X. Yang, M. Li, and W. Li, “Small-sample learning with salient-region detection and center neighbor loss for insect recognition in real-world complex scenarios,” Computers and Electronics in Agriculture, vol. 185, p. 106122, Jun. 2021, doi: 10.1016/j.compag.2021.106122.