Phishing E-mail Detection with Machine Learning and Deep Learning: Improving Classification Performance with Proposed New Features

Hadjer Brioua, Havvanur Siyambaş, Durmuş Özkan Şahin

Abstract—**Today, with the increasing use of the internet, individuals who use email have become potential targets for fraudsters. These malicious groups send fake or misleading emails to steal sensitive information such as identity, bank, and social media credentials. This tactic is known as phishing. This study proposes a machine learning-based system for detecting phishing attacks using the SeFACED dataset, which was adjusted for binary classification with 12,498 normal and 5,142 fraudulent email data points. Python was used for programming, with Google Colab and Jupyter Notebook as development platforms. Email data underwent data collection, cleaning, and word stem separation processes. Three feature extraction techniques were used: Bag of Words, TF-IDF, and Word2Vec. Six algorithms, including Logistic Regression, Random Forest, Support Vector Machines, Naive Bayes, Convolutional Neural Network, and Long Short-Term Memory, were employed for classification. Performance was evaluated using metrics like accuracy, precision, recall, and F1-score. New attributes proposed to enhance detection included CSS tags, HTML tags, black-list words, link errors, and grammar and spelling errors. The addition of these features generally improved classification results.**

Index Terms—**Phishing, Phishing e-mail, Phishing attacks, Machine learning, Deep learning, Classification, Phishing e-mail classification.**

I. Introduction

WITH the increasing use of the internet worldwide, access to data, services, and products has become easier. Although this has improved accessibility, it has also made systems vulnerable to attacks. As a result, cyber-attacks occur on personal computers, bank accounts, and social media accounts. The most common type of cyber attack is phishing. There are several types of phishing attacks [1], [2]:

- Email Phishing [3]: This is the most common type of phishing attack. An email is sent to the target individuals, giving the impression that it comes from a legitimate organization. Scammers direct recipients to click on a link in the email to steal their sensitive information.
- SMS Phishing [4]: This type of phishing attack is carried out using text messages sent via smartphones.
- Website Phishing [5]: The content of the website in this type of phishing is fake. Scammers request users to enter their information on the relevant website.

When the reports of the Anti-Phishing Working Group (APWG) are examined over the years, it is seen that millions of phishing attacks have been made [6]. Email phishing attack, which is a social engineering attack, is one of the most common phishing attacks [7], [8]. In this study, a machine learning-based architecture for detecting email phishing attacks was developed. To improve the performance of the developed system, new features were added to the system. It is generally observed that the addition of new features improves the model's performance.

A. Literature Review

In the study conducted by Ahi and Soğukpınar, a hybrid method called 'H-OLTA' was proposed to determine whether the relevant email is phishing or not by combining deep learning algorithms such as Long Short-Term Memory (LSTM) [9]. This method's success is higher than that of other classifier algorithms, and it is created by combining multilayer perceptron (MLP) and LSTM algorithms. The accuracy of the developed model is determined to be 96.84%. Two main features were identified to train the model: the subject and body parts of the email text. With their developed method, the subject and body parts of the email are examined separately. Then, a feature matrix is created for the relevant parts, which is used to train the model using deep learning algorithms. The deep learning algorithms used are MLP and LSTM. The datasets used in this study are Jose Nazario's phishing email dataset and the Enron Email Dataset. The dataset, consisting of 4512 emails, is divided into 80% training and 20% test data. The performance metrics used are accuracy, precision, recall, F1-score, and false positive rate (FPR).

In this study, Abdulraheem et al. have approached phishing email detection as a classification problem, demonstrating how machine learning algorithms are used to categorize whether the given email is a phishing attack [10]. The algorithms used in the research include Logistic Model Tree (LMT), MLP, and decision tree. The algorithm achieving the highest accuracy rate in classifying phishing emails is LMT, with an accuracy rate of 96.924%. The dataset used was created by Mohammad et al., containing 11,000 website samples, out

Hadjer Brioua is with the Department of Computer Engineering, Faculty of Engineering, Ondokuz Mayıs University, Samsun, 55200 TURKEY e-mail: hadjer.brioua@bil.omu.edu.tr

Havvanur Siyambas is with the Department of Computer Engineering, Faculty of Engineering, Ondokuz Mayıs University, Samsun, 55200 TURKEY e-mail: havvanur.siyambas@bil.omu.edu.tr

Durmuş Özkan Şahin is with the Department of Computer Engineering, Faculty of Engineering, Ondokuz Mayıs University, Samsun, 55200 TURKEY e-mail: durmus.sahin@bil.omu.edu.tr

Manuscript received May 27, 2024; accepted Jan 10, 2025.
DOI: 10.17694/bajece.1490596

of which 2,500 host phishing URLs. The dataset includes 2,456 instances and 30 features. These 30 features are divided into 4 groups: address bar, unnatural elements, HTML and JavaScript, and domain. The performance metrics used include accuracy, precision, recall, F1-score, and kappa statistic.

In this study, Paradkar used various classification and deep learning algorithms to determine whether an email is phishing [11]. The data went through processes such as data preprocessing and tokenization to convert them into a format suitable for classification. The dataset used is the ENRON CORPUS, consisting of 20,000 email samples, with 8,336 phishing emails and 11,664 normal emails. The dataset was divided into 75% training and 25% test data. The classification and deep learning algorithms used include LR, decision trees, Support Vector Machines (SVM), LSTM, and CNN. According to the study, machine learning algorithms could have been more effective in text classification, but deep learning algorithms achieved high accuracy rates. The highest accuracy rate, at 99.05%, was obtained with CNN.

In this study, Livara and Hernandez addressed the use of machine learning techniques to determine whether emails are phishing or not, and they also investigated the performance of these techniques on imbalanced datasets [12]. The researchers utilized the Phishing Email Collection dataset, obtained from Kaggle, containing 525,754 emails. 90% of the dataset was assigned for training, while the remaining 10% was used for testing. Various visualization tools, such as dot plots and distribution plots, were employed to understand the dataset better. Five machine learning algorithms were used for classification: Naive Bayes (NB), AdaBoost, SVM, LR, and RF. The performance metrics used included accuracy, precision, recall, and F1-score. After extracting features and applying the specified classification algorithms, the RF classifier yielded the highest precision, F1-score, and recall rates. The SVM classifier demonstrated the lowest precision rate at 92%; similarly, lower values were obtained for recall and F1-score.

Akinyelu and Adewumi obtained 2000 phishing email data from Nazario's public phishing email archive [13]. They extracted 15 significant phishing features and then created vector representations of these features for each email. This representation was used to train the relevant classifier. Only the RF classifier was used to train the model. In this study, classifiers were trained and tested using 10-fold cross-validation. The algorithms were tested with datasets of different sizes to measure their performance on small and large datasets. Performance metrics used include false-positive rate, precision, recall, and F1-score. The algorithm showed its best performance when tested on the largest dataset. When the RF classifier was used, the classification accuracy rate was 99.7%, the false-negative rate was 2.50%, and the false-positive rate was 0.06%.

In this study, Dewis and Viana used machine learning and various natural language processing techniques to classify whether the relevant emails were phishing [14]. Experiments were conducted using five different datasets. Each dataset was divided into 70% training and 30% testing samples. Performance metrics used included accuracy, precision, recall, and F1-score. Deep learning algorithms are known to achieve higher accuracy rates when dealing with large datasets, and to mitigate the effects of sudden drops in parameters between hidden layers, more dense layers were added to the MLP algorithm [14], [15], [16], [17]. When the LSTM algorithm was applied to text-based datasets, a 99% accuracy rate was achieved, while for numerical-based datasets, a 94% accuracy rate was achieved for the MLP algorithm.

Eryılmaz et al. combined machine learning and text mining techniques to identify spam emails [18]. The researchers used the Turkish Email dataset. From this dataset, 600 emails were allocated for training the model, and 200 emails were reserved for performance evaluation. The dataset first underwent a preprocessing stage, followed by the use of bag-of-words and TF-IDF approaches to weight and vectorize each word. Different classifiers were used to test the model's success. These algorithms included Sequential Minimal Optimization (SMO), Random Forest (RF), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Logistic Regression (LR), NB, and Multilayer Perceptron (MLP). Metrics such as F1-score, precision, and recall were used to evaluate model performance. Among the different classification algorithms applied, the most successful was SMO, achieving a classification performance of 0.985 in terms of F1-score. In contrast, the least successful was NB, with a classification performance of 0.931.

Singh et al. developed a model using machine learning and deep learning techniques such as K-Means, NB, LSTM, Convolutional Neural Network (CNN), and the BERT model to categorize whether emails are spam or not [19]. The language used to develop the model was Python version 3.7, and the model was developed using natural language processing. The performance metrics used were accuracy and F1-score. The study's results showed that they achieved 92%, 94%, 96%, and 98% accuracy rates for KNN, NB, LSTM, and the BERT model, respectively. The CNN algorithm outperformed all other classification algorithms, demonstrating the most efficient performance in determining whether an email is spam with 99% accuracy.

Sonare et al. explored the effects of using machine learning and deep learning algorithms to determine whether an email is phishing email [20]. They sourced the dataset from the Kaggle platform. The dataset consists of two columns: one indicating the category of the email (spam or normal) and the other containing the email content in the "Message" column. Their methodology includes six steps: loading the data, data collection, and preprocessing, label encoding, splitting the data into training and test sets, feature extraction, and training the model. During the data preprocessing stage, irrelevant and unstructured data were cleaned, and words were stemmed. Label encoding was used to digitize the data. Feature extraction was employed to digitize raw data. The classification and deep learning algorithms used were MLP, SVM, DT, and LR. The performance metrics utilized were precision, F1-score, and recall. As a deep learning algorithm, MLP demonstrated the best performance with a high accuracy rate of 98%. The algorithm with the lowest performance was DT, with an accuracy rate of 94%.

In the research conducted by Adzhar et al., a comparative study was performed on the machine learning algorithms NB, SVM, DT, and RF used for email phishing detection [21]. The study aimed to evaluate previous phishing detection studies to determine which machine learning techniques best detect phishing emails. The definition, characteristics, and categories of phishing attacks were provided in detail. According to this study, a phishing email has five characteristics: it seems too good to be true, creates a sense of urgency, uses links, includes attachments, and comes from someone the user does not know. Phishing attacks were categorized into four groups: link-based, text-based, image-based, and attachment-based. Following the characteristics of phishing emails and the categorization of phishing attacks, the study examined several machine learning algorithms (NB, SVM, DT, RF) that can be used for phishing email detection. A comparative study was conducted on these techniques, and as a result, SVM and RF algorithms were determined to be the best techniques for detecting phishing emails.

In the study by Gupta et al., a new approach was used to detect phishing URLs [22]. The machine learning techniques used in this research are RF, KNN, SVM, and LR. The features were selected based on words. As a result, the RF algorithm achieved the highest accuracy rate.

The study by Moradpoor et al. aims to detect phishing emails [23]. Therefore, a neural network with 6 components was implemented. Phishing emails in the dataset used in this study were obtained from the Phishcorpus dataset, while normal emails were obtained from the SpamAssassin dataset. Initially, using Python code, phishing, and normal emails were selected from the two datasets and represented as normal = 0 and phishing = 1. Then, for each email, the number of web links, the presence of HTML tags, the presence of JavaScript code, and the number of email sections were determined and stored in a boolean or integer variable. Data cleaning and feature extraction were performed in the next step using Word2Vec methods. After vectorization, all variables obtained from the process were saved in a .csv file with 7 columns containing email, vector average, number of web links, HTML presence, JavaScript presence, email section count, and email type. The dataset was divided into 70% training, 15% validation, and 15% testing. A neural network model consisting of Input Matrix and Target Matrix components, 10 hidden layers, 5 input features, 1 output layer, and 1 output feature was developed. The results were evaluated using a confusion matrix and network performance metrics.

In the study by Fayoumi et al., a dataset with 9 features was used to detect phishing emails using machine learning algorithms [24]. The features used include the number of dots in the link, the number of links in the email, the presence of JavaScript codes in the email, the presence of form and HTML tags, the use of action words, and the presence of words like PayPal, bank, and account. In this study, the performances of NB, RF, and SVM algorithms in phishing email detection were compared. Accuracy and F1-score metrics were used to evaluate the results, and the SVM algorithm showed the highest performance.

The study conducted by Salahdine et al. aims to examine the performance of machine learning algorithms used in phishing detection [25]. This study used a dataset consisting of 2000 phishing emails targeting North Dakota University's email system. In the preprocessing step, values were converted into numerical values. The classification process was based on 10 features, such as inconsistencies in the sender's email address, suspicious file extensions, blacklist words, SSL certificates, etc., and SVM, LR, and Artificial Neural Network (ANN) algorithms were used. Metrics such as true positive, false positive, false negative, and accuracy were used to evaluate the results. In this study, the ANN model showed the highest performance. Different activation functions were tried for ANN, and the most successful result was obtained with ReLu.

In Sekiya and Wei's study, the performance of batch machine-learning techniques was examined for detecting phishing websites [26]. Primary machine learning algorithms such as K-Means, SVM, LR, NB, Linear Discriminant Analysis (LDA), Classification & Regression Trees (CART), and RF were compared. It was observed that RF performed the highest in this comparison. Then, ensemble machine learning algorithms such as AdaBoost, Gradient Boosted Decision Trees (GBDT), XGBoost, and LightGBM were also compared, and it was found that RF provided the highest accuracy and LightGBM exhibited the fastest performance. Deep learning models showed better and faster performance when applied to large datasets compared to traditional machine learning. Still, they also have disadvantages such as model architecture design, manual parameter tuning, high training time costs, and computational complexity. This could lead to potential accuracy improvement. Batch machine learning methods have the potential to provide higher accuracy rates because they combine different models. In this study, CART and RF demonstrated the highest performance. Consequently, it is suggested that automatic feature selection methods could address problems such as dealing with large datasets using batch machine learning algorithms.

Jain and Gupta's study focused on detecting phishing attacks using machine learning and hyperlink analysis [27]. The foundation of the study is to develop a machine-learning model by examining hyperlinks in existing HTML codes of browsers. 12 features, such as the total number of links, internal and external links, errors, redirects, and empty links, were used to develop the model. Initially, link features were extracted. In the next stage, feature vectors were created for each website. The performance metrics used in this study include true positive rate, false positive rate, true negative rate, false negative rate, F1-score, accuracy, precision, and recall. LR exhibited the highest performance in this study.

Ahammad et al. utilized phishing emails collected from various sources and normal emails from the Spam Classification dataset in their study [28]. Initially, the data underwent preprocessing, including tokenization and stemming processes. After preprocessing, the words in phishing-containing emails were visualized using the Cloud Module, where the density of words was determined so that more frequently used words had higher density. Then, a corpus containing 100 words related to phishing was created. The next step involved feature engineering, where a new dataset was created. Each word in the corpus represented a feature, and the frequency of each phishing word in the email text was determined as the corresponding value for this feature. Due to the high number of features, feature reduction techniques such as principal component analysis, forward feature selection, backward feature selection, non-negative matrix factorization, and recursive feature elimination were explored, along with cross-validation. Machine learning techniques used in this research included LR, DT, SVM, NB, and KNN. A deep neural network with an input layer of 100 features, 1-2 hidden layers, and one output layer was employed alongside machine learning techniques. The results were evaluated by comparing the accuracy rates and the number of features provided by models that gave the most suitable number of features in each dimension reduction technique. Forward feature selection yielded the highest accuracy rate with the NB algorithm.

Thapa et al. conducted a pioneering study applying federated learning (FL) to phishing email detection [29]. The study investigated the performance of FL on distributed datasets using two state-of-the-art models: THEMIS and BERT. FL enables collaborative model training across multiple organizations without sharing raw data, thus preserving data privacy. The results demonstrated that FL achieved performance comparable to centralized learning (CL) under balanced data distributions, with test accuracies of 96.1% for BERT and 97.9% for THEMIS. However, performance varied under scenarios with asymmetric data distributions or extreme dataset diversity, highlighting model dependency. This study underscores the potential of FL as a privacy-preserving approach to phishing email detection.

Wosah et al. proposed a framework for mitigating phishing attacks by integrating stylometric features, gender identification, and email header analysis into a Colour Code Email Verification (CCEV) system [30]. The framework leverages natural language processing and LSTM techniques to analyze email authenticity. By assigning color codes—green for safe, amber for suspicious, and red for high threat—the system provides real-time sender verification at the recipient's end. The study utilized the Enron email dataset for model development and evaluation, demonstrating that the system effectively assists users in distinguishing between legitimate and phishing emails, thereby enhancing cybersecurity against sophisticated spear-phishing attacks.

Jamal et al. proposed the Improved Phishing and Spam Detection Model (IPSDM), leveraging the capabilities of large language models (LLMs) to classify phishing, spam, and ham emails [31]. The study fine-tuned and optimized transformer-based models, specifically DistilBERT and RoBERTA, demonstrating their superior performance over traditional approaches in both balanced and imbalanced datasets. IPSDM achieved significant improvements in classification metrics, including accuracy, precision, recall, and F1-score, by addressing class imbalance using adaptive synthetic sampling (ADASYN) and mitigating overfitting issues through advanced training techniques. The findings underscore the potential of LLMs to provide innovative and effective solutions to longstanding challenges in email security, such as phishing and spam detection.

Al-Subaiey et al. proposed a novel web-based platform for phishing email detection by integrating Explainable AI (XAI) techniques and machine learning models [32]. The study utilized six publicly available datasets, merging them into a single corpus of approximately 82,500 emails to enhance generalizability and robustness. The proposed platform employed TF-IDF for feature extraction and SVM for classification, achieving an F1-score of 0.99. Explainable AI techniques, such as LIME, were implemented to increase user trust by providing insights into model predictions. The platform was deployed as a user-friendly web application, enabling real-time phishing detection and allowing users to provide feedback for continuous model refinement. This study bridges the gap between high-performing models and their practical application, offering a scalable solution to combat phishing emails effectively.

B. *Motivation and Contribution*

Phishing attacks, particularly those targeting email users, continue to evolve, employing increasingly sophisticated tactics to deceive users. While existing studies have extensively utilized machine learning and deep learning techniques such as CNNs, LSTMs, and GRUs, they often overlook structural and linguistic features that can play a critical role in distinguishing phishing emails from legitimate ones. For example, CSS and HTML tags, black-listed words, and spelling or grammatical errors are common indicators of phishing emails that remain underexplored in the literature. This study aims to address this gap by introducing a novel feature set tailored to capture these overlooked characteristics. The main contributions of the study can be summarized as follows:

- We propose a set of innovative features, including counts of CSS and HTML tags, black-listed words, and grammatical errors, which significantly enhance the classification performance of phishing email detection systems.
- Our study demonstrates the effectiveness of combining these new features with traditional text representation methods (e.g., TF-IDF, Word2Vec) in improving model accuracy, precision, recall, and F1-score.
- We provide a detailed comparison with existing methods in the literature, highlighting that the incorporation of these features leads to state-of-the-art performance (e.g., achieving an F1-score of 99.53% with RF and TF-IDF).
- The proposed features and methods are validated on a real-world dataset, showcasing their potential for practical application in combating phishing threats.

C. *Organization*

The remaining parts of the study are organized as follows: Section II will discuss the classifiers, platforms, and methods used in machine learning and deep learning-based phishing email detection. Section III will address the proposed new features. Section IV will present and interpret the results obtained from the study. Finally, Section V will provide a general assessment and information regarding future studies.

II. Experimental Settings

This section will cover the programming language and libraries used, the dataset, data preprocessing steps, feature

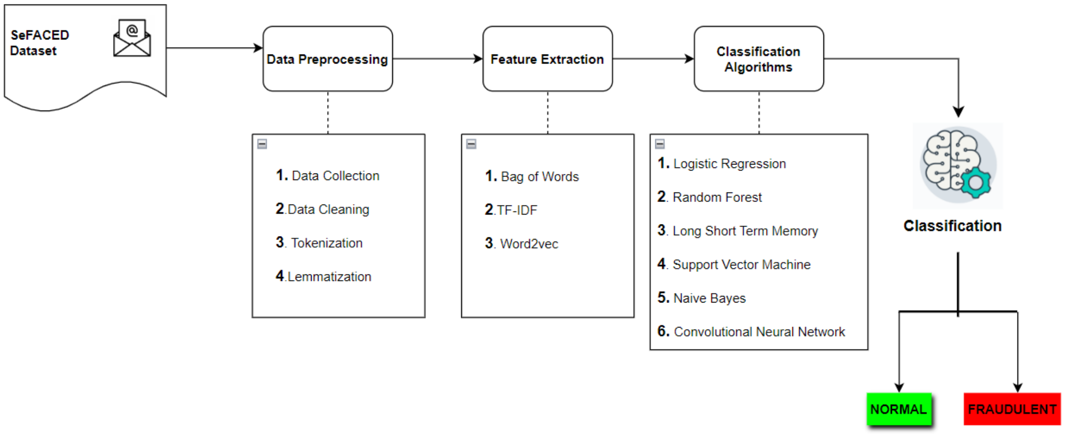

Fig. 1. Steps of Constructing the Proposed Model

extraction techniques, classification algorithms, and metrics used to evaluate classification performance. Figure 1 provides the general architecture of the proposed machine learning-based phishing email detection system.

A. Programming Language and Libraries Used

The models in this study were developed using the Python programming language. Various open-source libraries and tools were utilized to streamline and enhance the machine learning and deep learning processes. Libraries such as Scikit-learn (for implementing machine learning algorithms and data preprocessing), TensorFlow (for building and training neural network-based models), and Pandas (for data manipulation and analysis) played pivotal roles in model development. Additional libraries, including NLTK (for natural language processing tasks) and Numpy (for numerical computations and array operations), were employed to ensure robust data preprocessing and feature engineering. The development and experimentation were carried out on platforms like Google Colab and Jupyter Notebook, which facilitated efficient execution, debugging, and visualization of the results.

B. Dataset Used

The dataset used in the study is the SeFACED dataset [33]. This dataset is obtained by merging three different datasets. It contains 12498 normal, 5142 fraudulent, 19190 harassment, and 5323 suspicious emails. Each email's header information, such as sender and subject, has been removed. The normal emails in the SeFACED dataset are taken from the Enron Corpora, fake emails are from the Phished Emails Corpora, suspicious emails are from the Email Forensics dataset, and harassment emails are from the Hate Speech and Offensive dataset. 12498 normal emails and 5142 fraudulent emails were selected to create the dataset used in the study. 80% of the dataset was used for training, while 20% was used for testing. For a fair comparison, experiments were carried out by running the random_state parameter of the train_test_split module in the Sklearn library with the value 42 in all cases, since all algorithms must use the same training and the same test split.

C. Steps of Preprocessing

Email data needs to undergo a series of processing steps to be prepared for use in the model. The first stage is the data preprocessing stage. In this step, data collection, data cleaning, tokenization, and stemming processes are applied.

1) Data Collection: The data used is obtained from the SeFACED dataset, which is divided into 4 classes and consists of 42153 email texts [33]. However, in this study, binary classification is performed, so 12498 normal and 5142 fraudulent emails were used.

2) Data Cleaning: The data cleaning process involves removing punctuation marks, converting the text to lowercase, removing stop words from the text, removing links, numbers, special characters, and HTML tags, and cleaning up spaces. Some Python scripts have been used to perform these steps.

3) Tokenization: Email texts have been divided into smaller units to make the data more organized and manageable. In this step, the ”tokenize” method from the NLTK library has been used.

4) Stemming: Stemming is the process of reducing words in a text to their bases or roots. The ”stem” method from the NLTK library has been used to perform this operation.

D. Feature Extraction Techniques

Methods such as Bag of Words, TF-IDF, and Word2Vec were used to perform feature extraction.

1) Bag of Words: Bag of Words is a technique used to transform the words in each document of a dataset into a vector that represents their frequencies. The ”CountVectorizer” method from the Sklearn library has been used to create the Bag of Words.

2) TF-IDF: Term Frequency-Inverse Document Frequency (TF-IDF) is a technique used to represent both the frequencies of words and the importance of each word in a document. The ”TfidfVectorizer” method from the Sklearn library has been used to implement this technique.

3) Word2Vec: In natural language processing, Word2Vec is a method for obtaining vector representations of words. These vectors use the surrounding words to infer information about the word’s meaning. The Word2Vec algorithm models text in a large corpus in order to estimate these representations. In this study, the Word2Vec technique has been utilized using the Gensim library.

E. Classification Algorithms

Six machine and deep learning algorithms have been used to create models in this study. These are LR, Random Forest (RF), LSTM, Support Vector Machine (SVM), NB, and CNN.

1) Logistic Regression): LR is one of the popular classification algorithms and is mostly used in binary classification problems. It is a supervised machine learning algorithm used when the categorical dependent variable is discrete. The sigmoid function is generally used as the activation function. The dependent variable is usually a binary variable defined as 1 and 0 [34].

2) Random Forest: RF is fundamentally based on the principle of aggregating the predictions produced by many decision trees. This algorithm is generally used in classification and regression problems. It prevents overfitting errors that may occur in decision trees. There is a linear relationship between the number of trees in the algorithm and the classification result obtained [13].

3) Long Short Term Memory: In deep learning, LSTM is a frequently used recurrent neural network architecture designed to prevent long-term dependencies. It consists of gates that control the input or output of information to the relevant cell. The LSTM architecture is widely used in many areas, such as text and language processing, speech recognition, and handwriting recognition [35].

4) Support Vector Machine: SVM is frequently used in classification problems, but it is also a supervised learning algorithm used in areas such as clustering and anomaly detection. Essentially, this algorithm separates the data with a hyperplane, also known as the decision boundary, which is mainly used to separate data consisting of two classes [24].

5) Naive Bayes: The NB algorithm is based on Bayes’ theorem, frequently used in probability. This classifier is a commonly used supervised learning algorithm in machine learning. It works by calculating the probability of each possible outcome for a given data point and then performing classification based on the resulting probability values [24].

6) Convolutional Neural Network: CNN is a type of artificial neural network typically composed of input layers, convolutional layers, pooling layers, and fully connected layers. It is also a subfield of deep learning. In addition to the input and output layers, it has multiple hidden layers. It is a popular tool used in fields such as image processing, image and video recognition, and image classification [36].

F. Classification Performance Metrics

In this section, four performance metrics accuracy, recall, precision, and F1-score were used to evaluate the performance of the created models.

1) Accuracy: Accuracy is calculated as the ratio of the number of correctly predicted examples to the total dataset. It can also be referred to as the percentage of correctly classified data. The accuracy metric is given in Equation 1. The variables used in this metric and the other metrics are as follows:

- **TP:** True Positive. This refers to the case where the values predicted as positive are actually positive.
- **TN:** True Negative. This refers to the case where the values predicted as negative are actually negative.
- **FP:** False Positive. This refers to the case of predicting examples with a true negative value as positive.
- **FN:** False Negative. This refers to the case of predicting values as negative when they are actually positive.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

2) Precision: Precision indicates how many of the predictions identified as positive are actually positive. The precision metric is shown in Equation 2.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

3) Recall: Precision is the ratio of the number of true positive predictions to the total number of predictions made as positive. The mathematical representation of the precision metric is given in Equation 3.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

4) F1-score: The F1-score is a measure calculated by combining precision and recall metrics by taking the harmonic mean of these values. Particularly in imbalanced datasets, interpreting based solely on accuracy can be misleading. The F1-score can take values between 0 and 1, with higher values indicating better performance. The mathematical representation of this metric is given in Equation 4.

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

III. RECOMMENDED ATTRIBUTES FOR PHISHING EMAIL DETECTION

In the second part of the study, new features were proposed before the preprocessing stage to improve the classification performance. These features include the counts of HTML tags, CSS tags, black-listed words, links, language, and spelling errors in each email. These features address key gaps in previous studies, where contextual and structural characteristics of emails were often overlooked, focusing instead on content-based analysis. By integrating these features, the proposed model provides a more comprehensive approach to phishing email detection. This information was extracted from email texts and updated on a .csv file. These additional features were added as columns to matrices created using feature extraction methods.

Algorithm 1: Phishing Email Detection Feature Extraction

```
Input: Email dataset $D$, Feature extraction methods $F$
Output: Enhanced feature matrix $M$
Initialize an empty feature matrix $M$;
foreach email $e \in D$ do
    Extract raw text $T$ from email $e$;
    Compute structural features:;
    $HTML\_Count \leftarrow$ Count of HTML tags in $T$;
    $CSS\_Count \leftarrow$ Count of CSS tags in $T$;
    $Blacklist\_Words \leftarrow$ Count of black-listed words in $T$;
    $Link\_Count \leftarrow$ Number of links in $T$;
    $Language \leftarrow$ Detected language of $T$;
    $Spelling\_Errors \leftarrow$ Number of spelling errors in $T$;
    Append computed features to email's feature vector;
    Apply feature extraction methods $F$ to $T$;
    Add all extracted features as columns to $M$;
end
Update $M$ by saving the enhanced feature matrix to a .csv file;
return $M$;
```

Algorithm 1 defines a process for extracting new structural and content-based features to enhance phishing email detection. It calculates features such as the counts of HTML and CSS tags, black-listed words, number of links, language, and spelling errors from emails, integrating them into the existing feature matrix to improve classification performance.

The proposed features, such as counts of HTML tags, CSS tags, black-listed words, and grammatical errors, were designed to complement traditional text representation techniques like Bag-of-Words, TF-IDF, and Word2Vec. These features enrich the representation of emails by providing structural and linguistic information that is often overlooked in conventional approaches. By integrating these additional attributes, we aim to enhance the ability of classifiers to identify subtle distinctions between phishing and legitimate emails. This holistic feature representation ensures that the model leverages both semantic and structural information during the classification process.

The proposed features are seamlessly combined with text representation outputs to create a unified feature vector for each email. This vector incorporates the text-based features derived from methods like TF-IDF with the contextual cues provided by the novel attributes. As a result, the classification decision is made for the email as a whole rather than its individual segments, addressing potential challenges in judging emails composed of multiple text pieces. This integration improves the classifier's ability to generalize across varied phishing attempts and real-world email data, contributing to robust phishing detection performance.

A. *HTML Tags Count*

BeautifulSoup library was used to calculate the number of HTML tags in each email text, and this information was saved to the extra features .csv file.

B. *CSS Tags Count*

BeautifulSoup library was used to calculate the number of CSS ($< style >$) tags in each email text, and this information was saved to the relevant file.

C. *Black-list Words Count*

Commonly used phishing email keywords were collected and saved to a .txt file [37]. Subsequently, using a Python function, the number of occurrences of these keywords in each email in the dataset was calculated and saved to the extra features file.

D. *Links Count*

The number of links in each email in the dataset was calculated using a Python script containing a regular expression, and this information was saved to the extra feature file.

E. *Grammar Errors and Misspelled Words Count*

The ”Language_tool_python” library and the ”Enchant” module were used to calculate spelling and language errors in each email, and this information was saved to the extra features file.

IV. RESULTS AND DISCUSSIONS

In this section, the results obtained from the study will be presented. In Section IV-A, the classification results without using the proposed features will be provided. In Section IV-B, the classification results obtained by adding the proposed features will be presented. Finally, a comparison will be made by presenting the results obtained from the literature alongside the results obtained from this study in tabular form in IV-C.

TABLE I
RESULTS OBTAINED BEFORE ADDING NEW ATTRIBUTES

Model	Accuracy	Precision	Recall	F1-score
LR (Bag-of-Words)	0.9836	0.9834	0.9938	0.9886
RF (Bag-of-Words)	0.9866	0.9870	0.9943	0.9906
NB (Bag-of-Words)	0.9763	0.9758	0.9914	0.9835
SVM (Bag-of-Words)	0.9851	0.9858	0.9934	0.9895
LSTM (Bag-of-Words)	0.9851	0.9805	0.9675	0.9739
CNN (Bag-of-Words)	0.9851	0.9865	0.9614	0.9738
LR (TF-IDF)	0.9790	0.9743	0.9967	0.9854
RF (TF-IDF)	0.9880	0.9878	0.9955	0.9916
NB (TF-IDF)	0.9702	0.9602	0.9996	0.9795
SVM (TF-IDF)	0.9907	0.9894	0.9975	0.9935
LSTM (TF-IDF)	0.9863	0.9896	0.9624	0.9758
CNN (TF-IDF)	0.9851	0.9775	0.9706	0.9740
LR (Word2Vec)	0.9506	0.9595	0.9717	0.9656
RF (Word2Vec)	0.9836	0.9810	0.9863	0.9886
NB (Word2Vec)	0.8796	0.9575	0.8696	0.9114
SVM (Word2Vec)	0.9547	0.9623	0.9746	0.9684
LSTM (Word2Vec)	0.9707	1.0	0.8985	0.9465
CNN (Word2Vec)	0.9953	0.9899	0.9939	0.9919

TABLE II
RESULTS OBTAINED AFTER ADDING NEW ATTRIBUTES

Model	Accuracy	Precision	Recall	F1-score
LR (Bag-of-Words)	0.9860	0.9874	0.9930	0.9902
RF (Bag-of-Words)	**0.9930**	**0.9923**	**0.9979**	**0.9951**
NB (Bag-of-Words)	0.9752	0.9892	0.9758	0.9824
SVM (Bag-of-Words)	0.9886	0.9886	0.9955	0.9920
LSTM (Bag-of-Words)	0.9901	0.9779	0.9878	0.9828
CNN (Bag-of-Words)	0.9892	0.9897	0.9726	0.9811
LR (TF-IDF)	0.9822	0.9829	0.9922	0.9875
RF (TF-IDF)	**0.9933**	**0.9931**	**0.9975**	**0.9953**
NB (TF-IDF)	0.9328	0.9646	0.9401	0.9522
SVM (TF-IDF)	0.9924	0.9939	0.9955	0.9947
LSTM (TF-IDF)	0.9898	0.975	0.9898	0.9824
CNN (TF-IDF)	0.9860	0.9795	0.9716	0.9755
LR (Word2Vec)	0.9611	0.9702	0.9754	0.9728
RF (Word2Vec)	0.9883	0.9878	0.9959	0.9918
NB (Word2Vec)	0.8443	0.9644	0.8113	0.8813
SVM (Word2Vec)	0.9620	0.9726	0.9742	0.9734
LSTM (Word2Vec)	**0.9971**	**0.9939**	**0.9959**	**0.9949**
CNN (Word2Vec)	0.9921	0.9990	0.9736	0.9861

A. *Results Without New Attributes*

In the first stage of the study, machine learning and deep learning models were created without adding the extracted features to the bag-of-words, TF-IDF, and Word2Vec matrices. Table I presents all the results. Among the algorithms used, the SVM algorithm with TF-IDF feature extracting technique achieved the best performance with an accuracy of 99.07%, precision of 98.94%, recall of 99.75%, and F1-score of 99.35%.

The lowest performance among the algorithms used was obtained by the NB algorithm using Word2Vec, with an accuracy of 87.96%, precision of 95.75%, recall of 86.96%, and F1-score of 91.14%.

B. *Results Obtained by Adding New Attributes*

At this stage, the previously created matrices were augmented with additional features, such as the number of HTML tags and spelling errors, and these features were added as columns. The classification results with the addition of new features are provided in Table II. Among the machine learning algorithms, the RF algorithm using TF-IDF feature extraction technique achieved the best performance with an accuracy of 99.33%, precision of 99.31%, recall of 99.75%, and F1-score of 99.53%. The performance of this algorithm increased by almost 1% compared to its performance without the extra features.

The LSTM algorithm with word2Vec feature extraction technique showed the best performance among deep learning algorithms with an F1-score of 99.49% and an accuracy of 99.71%. It was observed that the accuracy performance increased by 2% compared to the algorithm's previous performance. These results demonstrate the significant improvement achieved by incorporating the proposed features, as the LSTM model's performance surpasses many state-of-the-art approaches highlighted in the literature, further validating the effectiveness of the proposed methodology.

Among the machine learning algorithms, the lowest performance was obtained by the NB algorithm using Word2Vec, with an accuracy of 84.43%, precision of 96.44%, recall of 81.13%, and F1-score of 88.13%.

When the proposed features that distinguish phishing email attacks from normal emails are generally evaluated, it is seen

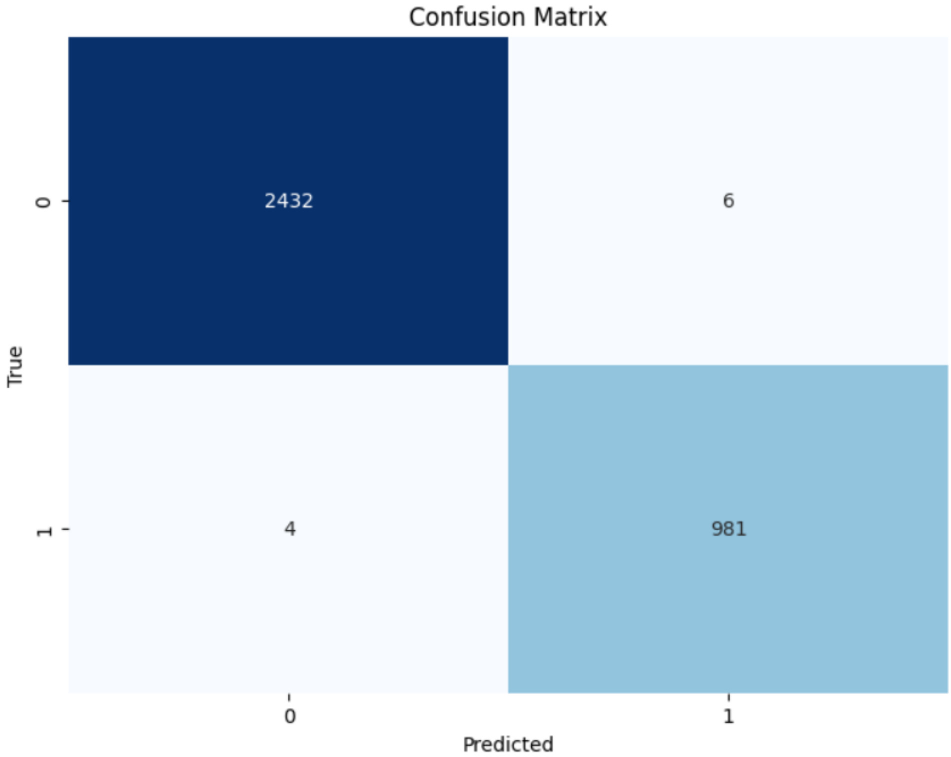

Fig. 2. Confusion Matrix Obtained by LSTM Algorithm After Adding the New Features

that the classification performance of machine learning and deep learning algorithms mostly increases. Performance decreases when Word2Vec text representation and NB algorithm are used. In tests performed without adding the recommended features, lower performance was achieved compared to other cases. The most important reason for this is Word2Vec text representation. Because Word2Vec has a more complex structure for the NB algorithm compared to TF-IDF and Bag of Words text representation.

When Word2Vec text representation and LSTM network are used together with the proposed features, the highest classification performance is achieved among all the experiments. According to the accuracy metric, this result is reported as 0.9971. The complexity matrix for this experiment is given in Figure 2. There are actually 2438 normally labeled samples included in the test data. Only 6 of these samples are misclassified. On the other hand, 4 out of 985 fraudulent samples in the test data are misclassified. LSTM incorrectly predicts the labels of 10 samples in total. Considering all the experiments, this result is the highest performance result achieved in terms of accuracy metric. The contribution of the proposed features to the classification performance comes to the fore with this experiment.

C. Comparison of Results Obtained from Existing Studies

In this section, a comparison will be made between the results obtained from other studies and the relevant study. The results of existing studies are provided in Table III, which includes the results of articles related to phishing. The results obtained with various classification algorithms have been evaluated according to the relevant performance metrics. Among the machine learning algorithms, the study conducted by Livara et al. [12] achieved the highest performance ratio. The algorithm used in their study was RF, and it achieved the highest performance with an F1-score of 99.4%. Among the deep learning algorithms, the study conducted by Dewis et al. [14] achieved the highest ratio. In [14], the LSTM algorithm achieved a success rate of 99% based on the F1-score. The results of both studies are lower than the results of the proposed model. When the text representation obtained by adding the recommended extra features is given to RF with TF-IDF features extraction technique, 99.53% performance is achieved.

In the conducted study, the addition of extra features improved the classification performance in general. For example, the classification performance reached a 99.02% F1-score with the LR algorithm using the Bag of Words feature extraction technique. However, it was observed that the performance of the NB algorithm decreased slightly when extra features were added compared to the classification performed without adding extra features.

V. GENERAL EVALUATION AND DISCUSSIONS

According to researches, many phishing attacks occur via email, hence the aim of classifying phishing attacks using machine and deep learning algorithms. A comprehensive literature review of 18 articles was conducted in the study. The dataset used is the SeFACED dataset, consisting of 12,498 legitimate and 5,143 phishing email data. During model creation, the data underwent preprocessing, which is crucial for

TABLE III
COMPARISON WITH EXISTING STUDIES

Study	Dataset Size	Used Method	Performance
Livara and Hernandez [12]	525,754 emails	RF	99.4% (F1-score)
Akinyelu and Adewumi [13]	2,000 emails	RF	98.45% (F1-score)
Paradkar [11]	8,336 phishing, 11,664 normal emails	CNN	98.26% (F1-score)
Ahi and Soğukpınar [9]	2,256 secure, 2,256 phishing emails	H-OLTA (Hybrid MLP-LSTM)	96% (F1-score)
Ahammad et al. [28]	Not specified	NB	96% (Accuracy)
Fayoumi et al. [24]	Not specified	SVM	99.80% (F1-score)
Dewis and Viana [14]	6 different datasets	LSTM	99% (Accuracy)
Abdulraheem et al. [10]	11,000 websites, 2,500 phishing emails	LMT	96.9% (Precision)
Proposed Method	**5,142 phishing, 12,498 normal emails**	**RF (TF-IDF with extra features)**	**99.53% (F1-score)**

normalizing the data and removing duplicate entries. The normalized data was digitized using feature extraction techniques to be utilized in the model. The algorithms used are LR, RF, LSTM, SVM, NB, and CNN. The performance metrics used to evaluate the models are F1-score, accuracy, precision, and recall values. In the second part of the study, the results of classification performance with and without additional features were analyzed in detail. Overall, the classification performance significantly improved when additional features were added. In the classification using Bag-of-Words with additional features, the algorithm that showed the highest increase in classification performance according to the F1-score was LSTM, with a rate of 98.28%. Similarly, in the classification using TF-IDF with additional features, LSTM showed the highest increase in classification performance with a rate of 98.24%. In the classification using Word2Vec with additional features LSTM showed the highest increase in classification performance with a rate of 99.49%.

REFERENCES

[1] R. Alabdan, “Phishing attacks survey: Types, vectors, and technical approaches,” *Future internet*, vol. 12, no. 10, p. 168, 2020.

[2] K. L. Chiew, K. S. C. Yong, and C. L. Tan, “A survey of phishing attacks: Their types, vectors and technical approaches,” *Expert Systems with Applications*, vol. 106, pp. 1–20, 2018.

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

[4] M. Jakobsson, “Two-factor inauthentication–the rise in sms phishing attacks,” *Computer Fraud & Security*, vol. 2018, no. 6, pp. 6–8, 2018.

[5] A. Basit, M. Zafar, X. Liu, A. R. Javed, Z. Jalil, and K. Kifayat, “A comprehensive survey of ai-enabled phishing attacks detection techniques,” *Telecommunication Systems*, vol. 76, pp. 139–154, 2021.

[6] APWG, “Apwg phishing activity trends report,” 2025. [Online]. Available: https://apwg.org/trendsreports

[7] S. Gupta, A. Singhal, and A. Kapoor, “A literature survey on social engineering attacks: Phishing attack,” in *2016 International Conference on Computing, Communication and Automation (ICCCA)*, 2016, pp. 537–540.

[8] J. Rastenis, S. Ramanauskaitė, J. Janulevičius, A. Čenys, A. Slotkienė, and K. Pakrijauskas, “E-mail-based phishing attack taxonomy,” *Applied sciences*, vol. 10, no. 7, p. 2363, 2020.

[9] Ş. Ahi and İ. Soğukpınar, “Derin öğrenme modelleri ile kimlik avı e-posta tespiti,” *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi*, vol. 13, no. 2, pp. 17–31, 2020.

[10] R. Abdulraheem, A. Odeh, M. Al Fayoumi, and I. Keshta, “Efficient email phishing detection using machine learning,” in *2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE, 2022, pp. 0354–0358.

[11] N. S. Paradkar, “Phishing email’s detection using machine learning and deep learning,” in *2023 3rd International Conference on Advances in Computing, Communication, Embedded and Secure Systems (ACCESS)*. IEEE, 2023, pp. 160–162.

[12] A. Livara and R. Hernandez, “An empirical analysis of machine learning techniques in phishing e-mail detection,” in *2022 International Conference for Advancement in Technology (ICONAT)*. IEEE, 2022, pp. 1–6.

[13] A. Akinyelu and A. Adewumi, “Classification of phishing email using random forest machine learning technique,” *Journal of Applied Mathematics*, vol. 2014, 2014.

[14] M. Dewis and T. Viana, “Phish responder: A hybrid machine learning approach to detect phishing and spam emails,” *Applied System Innovation*, vol. 5, no. 4, p. 73, 2022.

[15] X.-W. Chen and X. Lin, “Big data deep learning: challenges and perspectives,” *IEEE access*, vol. 2, pp. 514–525, 2014.

[16] M. Coşkun, Ö. Yıldırım, A. Uçar, and Y. Demır, “An overview of popular deep learning methods,” *European Journal of Technique (EJT)*, vol. 7, no. 2, pp. 165–176, 2017.

[17] M. K. Sharma, R. Kumar, D. K. Sinha, K. Senthilkumar, D. Dhabliya, and G. Ahluwalia, “Exploring the benefits of deep learning for data science practices,” in *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*. IEEE, 2024, pp. 1–7.

[18] E. E. Eryilmaz, D. O. Şahin, and E. Kılıç, “Machine learning based spam e-mail detection system for turkish,” in *2020 5th International Conference on Computer Science and Engineering (UBMK)*. IEEE, 2020, pp. 7–12.

[19] S. T. Singh, M. D. Gabhane, and C. Mahamuni, “Study of machine learning and deep learning algorithms for the detection of email spam based on python implementation,” in *2023 International Conference on Disruptive Technologies (ICDT)*. IEEE, 2023, pp. 637–642.

[20] B. Sonare, G. J. Dharmale, A. Renapure, H. Khandelwal, and S. Narharshettiwar, “E-mail spam detection using machine learning,” in *2023 4th International Conference for Emerging Technology (INCET)*. IEEE, 2023, pp. 1–5.

[21] A. A. Adzhar, Z. Mabni, and Z. Ibrahim, “A comparative study on email phishing detection using machine learning techniques,” in *2022 IEEE International Conference on Computing (ICOCO)*. IEEE, 2022, pp. 96–101.

[22] B. B. Gupta, K. Yadav, I. Razzak, K. Psannis, A. Castiglione, and X. Chang, “A novel approach for phishing urls detection using lexical based machine learning in a real-time environment,” *Computer Communications*, vol. 175, pp. 47–57, 2021.

[23] N. Moradpoor, B. Clavie, and B. Buchanan, “Employing machine learning techniques for detection and classification of phishing emails,” in *2017 Computing Conference*. IEEE, 2017, pp. 149–156.

[24] M. Al Fayoumi, A. Odeh, I. Keshta, A. Aboshgifa, T. AlHajahjeh, and R. Abdulraheem, “Email phishing detection based on naïve bayes, random forests, and svm classifications: A comparative study,” in *2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE, 2022, pp. 0007–0011.

[25] F. Salahdine, Z. El Mrabet, and N. Kaabouch, “Phishing attacks detection a machine learning-based approach,” in *2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*. IEEE, 2021, pp. 0250–0255.

[26] Y. Wei and Y. Sekiya, “Sufficiency of ensemble machine learning methods for phishing websites detection,” *IEEE Access*, vol. 10, pp. 124 103–124 113, 2022.

[27] A. K. Jain and B. B. Gupta, “A machine learning based approach for phishing detection using hyperlinks information,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, pp. 2015–2028, 2019.

[28] S. M. M. Ahammad, T. Raviteja, J. Koushik, P. V. Dinesh, and A. Ashok, “Machine learning approach based phishing email text analysis (ml-peta),” in *2022 Third International Conference on Intelligent Computing*

Instrumentation and Control Technologies (ICICICT). IEEE, 2022, pp. 1087–1092.

[29] C. Thapa, J. W. Tang, A. Abuadbba, Y. Gao, S. Camtepe, S. Nepal, M. Almashor, and Y. Zheng, "Evaluation of federated learning in phishing email detection," *Sensors*, vol. 23, no. 9, p. 4346, 2023.

[30] P. N. Wosah, Q. Ali Mirza, and W. Sayers, "Analysing the email data using stylometric method and deep learning to mitigate phishing attack," *International Journal of Information Technology*, pp. 1–14, 2024.

[31] S. Jamal, H. Wimmer, and I. H. Sarker, "An improved transformer-based model for detecting phishing, spam and ham emails: A large language model approach," *Security and Privacy*, p. e402, 2024.

[32] A. Al-Subaiey, M. Al-Thani, N. A. Alam, K. F. Antora, A. Khandakar, and S. A. U. Zaman, "Novel interpretable and robust web-based ai platform for phishing email detection," *Computers and Electrical Engineering*, vol. 120, p. 109625, 2024.

[33] M. Hina, M. Ali, A. R. Javed, F. Ghabban, L. A. Khan, and Z. Jalil, "Sefaced: Semantic-based forensic analysis and classification of e-mail data using deep learning," *IEEE Access*, vol. 9, pp. 98 398–98 411, 2021.

[34] AWS-Blog, "Lojistik Regresyon Nedir? - Lojistik Regresyon Modeline Ayrıntılı Bakış," 2025. [Online]. Available: https://aws.amazon.com/tr/what-is/logistic-regression/

[35] E. Gavcar and H. M. Metin, "Hisse senedi değerlerinin makine öğrenimi (derin öğrenme) ile tahmini," *Ekonomi ve Yönetim Araştırmaları Dergisi*, vol. 10, no. 2, pp. 1–11, 2021.

[36] H. Li, "Computer network connection enhancement optimization algorithm based on convolutional neural network," in *2021 International Conference on Networking, Communications and Information Technology (NetCIT)*. IEEE, 2021, pp. 281–284.

[37] A. Onar, "English Spam Words List," 2025. [Online]. Available: https://github.com/OOPSpam/spam-words/blob/main/spam-words-EN.txt

BIOGRAPHIES

Hadjer Brioua has been an undergraduate student at Ondokuz Mayıs University, Faculty of Engineering, Department of Computer Engineering since 2019. She is currently pursuing the B.Sc. degree in Computer Engineering. Her research interests include machine learning, text mining, and deep learning.

Havvanur Siyambaş has been an undergraduate student at Ondokuz Mayıs University, Faculty of Engineering, Department of Computer Engineering since 2020. She is currently pursuing the B.Sc. degree in Computer Engineering. Her research interests include machine learning, text mining, and deep learning.

Durmuş Özkan Şahin received a Bachelor's degree in Computer Engineering from Süleyman Demirel University Isparta in 2013 and a Master's degree in Computer Engineering from Ondokuz Mayıs University Samsun in 2016. Finally, he received a PhD's degree in Computational Sciences from Ondokuz Mayıs University Samsun in 2022. His research interests include machine learning, text mining, information retrieval, and Android malware analysis. He is currently an Assistant Professor of the Department of Computer Engineering at Ondokuz Mayıs University.