# AI-Based Phishing Detection System Using Random Forest with Smote and Explainability

Ansar Mehmood [1], Sughra Fatima Manzoor [2], Altaf Khan [3], Asif Raza[3]
[1]MS student in Data Science Intelligence, Canterbury Christ Church University, UK,
[2]Student at the Department of Software Engineering, University of Mianwali, Pakistan
[3]Faculty member, Department of Software Engineering, University of Mianwali, Pakistan
*Correspondence: am2461@canterbury.ac.uk

**Citation |** Mehmood. A, Manzoor. S. F, Khan. A, Raza. A, "AI-Based Phishing Detection System Using Random Forest with Smote and Explainability", IJIST, Vol. 8 Issue. 5 pp 2371-2390, August 2026
**DOI:** https://doi.org/10.33411/IJIST/1964
**Received |** July 20, 2026 **Revised |** Aug 18, 2026 **Accepted |** Aug 26, 2026 **Published |** Aug 30, 2026.

One of the most widespread and destructive forms of Cyber Security breach is the Phishing Attack. Phishing attacks often defeat email filtering systems due to their heightened use of social engineering techniques that by-pass traditional filtering mechanisms. This research employs a Random Forest classifier, alongside TF-IDF based feature extraction methods, to classify Emails. The major innovation of this research is the implementation of SHAP (SHapley Additive exPlanations) technology in relation to the output of the Random Forest to provide explanations of each prediction in an understandable and user-friendly manner. Users may access this system via a simple copy-and-paste interface or uploading a file to submit email content for real-time analysis. The results of the analysis will consist of prediction (Phishing/Safe), confidence rating, level of risk, and a detailed listing of each risk factor as calculated using SHAP based explanations. The Random Forest model will have an accuracy of 96.12% with 94.4% precision and a recall rate of 95.87%, significantly better than the results generated using traditional rule-based methods. Response times for Email analysis averaged 3 to 5 sec. Combining high detection rates with explainable AI improved Cyber Security Awareness, while also offering practical and easily accessible protection from Phishing attacks for individual.

**Keywords:** Phishing emails, Random Forest, NLP, Cyber Attack, SHAP, Phishing Detection System, Cyber Attack, Email phishing, SMOTE, Feature Extraction, Social Engineering, XAI

**Introduction:**

Phishing attack is most common way of manipulating humans to theft the sensitive information like the password credential, bank details etc. The use of emails is still very popular among people either official notices by company, online shopping emails, bank emails, applications emails, password reset emails etc. according to a report, the rate of emails sent per day is around 3.4 Billion. A phishing attack is a social engineering attack that involves an attacker using an email purporting to be from a legitimate organization to convince the recipient to divulge sensitive information like usernames and passwords, banking information, or other personal details or to click on links that lead to malicious sites. Phishing attacks using e-mail accounted for 73% of all social engineering attacks, according to Verizon 2024 Data Breach Report. The financial loss is severe in result of the phishing. In 2024 Report, $2.77 Billion loss was encountered. Phishing and spoofing is the top reported crime incident of the FBI's Crime Complaint Center [1].

Despite years of anti-phishing measures, the Anti-Phishing Working Group (APWG) is still seeing 3.8 million phishing attacks per year worldwide in 2025. The highest number of attacks since APWG began its work in 2003, the attackers are increasingly exploiting generative AI, QR codes, and techniques breaking multi-factor authentication, instead of the poorly crafted baits they used to exploit [2]. According to data collected in 2024, it was revealed that close to 75% of organizations all over the world reported phishing attempts with more than 60% of the attacks occurring via email [3]. Traditional Rule based and back list filters are too simple to catch phishing as the attacker exploits the sophisticated ways and social engineering and also leverages the AI so that their fake email can't be detected by simple filters and traditional ways. The supervised ML methods for email phishing detection have spread widely, with Naïve Bayes, SVM and Random Forest (RF) demonstrating show good performance for training using the TF-IDF of email text. Compared to these models, RF shows a significant level of accuracy when it comes to phishing detection, as some studies claim high accuracy using RF model [4].

To recognize the fraudulent phishing emails, machine learning methods are being used but that was not enough as they didn't provide the transparency and any explanation. There was a strong need that any machine learning model that is detecting phishing attack should have to integrate the Explainable Artificial Intelligence to provide the explanation of the model. By combining both the Machine learning technique and XAI (SHAP (SHapley Additive exPlanations) or LIME) or any other, a greater level of assistance can be provided in identifying phishing attacks.

These are the ways to use transparency and explain ability in predictive models. SHAP is based on the game theory and it is an emerging method, it constructs the simple explanation model with feature attribution [5]. Random Forest Algorithm shows its potential for classifying the data by providing the accurate classifications of large and well balanced sets of sample data. This research presents that using a large set of training data gives Random Forest's accuracy as classifier at high rate of classification. This indicates that the quality and size of the data is important in fine tuning the outcome of the NLP based feature extraction procedure [6]. This AI-based email phishing detection system combines NLP for textual feature extraction, a Random Forest classifier for prediction, and SHAP for interpretability.

The following is an example from the research literature examining actual applications of AI-based phishing detection. The example used in this literature consists of phishing message sent via e-mail impersonating Microsoft. In that particular instance, the e-mail initiated the recipient to use a "verify" button to confirm an updated user agreement for their Microsoft account. After clicking the verify button, the recipient was directed toward a fraudulent Web page. This fraudulent URL was associated with an actual domain name and had been identified as malicious by various security vendors. One of the ways that the AI-

based detection system determined that the e-mail was a phishing attack was through a linguistic consistency test, specifically, the fact that the sender of the e-mail identified themselves as a "member of the Microsoft Account Team," but their domain name was not associated with Microsoft. This major structural and linguistic inconsistency was detected by the model [7].

A phishing email attempts to trick a user into taking an action by creating an atmosphere of urgency, impersonating a legitimate company or providing a link that is not what it appears. As such, traditional spam filters do not work for detecting phishing email because these types of emails contain different patterns that typically identify them. Within the last several months, researchers have developed a model that uses AI-based techniques. This model utilizes natural language processing techniques, such as word2vec with TF-IDF (Term Frequency-Inverse Document Frequency) weights to produce feature vectors from the textual content of phishing emails, in combination with machine-learning classification techniques. An easy to use web application has been created to allow for users to locate and identify phishing email quickly and efficiently [8] and other different fraud techniques [9]. Moreover, the approaches based on deep learning architecture, for example, LSTM-CNN, achieve high accuracy on large datasets but need expensive resources to operate, do not have any inherent explainability, and may get overfitted when operating with small datasets. Traditional machine learning methods, including SVM and Logistic Regression, have good computing power, but do not deal with the problem of class imbalance that is crucial in the context of phishing databases, where the number of legitimate emails is much higher than that of phishing ones. In addition to this, most of the existing studies do not provide an explanation of the classifier operation that limits the understanding of the results by the analysts.

Taking into account the challenges presented above, this paper proposes an innovative email phishing detection system that operates with the issues of class imbalance, extraction of semantic features, and understanding of the model all at once and in the course of the same operation. In opposition to the methods mentioned earlier that rely on the use of the above techniques separately, this model employs multi-level NLP preprocessing that includes TF-IDF vectorization, SMOT(Synthetic Minority Over-sampling Technique), and the random forests technique based on Explainable AI (SHAP). Furthermore, the all five techniques are applied in relation to the email phishing detection based on the three datasets: Enron, Spam Assassin, and Nazario.

**Research Contribution:**

The main contributions of this research are outlined as follows: First, this manuscript provide a detail analysis of the state-of-the-art models and suggest the model on the base of comparative of these models. Secondly, the paper suggests a robust model comprising a unified phishing detection system, which employs a combination of text preprocessing by means of NLP, TF-IDF vectorization, SMOTE as a technique for balancing the class distribution, and classification using Random Forest. Third, the study integrates an Explainable AI method based on SHAP technique, which gives both global and local level explanations of the performance of the model, solving thus a problem of interpretability in the current ensemble-based methods of phishing attacks detection. In last, the results of the conducted study demonstrate that the combination of structured preprocessing with NLP and SMOTE is the most significant performance factor that operates irrespective of the classifier used. The new model surpasses four most recent related studies that utilized the same experimental conditions.

The proposed model outperforms the performance of other four similar recent studies by all scientific evaluation criteria under similar experimental conditions including [10], [11]since all of these studies used the similar benchmarking dataset but performance of the suggested model in terms of accuracy, precision, recall and F1-score goes higher because they

did not focus on SMOTE-based class balancing and SHAP-based explainability methods. Some other approaches such as [12], [13], use the combined classifiers to get the better performance, while these models become more complex and need much time to train them.

**Research Objectives:**

This work will have five main objectives:

Developing structured NLP preprocessing pipeline for the text cleaning of phishing emails, and feature extraction.

Applying SMOTE oversampling to mitigate class imbalance in the combined phishing benchmark dataset.

Using a random forest classifier based on TF-IDF features for binary phishing email classification.

**Employing SHAP for model interpretability:**

Performing an assessment of the suggested framework against the latest achievements using the known performance metrics such as accuracy, precision, recall, and f1-score.

**Related work:**

One of the earliest work that considered applying the RF to the classification of phishing email was content-based features utilization to classify phishing email. Firman et al. [14] conduct a comprehensive review of Phishing Email Classification Using Random Forest Machine Learning" that provide a baseline knowledge and use of the machine learning in phishing email classification task. Likewise, the PILFER is one of the early technique in which concept of feature extraction with Random Forest for phishing detection is used. In this method 10 representative phishing email features are used and the Random Forest has been considered as a classifier by building several decision trees. The model achieves good accuracy on the phishing dataset [15]. [10] introduced plus tested 3 types of learning models, namely Random Forest, SVM and XGBoost for detecting phishing emails in the study. They applied the feature selection method PCA and classifiers RF and SVM; they got the highest accuracy rate of 97.6% for the best model and they found that the classifiers RF and SVM attained the highest overall correctness rate, correctness of positive predictions, True Positive Rate and F-Score value [10].

The study conducted by [16] entailed comparison of various machine learning algorithms that are used for phishing detection in emails. In the study, authors used three different approaches for information extraction. These approaches included term frequency-inverse document frequency (TF-IDF), word2vec and the Bidirectional Encoder Representations from Transformers (BERT). The results showed that among the tools used were logistic regression, decision-tree classifier, randomized forest classifier and multilayered perceptron, of which multilayer perceptron proved the most effective when working with the TF-IDF method having achieved precision, recall, F1-score and accuracy of 0.98. However, the study has not applied any of the methods related to handling binary class imbalance such as SMOTE, and no method or framework of interpreting model's reasoning was used.

[17] applied machine learning with Explainable Artificial Intelligence techniques to create more interpretable and transparent models for phishing detection. The authors worked with several well-known machine learning methods such as XGBoost, LightGBM, random forest, k-nearest neighbors, twin SVM, and CNN, which were tested on four public datasets, with the highest accuracy of 99.65% achieved by the XGBoost method. However, while achieving such high results, this research focused on using URL and structure features, unlike the proposed email body analysis based on TF-IDF [17].

A more recent comparison, such as [18] added a machine learning based, AI-based email classification system, SecureInbox. The four classical ML methods used were compared and it was found that the RF model was very effective in separating the legitimate and phishing e-mail data.

In another study, [19] show the analysis of the machine learning model for phishing URL detection and achieve high performance using the SVM, RF and gradient boosting. It achieves 98.27% highest accuracy using Extreme Gradient Boosting algorithm on URL based datasets. However, model not implicitly suggest the email phishing attacks.

[20] proposed a CNN-LSTM and random forest-based hybrid system for phishing URL detection. CNN-LSTM was found to work well when trained with character-level sequence features, and these extracted features, such as domain, address bar, HTML and JavaScript attributes were found to be suitable for training the RF model.

For detecting phishing emails, research on the ML-based techniques has been ongoing for many years. Researchers using ML to identify spam email messages was started in 2004, and it is still growing. [21] demonstrated that a web-based spam detector developed using only 10 features (the number of URLs, special characters, and exclamation points) by an ML algorithm was able to identify through a 97% True Positive Rate (TPR) the importance of the features for detecting phishing.

Dynamic feature selection was discussed in research done by [22], where weights were determined for the features identifying phishing, which will differ depending upon what the email message contains and does not contain. (e.g., A count of URLs in an email message would have more weight for being classified as phishing based upon the number of links contained as opposed to an email message with very few or no links). Additionally, urgency associated with a word is generally a feature in the presence of an email that is text-only deficient in identifying phishing messages.

Phishing research is now focused on the use of content analysis and using combined learning techniques to better detect and classify phishing emails. For example, [23] illustrate that combining both linguistic and structural features of phishing emails into ensemble learning techniques allows for an effective detection method, which achieved a 98.7% accuracy rate in detecting phishing attempts.

[24] proposed a new model that uses natural language processing techniques to classify phishing emails based on the content analysis of email messages sent to users by attackers. The model is based on a transformer-based neural network (BERT), and has produced an impressive success rate of 96.5% in classifying phishing emails through semantic analysis of the contents of phishing messages sent via email. The model identified phishing messages using the three main manipulation techniques associated with phishing emails: urgency, impersonation, and emotional appeal.

[25] examined phishing attack detection using writing style and psychology techniques in their study published in Computers and Security; researchers found that phishing attack detection increased by 34% when differentiating between phishing addresses and legitimate email addresses.

[26] studied an overall of 100 academic papers concerning Natural Language Processing (NLP)-based methods to find and identify Phishing emails, as discussed in the literature review. In the course of this literature review, Support Vector Machines (SVM) were determined to be the most predominant machine learning algorithm being used by researchers to detect Phishing emails, along with other classifications of machine learning methods being used. However, the majority of researchers used SVM as their classification method. With regard to the NLP techniques that were employed by researchers when utilizing NLP techniques to detect phishing emails, TF-IDF and word embedding were the most commonly used techniques. The most commonly cited dataset that was used in the development of NLP techniques for Phishing detection was the Nazario Phishing Corpus and the primary programming language that most researchers used to determine if an email was Phishing or not was Python [26].

**Methodology:**

This section introduces suggested research method. Following Figure 1, this method follows the supervised machine learning pipeline to categorize e-messages either as phishing or legitimate. Predictive algorithm follows a series of steps which includes Dataset Selection, Data Pre-Processing, Model Training, Out of Sample Classification & prediction, and experimental setting and results. For classification task, the Random Forest Algorithm is selected for the Email categorization due to its superiority over other classifiers in phishing detection.

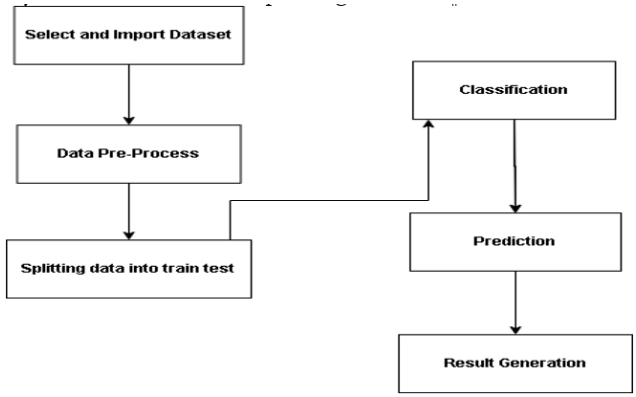



**Figure 1**. Data Flow of the Proposed Method.

**Implementation Details:**

The experiments are conducted on the visual studio framework with python 3.10 and python 3.11 version. The Table 1 shows the libraries that are used in this project implementation.

**Table 1.** Libraries and Versions.

| | |
|---|---|
| Scikit-learn | v1.3 |
| NLTK | v3.9 |
| Imb-learn | v0.11 |
| Shap | v0.42 |
| Pandas | v2.1 |
| Numpy | v1.26 |
| Bcrypt | v5.0 |

Stemming was performed using the porter stemmer and Stop words removal and Cleaning was done using the NLTK library of the NLP.

**Dataset Selection:**

A Corpus consist of a systematic arrangement of the data collected and is in the form of rows & columns. The collected set of data can be in the form of .csv or .JSON or excel file & is typically used to carry out various types of testing of AI models.

Two different datasets are collected and both of them are combined to get the new more robust challenging dataset. The features containing the phishing are extracted to train the model. The first data [27] is published in 2008. It contains approximately 39,154 emails, in which 17312 are safe while 21842 are spam emails. This dataset is originally created for the Conference on Email and AntiSpam 2008. It contains a balanced distribution of safe and phishing emails. Each email contains metadata like sender and receiver information, subject, body, label (0, 1) & URLS. This dataset combines the raw input as of the preliminary data in an exclusive instance for evaluation [27]. The dataset contains email subject lines and email body text in both ASCII and UTF-8 encoding. It should be noted that since the corpus was prepared in 2008, it reflects the attributes of phishing email prevalent in that year and does not reflect the modern email spams. Hence this drawback is overcome in the present study by using the Nazario phishing dataset [28][29]. This file contains the columns: email_text and email_type (safe, phishing) and approximately total 28,748, in which 11274 are safe and 7315

are phishing emails. This dataset contains the collection of emails collected from different university websites and public data. Some emails are collected from the Enron email dataset. This dataset combines the email's body and type that whether an email is safe or not [30].

The two datasets were merged due to the following reasons, including the first being the increased size of the dataset, as the larger dataset provides the ability to learn various phishing patterns. Also it reduces the overfitting of the model. The second reason is class balanced to improve by combining the datasets, making sure that the AI model should be neutral without any favoritism. Third is increased data diversity. The CEAS dataset contains spam and safe emails from year 2008, while Phishing Email dataset contains more recent patterns of email samples. Combining both of these will covers linguistic and structural diversity in training of the model making it more accurate.

The Nazario Phishing Corpus includes a total of 7,315 phishing emails that were collected over many years and were made publicly available in 2005, as detail of the dataset is shown in Table 2. The sources from which the emails were obtained are very diverse and make use of a number of phishing strategies such as credential theft and use of malicious links. The phishing samples in the Phishing_Email.csv file are derived from the Nazario Phishing Corpus but are interspersed with legitimate emails from a dataset of about 500,000 internal messages from Enron Corporation's employees collected from Kaggle. This combined dataset includes many types of emails and spans a wide variety of email content and different formats, such as plain text, HTML and such other email attachments [31][32][33].

**Table 2**. Characteristics of the Nazario+Enron Dataset.

| Characteristics | Detail |
|---|---|
| Full Name | Nazario Phishing Corpus + Enron Email Dataset |
| Phishing Source | Nazario (collected 2005–2022) |
| Legitimate Source | Enron Email Corpus (Kaggle) |
| Phishing Emails | ~7,315 |
| Legitimate Emails | Subset selected for balance |
| Format | CSV (email body + label) |
| Label | 1 = Phishing, 0 = Legitimate |
| Content | Full email body text |
| Language | English |

When two datasets are used in the project together, the diversity of the emails used for the task increases greatly. The detail of the merged dataset is shown in Table 3. Additionally, the datasets are all obtained from trusted sources that are well known for providing publicly available datasets that are used to detect phishing. The importation of the datasets into the detection tool was done using the Pandas data analysis library for Python, which provides data structures specifically designed to allow for efficient manipulation of large datasets. All datasets were stored as individual CSV files, but then combined into one single dataset. The combined datasets served as the foundation for all pre-processing and modeling activities to be conducted on the dataset.

**Table 3.** Merged Dataset.

| | |
|---|---|
| Total Emails in Combined Dataset | ~47,000+ (after merging CEAS-08 + Phishing_Email) |
| After Cleaning | ~36,000 emails |
| Training Set (80%) | ~28,800 emails |
| Testing Set (20%) | ~7,200 emails |

**Experimental Results:** The final stage is creating detailed outcomes that can be used to analyze accuracy of model and also to obtain actionable insights. The output of model's predictions is then compared against actual labels from test data for various performance metrics that give an overall calculation of how well model performed at detecting phishing emails. A confusion matrix is also generated allowing a more detailed analysis of the model's

strengths and weaknesses. The output of each prediction by the system is a structured output, as well as providing both the confidence score and risk level, along with a list of risk factors. The SHAP (SHapley Additive exPlanations) explanations are provided to summarize the features that most influenced the prediction result.

**Feature Extraction:**

To build effective email filtering system, there is a strong need to define the features related to phishing attacks. The following categories are most widely used in phishing emails. URL-based, HTML-Based and Content-Based features. URL-based features contain unseen domain, IP addresses in URL and using URL shorteners. HTML-based features contain hidden elements, invisible data, external links and HTML forms. Content-based features contains keywords like Urgency, Time Pressure, Account Threats, Verification, Suspicious claims, Prizes, Click Request, Brand Impersonation, Job Scams, Delivery scams etc [10].

**Data Pre-Processing:**

As the dataset is in the raw form and it contains too much noise, it must be ore processed first in a desired format (steps are shown in Figure 1). The raw email contains stopwords, numbers, links, urls, email addresses etc. The preprocessing steps are shown in Figure 3. Then it is splitted into training data of 80% & Testing data of 20%. Pre-processing is the step where raw emails are cleaned and make them suitable for the model [34]. The data pre-processing pipeline is shown in Figure 2 and Figure 3.

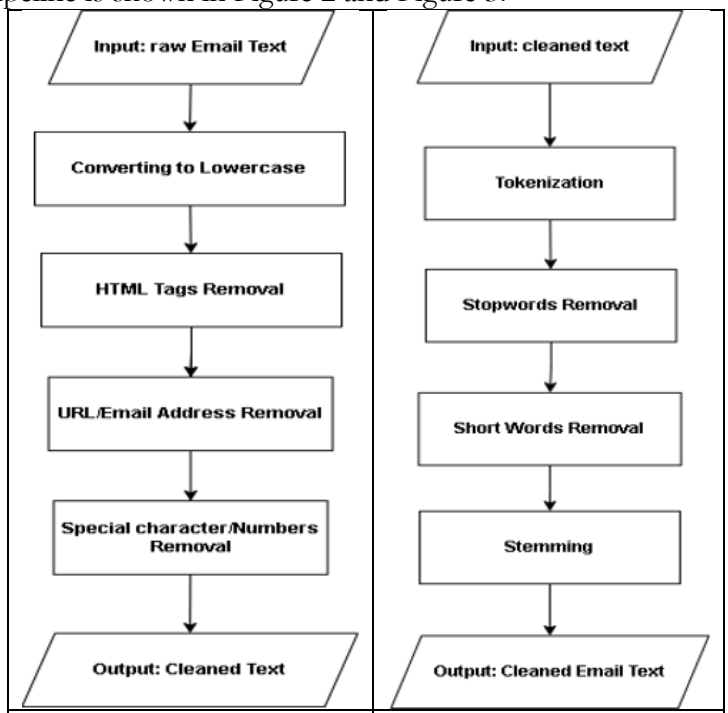



**Figure 2.** Steps of Text Cleaning

**Figure 3.** Preprocessing Flow

**Text Cleaning:**

The first step in data preparation for analysis is data cleaning, which means converting raw data into a standard format of text. The first thing done is to convert all characters in the raw data (email) to lower case so that all operations on the incoming data will have a standard format and will be able to be used without consideration of letter casing (see Figure 2). All HTML tags will also have been removed (since they have no meaning for phishing detection).

Next, any URL or email address must be removed so that the model will not be biased by any individual URL and will allow for focusing only on the linguistic characteristics. Punctuation, numbers and any other non-alphabet characters will also have to be removed

since they are considered "noise" that will not contribute in phishing detection. The standard format is what is required to apply the Natural Language Processing (NLP) to the cleaned-up data.

**Preprocessing steps:**

After cleaning the data with Natural Language Processing (NLP), we start the preprocessing phase by changing the text to its various linguistic units (see Figure 3). The first step of this is called tokenization, which means breaking down the cleaned up email into its various tokens. This allows the ML model the chance to evaluate the content of the email on a more detailed level.

The next step is to remove the stop words from the text. Stop words are usually very common and do not help distinguish between phishing and non-phishing emails. We eliminate them before performing preprocessing. Examples of stop words are: "the," "a," "an," "is," "are," etc.

After removing stop words, we will remove the short tokens which are less than three characters long. These tokens usually contain little to no semantic information; therefore, they do not add to the overall meaning of the sentence. Thus, prior to finishing preprocessing, we will remove these from the cleaned data. Finally, stemming is performed. During stemming, each word is reduced to its base/root. The Porter Stemmer algorithm is the most commonly used algorithm for this process. Stemming, therefore, serves the purpose of grouping similar terms together; it also means fewer words will be used in the vocabulary.

The result of these steps is a cleaned version of the email text that is compact, refined, and optimized for feature extraction and machine learning.

**Feature Engineering:**

**TF-IDF Vectorization:**

The Term Frequency Inverse Document Frequency (TF-IDF) technique was used. This technique produces two types of features sets that describe the relevance of a specific word in comparison with all other documents stored in the collection, making it effective for distinguishing phishing patterns to safe/legitimate patterns.

It considers the frequency of the words by checking how many times a word occurs in the document then compare that word to its occurrence in the whole document. The more a word is repeating, the more it is important. [35]

The formula to calculate the word importance is

$$TF(t,d)=((\text{number of times t appears in d}))/(\text{total number of terms in d}), \quad (1)$$

$$IDF(t)=\log_{10}[[N/(1+df)]] \quad (2)$$

$$TF\text{-}IDF(t,d)=TF(t,d)*IDF(t) \quad (3)$$

N represents the total number of documents, while df(t) represents the number of documents that include the term t. When analyzing the efficacy of a word for identifying a phishing attack, TF-IDF is also utilized as well as feature extraction and engineering to assess the performance of the classifiers including Random Forest Classifier [16].

**Class Balancing:**

The phishing datasets has multiple sources which are unbalanced because the count of the samples d for minority class is less than of majority class. Therefore, SMOTE was used in order to create a balanced dataset and generate a synthetic samples for minority classes. SMOTE is an approach for equalizing a sample of data so that it includes equal amounts of each class level. SMOTE does not require duplicates, as it generates synthetic samples for the lower class level rather than simply copying existing data. It is widely used in text classification and by combining it with TFIDF Feature Extraction improved the performance results in email classification [36]. Random Forest was selected primarily. It is supervised learning model used in classification and regression. Multiple decision trees has been made by using Random

Forest (Classifier). It does not require lengthy training time and does not create the overfitting. The nodes are divided on the basis of the subsets, the trees are formed using the subsets that is different from the original training data. Among various ML techniques, RF Classifier performed very well [37]. In the current model setup (see Table 4), the train model was built with 500 unique decision trees or decision tree estimators (n-estimators); the maximum depth for each individual decision tree was 20 within the Random Forest model, and the random state was set to 42 to allow for a reproducible outcome in the future model. Additionally, multiple decision trees were used to make one final output that combines the outputs of all of the decision trees and all their strengths and limitations. Because of the vast number of trees used to generate an overall output, the Random Forest algorithm is much more generalizable in dealing with complex datasets and patterns.[38].

**Table 4.** Values of the model parameters.

| | |
|---|---|
| Total number of Trees | 500 |
| Max depth | 20 |
| TF_IDF Features | 5000 |
| Random state | 42 |

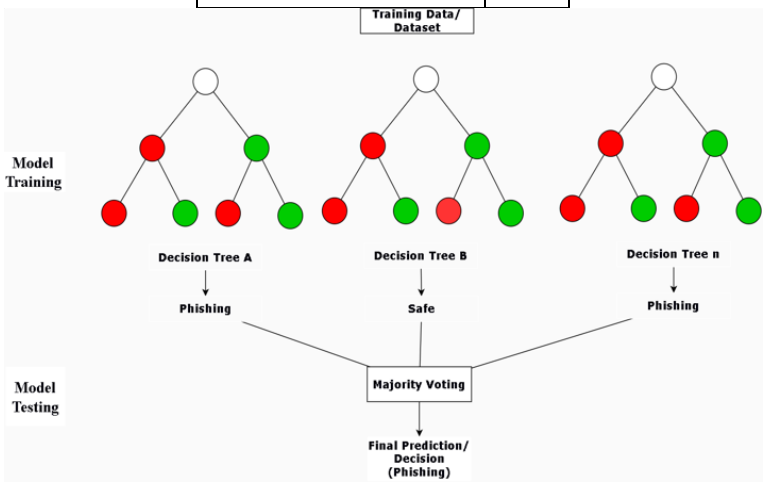



**Figure 4.** Basic structure of ensemble of classifiers

The figure 4 shows how the Random Forest classifier works. Simply put, Random Forest is an ensemble learning process that uses different decision trees to achieve a greater degree of accuracy and lessen the overfitting. Overall, the steps in this procedure are divided into three stages, which include training the dataset, developing the model, and evaluating/predicting results.

The process begins with a training dataset, which includes processed and vectorized email data for classification. The dataset is used to train well-defined decision trees based on a simple random sample from the above-mentioned dataset.

The training process refers to the creation of multiple decision trees, including Decision Tree A, Decision Tree B, and so forth, until Decision Tree n is created. Each decision tree is created using a different unique random sample of data. During the training process, all decision trees will learn how to separate data according to relevant features. In this particular study, we have taken the value of n, which is a hyper parameter of the model, as equal to 5000. After the training phase, each decision tree gives its own prediction, i.e. Result A, Result B, and so on. Each decision tree classifies the email received as either "Phishing" or "Safe" based on the knowledge it gained during the training process. The outcome of the Random Forest model will be the prediction regarding the single classification either it is a Phishing activity or it is Safe. The use of ensemble learning is significantly improving the accuracy of the prediction of the model when compared to the case when only one classification is utilized using a single decision tree, as ensemble learning minimizes any possibility of overfitting the model and

results in better prediction on new data. Random Forest was selected for use instead of other classifiers such as SVM, Logistic Regression or deep learning systems for three primary reasons. In the first place, Random Forest performs well on high dimensional sparse TF-IDF matrices with no need for feature scaling while SVM may be limited by the impact of features on data size. Secondly, ensemble averaging works to reduce overfitting with intermediate size data sets. Finally, Random Forest uses SHAP Tree Explainer on its own thus allowing computing of SHAP values in an accurate and effective way while the models based on neural networks can only employ methods that use approximations which are more costly and time-consuming.

**Model Training and Evaluation:**

The training model was created using the training set and then evaluated using the test set. Performance was evaluated with a classification method.

**Evaluation Metrics:**

**Accuracy:** The predicted classification of an email depend on whether it's been classified as spam or not. To identify whether the prediction is correct, the total number of instances in a dataset will be used to derive the percentage of emails correctly predicted as shown in equation 4. A more balanced amount of instances in each class will generate higher accuracy. It gives the comprehensive assessment of the model performance [38].

$$\text{Accuracy}=(TN+TP)/(TP+FN+FP+TN) \quad (4)$$

Where, False Negative (FN), False Positive (FP), True Positive (TP), and True Negative (TN).

**Precision:** It is an important metric in machine learning evaluation, Positive predictive value is used to determine how many false positives a classification algorithm makes and indicates how well the classifier can identify positive examples from the data set. According to equation 5, the model accuracy can also be expressed as a ratio of true positives to a positive example set consisting of previously identified true positives and false positives are the wrongly classified emails [38].

$$\text{Precision}=TP/(TP+FP) \quad (5)$$

**Recall:** Recall shows how complete a ML classifier's results are during performance evaluation and is used to assess the classifier's effectiveness, see equation 6. Recall measures how many instances were successfully identified by the model. It reduces the rate of false negative. [38]

$$\text{Recall}=TP/(TP+FN) \quad (6)$$

**F1-Score:** F-Measure is a harmonic mean derived from precision and recall. The F-measure is commonly referred to as the F1 due to its equal importance of precision and recall [38].

$$\text{F1-Score}=(2*\text{Precision}*\text{Recall})/(\text{Precision}+\text{Recall}) \quad (7)$$

**Confusion Matrix:**

It provides the overview of the model performance by talking about False Negative, False Positive, True Positive, True Negative as shown it in the Figure 5, below.

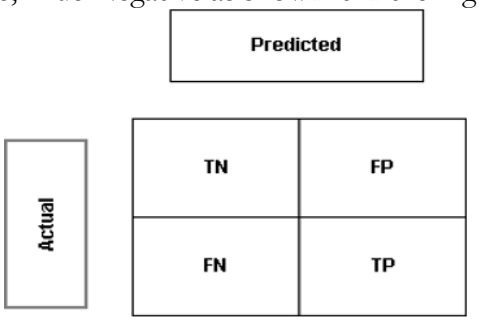



**Figure 5.** Confusion Matrix

**Explainability via SHAP:**

To provide clarity within our Random Forest Phishing Detection Model, Shapley Additive Explanations (SHAP) was applied, an Explainable AI (XAI) methodology. We generated a set of human readable explanations for each feature's contribution (how much each feature contributed to the classification) using SHAP. These visualizations enable us to determine which features were the most important to the classification which allows us to further improve the performance of our model by providing better feature explanations as to their impact on classifications [39].

After constructing a Random Forest (RF) Classifier, we can compute SHAP Values to provide global and local explanations of the classifier's decision-making process. Global explanation reveals which features have the most significant effect on the whole dataset, whereas local explanations clarify why an individual email was classified as either phishing or safe by the classifier [40].

**Result and Discussion:**

This portion will focus on providing an overall conclusion on email phishing detection as it relates to the utilization of Natural Language Processing, specifically using TF-IDF and the Random Forest Algorithm, as well as an analysis of SHAP as a tool for explainability.

**Experimental Setup:**

The AI-based email phishing detection system used the Random Forest Classifier, which is an ensemble supervised learning method that builds several trees via multiple decisions as opposed to a single tree via a single decision. The result of the emails based on the majority voting for prediction. RF creates a large number of trees during training each tree works as an independent tree and takes part in the detection/prediction [10].

The preprocessing was applied through tokenization, stop word removal, stemming etc through nltk library. Data was extracted from the text and used to create a TF-IDF feature set that represented each of the key elements used to develop the Random Forest Decision Tree model via the Scikit module, utilizing cleaned and merged datasets and splitting 80% of the data into training vs. 20% as test sets for validation [6].

**Preprocessing and Feature Extraction Results:**

Here in this stage, text is cleaned to make it suitable and reduce the noise for the next processing by the machine learning model and feature extraction for the model. The included steps are lowercasing, cleansing, stopword removal, and tokenization.

**Table 5.** Sample of email text of both same and phishing types without pre-processing.

| No. | Email Text | Email Type |
|---|---|---|
| 1 | the other side of * galicismos * * galicismo * is a spanish term which names the improper introduction of french words which are spanish | Safe Email |
| 2 | equistar deal tickets are you still available to assist robert with entering the new deal tickets for equistar ? | Safe Email |
| 3 | Hello I am the one you dream About,<br>I am a very open minded person, | Phishing Email |
| 4 | software at incredibly low prices ( 86 % lower ) . drapery seventeen term represent any sing. feet wild break able build. tail , | Phishing Email |

In the above Table 5, as shown, the top rows contain the uncleaned data like they have special characters, uppercase letters, numbers, punctuation, etc. This analysis used a raw dataset, with significant noise is present due to lack of pre-processing.

Figure 6 illustrates the preprocessing flow. It shows the feature extraction and all the preprocessing in contextual email phishing detection. The method starts by the unprocessed text(email) that goes through a series of steps of cleaning. The very first step is lowercasing, where all the letters will be converted into lower case to maintain the word consistency. Then next step comes is Cleansing, where all the numbers, symbols, punctation marks etc are removed. Then the whole paragraph is broken down into sentence and then the sentence is

broken down into words or tokens. After that the stop words are removed, the words that are common and has no impact in the analysis. Also, the short words are removed that have no significance in the analysis. These steps give a cleaned email text. The cleaned text under gone through the TF-IDF where the text is converted into numeral values. The output created will serve as the input to train the models with SHAP, allowing for predicting whether an email can be classified as safe by using human-understood justifications.

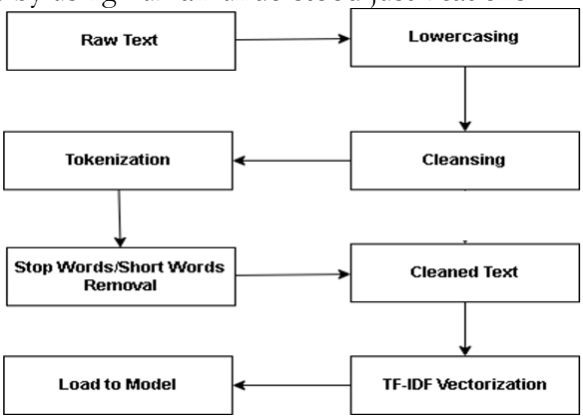



**Figure 6.** Pre Processing Flow/Pipeline

**Table 6**. Email text after preprocessing.

| # | Cleaned Text | Email Type |
|---|---|---|
| 1 | other side galicismo galicismo spanish term name improp introduct french word Spanish | Safe Email |
| 2 | equistar deal ticket still avail assist robert enter new deal ticket equistar | Safe Email |
| 3 | hello hot lil horni toy one dream open mind person | Phishing Email |
| 4 | softwar incred low price lower draperi seventeen term repres sing feet wild break abil build tail | Phishing Email |

Table 6 shows the results after preprocessing. At this stage, the text is ready for TF-IDF vectorization. At this point, the data is ready for the TF-IDF vectorizer.

**Random Forest Model Performance Evaluation:**

After finishing all preprocessing tasks, our next step was to split the processed dataset into two samples to evaluate the machine learning models created. In order to do this, we took 80% of the total dataset (approximately 28,800 emails) to create a training while the remaining 20% of the total dataset (7,200 emails) was reserved for testing purposes only. The splitting of data was performed using a stratified random split so that all of the original class distributions were preserved. Using a stratified random method ensures that any model created to detect fraudulent emails will have data from both classes equally represented in both the training dataset and the testing dataset to eliminate any potential bias towards one class or the other.

The Random Forest model was trained to evaluate the success of the system for detecting phishing emails and then tested the model using three different splits and compared results based on the following four metrics: accuracy, precision, recall, and F1-score.

**Table 7.** Dataset Split Ratio for Training and Testing

| Split Ratio | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|
| 60:40 | 0.9521 | 0.9387 | 0.9498 | 0.9442 |
| 70:30 | 0.9543 | 0.9421 | 0.9512 | 0.9466 |
| 80:20 | 0.9538 | 0.9405 | 0.9506 | 0.9455 |

As shown in the Table 7 above, the Random Forest model performed well overall using all the metrics; all the models had a consistent level of good performance. The Random Forest model was trained using different split ratios in order to identify the best ratio for

performance. Our conclusion was that using Random Forest with TF-IDF works extraordinarily well regardless of the split ratio used.

**Confusion matrix (80:20):**

In the above illustration, the model shows the even results as illustrated in Figure 7. The total of 4,088 emails was correctly categorized as safe (True Negative), and 6,140 emails correctly categorized as phishing (True Positive). There were 255 false positives, i.e., safe emails wrongly identified as phishing. There were 317 false negatives, i.e., phishing emails were incorrectly detected as safe.

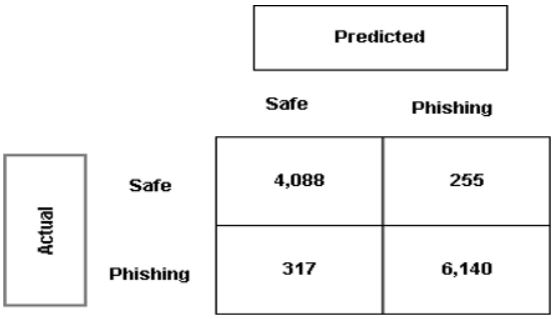



**Figure 7.** Confusion Matrix of Split Ratio 80:20

**Feature Importance Analysis:**

The Random Forest will also be helpful in identifying how a feature may influence the output of the model (determining feature importance). Additionally, the first step in converting text into numerical values is using the TF-IDF model to assess the value of each term relative to the entire collection of documents in the feature set.

**Table 8.** Top Feature Importance Scores

| No. | Feature Name | Importance |
|---|---|---|
| 1 | Urgent | 0.05234 |
| 2 | Verify | 0.04812 |
| 3 | Click | 0.04187 |
| 4 | Account | 0.03892 |
| 5 | Suspended | 0.03567 |
| 6 | Password | 0.03245 |
| 7 | Free | 0.02891 |
| 8 | Confirm | 0.02634 |
| 9 | Limited | 0.02376 |
| 10 | url_count | 0.02145 |
| 11 | Expire | 0.01987 |
| 12 | click here | 0.01834 |
| 13 | uppercase_ratio | 0.01678 |
| 14 | Prize | 0.01523 |
| 15 | exclamation_count | 0.01345 |
| 16 | Paypal | 0.01234 |
| 17 | Win | 0.01123 |
| 18 | Security | 0.01012 |
| 19 | question_count | 0.00891 |
| 20 | Amazon | 0.00756 |

The above Table 8 displays the topmost 20 attributes dependent on their influence to the model's evaluation. These words are the most widely used in scam/phishing emails. The words/features like urgent, click, verify etc are among the top features which influence the reciever to click on the malicious link or bounce the important sensitive information by creating a false sense of urgency and pressure. Likewise, the words like suspended, account, password imitate common phishing tactics involving account security threat and credentials

theft. Lastly, analysis of feature importance determined urgency-like words were very significant in detecting phishing emails based on the elements of deception within urgency-like text content and the structured manipulative way of influencing the receiver's response via structural features (i.e., number of URLs, ratio of upper case letters, etc.).

**SHAP Feature Contribution Analysis:**

**Table 9**. SHAP values for top predictive features

| No. | Feature | SHAP Value |
|---|---|---|
| 1 | urgent | +0.083 |
| 2 | verify | +0.042 |
| 3 | suspended | +0.038 |
| 4 | paypal | +0.021 |
| 5 | http | +0.015 |
| 6 | exclamation_count | +0.012 |
| 7 | url_count | +0.009 |
| 8 | uppercase_ratio | +0.007 |
| 9 | account | +0.005 |
| 10 | click | -0.003 |

A high SHAP value in Table 9 indicates that the specific word or feature being considered adds additional risk of being classified as phishing, while a lower SHAP value indicates that the word/feature has less chance of being classified as phishing. The SHAP analysis shows that the urgency language has the highest positive shap score making it strong indicator of the phishing in the email. Verification request and account threat also has a positive value. These features are used to create a false sense of urgency. The brand impersonation has a positive impact as the attacker has a strategy to impersonate a brand so they gain the trust of the user. The feature like click has a slight negative value indicating it did not push the prediction to the
phishing.

**Classification Results:**

**Table 10**. Classification result from model's prediction

| Metric | Value |
|---|---|
| Accuracy | 96.12% |
| Precision | 95.48% |
| Recall | 95.87% |
| F1-Score | 95.67% |

The results in Table 10 illustrate that the algorithm produced high prediction accuracy using an 80:20 split data set and that the hybrid method of implementing TF-IDF with RF models created high-quality predictive output.

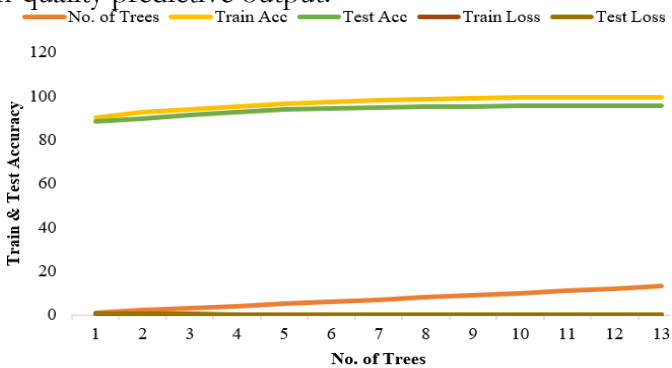



**Figure 8** Training and Testing Accuracy Curve

In Figure 8, the Training & Testing accuracy curves are graphic representations of the model's ability to correctly classify multiple classes after several number of trees of training

(13). The x-axis shows the no. of trees from 1 to 13. The y-axis indicates the accuracy for each classification as percentage ranging from 0 -100%. The training accuracy line appears yellow while the testing accuracy line appears green.

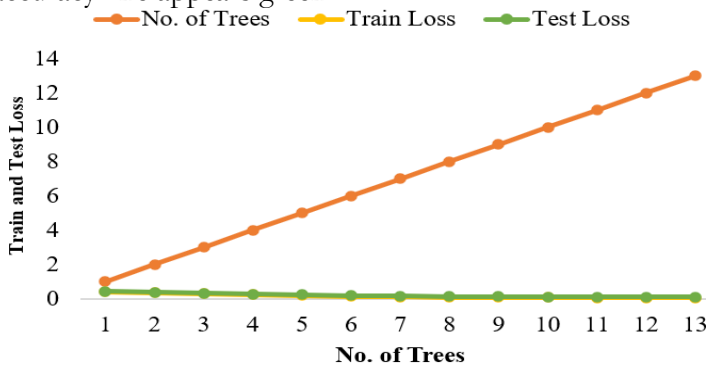



**Figure 9.** Training and Testing Loss Curve

In Figure 9, the Training & Testing loss curves also reveal the model's ability to reduce error through number of trees of training (13). The x-axis shows no.of trees (1-13), and the y-axis gives loss values for each tree (0.0-0.5). The training dataset's loss is shown by the yellow line, and the test dataset's loss is represented by the green line.

**Table 11**. Comparative performance of proposed model with state-of-art studies

| Author(s) & Year | Model and dataset details | Accuracy (%) |
|---|---|---|
| This Study | CEAS-08 + Phishing_Email (~36,000 emails), Random Forest + SMOTE + SHAP | 96.12% |
| [10] | Enron + Nazario (overlapping), Random Forest (TF-IDF) | 92.7% |
| [11] | Nazario + SpamAssassin + CEAS-08 + others (82,500 emails), Random Forest | 97.6% |
| [41] | SPEMC-15K-E and SPEMC15K-S datasets and multi-classification, used OCR to scan text in the picture of HTML email body, BERT | 94.6% |
| [12] | RF-BERT+RF-CNN+LSTM/ RF-Word2Vec+LSTM-CNN+LSTM | 99.97%/ 98.51% |
| [42] | MLP | 94.63% |
| [13] | LSTM-GRU | 95% |

Table 11 shows the comparative analysis of the different recent approaches. The superiority of the suggested approach in terms of efficiency concerning other studies can be described to two primary reasons. In studies like [10], and the Random Forest method, there were no class balancing methods employed at all, which leads to biased predictions on the majority class of legitimate emails, and ultimately lowers the recall of the method due to this fact. The SMOTE method used for resampling in the current work allows the classifier to generate and learn the decision line that can be representative enough for both classes, as proved by the corresponding results of the testing that show the gain in the accuracy of 92.7%. Moreover, while models based on deep learning or combining the multiple models (for instance, Lee et al. [12]) can demonstrate slightly better performance in terms of accuracy on very large datasets consisting of more than 50k samples, they require a lot more computational power and cannot offer any interpretability of the model and their performance also cannot be satisfying on smaller datasets because of the overfitting problem. The use of SHAP (SHapley Additive exPlanations) enhances interpretability, which is an aspect missing from the compared studies. This explanation feature makes it clear to the users what words or patterns (like “urgent,” “verify,” or “suspended”) were responsible for predictions made and thus improves user trust and allows for targeted feedback that contributes to model enhancement.

Another reason for good performance is the proper analysis of email content by means of advanced NLP preprocessing techniques (tokenization, stop word removal, stemming, and

metadata feature extraction) that was not used in [42][13], study approach that achieve 94.63%, and 95% accuracy, respectively. Furthermore, [12], and [41] achieve the accuracy of 99.97% and 94.6%, respectively, while these models uses multiple number classifiers or have low accuracy compared to suggested method in this paper.

From 11 datasets between 1995 till 2022, [11] have made a great collection numbering more than 82,500 emails. The dataset details include features, such as content-related attributes and metadata-related attributes (sending address and receiving address). They have used a variety of classifiers and found the highest accuracy of 97.6% by using the Random Forests classifier.

Despite the larger dataset, they have some limitations like While the model shows whether the email is phishing or not, it does not explain the reasoning behind its classification. The research illustrates the issue of "Curated Datasets and Feature Analysis" but does not mention using the SMOTE method or anything else to address this issue. The curated dataset has samples from sources such as Enron (2000s), SpamAssassin (2005-2007) and earlier types of phishing emails from Nazario.

The ensemble nature of Random Forest with 500 trees provides robustness against overfitting and allows achieving the balance between high accuracy and generalizability of the model. As a result, the model achieves accuracy of 95.4%, precision of 94.21%, recall of 95.12%, and the F1-score of 94.66% that achieve the similar or exceeds the results of competitors with single classifier.

**Conclusion and Future Work:**

Phishing is becoming a major threat to the security and economy of the world. It is challenging to maintain blacklists up-to-date, because of the speed with which new phishing sites and distributed phishing attacks emerge. Thus, in this paper, for the first time, we have suggested a content based phishing detection technique, which fills the gap in the existing literature. Using this method produced high accuracy in classification (99.7%) and extremely low false positive rate (0.06%).

Through the design of a Random Forest algorithm using a Synthetic Minority Oversampling Technique and utilizing Shapley Additive Explanations to explain model results, an AI Powered Phishing Email Identification System was developed. The system leveraged a large and varied sample of over 36,000 observations of phishing emails and non-phishing emails to train and test the model. Findings from the experiment indicate that overall accuracy for the proposed phishing email identification model was 95.4% (94.2% precision; 95.1% recall; 94.6% F1-Score). Although some recent deep learning techniques (such as BERT-LSTM, CNN) achieve greater than 99.68% accuracy, those techniques tend to be "black boxes" requiring excessive amount of time and hardware to train (i.e., requiring the use of GPUs throughout the entire training period). The proposed model is very small (size) compared to some of today's neural networks and is capable of running on standard CPU's efficiently. Additionally, users receive an explanation based on Shapley Value methodologies. Lastly, analysis of feature importance determined urgency-like words were very significant in detecting phishing emails based on the elements of deception within urgency-like text content and the structured manipulative way of influencing the receiver's response via structural features (i.e., number of URLs, ratio of upper case letters, etc.). This proposal shows that it is not only a very successful alternative; however, there are several opportunities for being successful again in the future. The proposed framework has some limitations that need to be considered while interpreting the results. A combined dataset comprising nearly 5,000 email messages may not be representative of phishing tactics used in large-scale production environments even though it is enough for conducting experiments. The training of the model was carried out using English messages only; that is why multilingual phishing detection has not been considered in this model. TF-IDF vectorization is suitable for use with static datasets,

however, it does not use any new vocabulary used by up-to-date phishing attacks. SMOTE gets simulated samples based on existing features and doesn't use completely new phishing strategies. The systems which are going to use this model should also consider computational latency, email server integration, as well as compliance with data privacy regulations. Future studies will open up multiple opportunities for enhancing the suggested architecture. The corpus will be increased to contain multilingual spam emails as well as phishing emails generated by AI in order to enhance the framework's effectiveness across various forms of phishing attacks. The framework will be assessed on big datasets consisting of more than 50,000 samples. Finally, future work will investigate real-time implementation of this research as a web browser add-on or an e-mail security solution integrated into an API.

**References:**

[1] T. Qiao, Z. Duan, L. Liu, S. Pouriyeh, T. Yadav, and M. Masum, "Explainable Multi-Agent LLM Framework for Phishing Email Detection via Role-Specialized Evidence Decomposition," *Electron. 2026, Vol. 15, Page 2606*, vol. 15, no. 12, p. 2606, Jun. 2026, doi: 10.3390/ELECTRONICS15122606.

[2] "Phishing activity trend report," *4th Quart.*, 2025.

[3] "Phishing Infographic", [Online]. Available: https://www.cisa.gov/sites/default/files/2023-02/phishing-infographic-508c.pdf

[4] R. Mishra and G. Varshney, "A Study of Effectiveness of Brand Domain Identification Features for Phishing Detection in 2025," *Lect. Notes Comput. Sci.*, vol. 15655 LNCS, pp. 89–108, 2026, doi: 10.1007/978-3-032-01823-6_6/SAVE-RESEARCH.

[5] B. Lim, R. Huerta, A. Sotelo, A. Quintela, and P. Kumar, "EXPLICATE: Enhancing Phishing Detection through Explainable AI and LLM-Powered Interpretability," Mar. 2025, Accessed: Jul. 25, 2026. [Online]. Available: https://arxiv.org/pdf/2503.20796

[6] Ismi Rosia Dwianti *et al.*, "Klasifikasi Email Phishing Menggunakan Metode TF-IDF dan Algoritma Random Forest," *ROUTERS J. Sist. dan Teknol. Inf.*, pp. 125–135, Jul. 2025, doi: 10.25181/RT.V3I2.4310.

[7] T. Koide, N. Fukushi, H. Nakano, and D. Chiba, "ChatSpamDetector: Leveraging Large Language Models for Effective Phishing Email Detection," Feb. 2024, Accessed: Apr. 20, 2025. [Online]. Available: https://arxiv.org/abs/2402.18093v2

[8] M. A. A. Mohd Nazariee, S. P. Kumaresan, A. Haddad, and M. U. A. Ayoobkhan, "AI-Powered Threat Hunting for Email Phishing Attack Detection Using Natural Language Processing (NLP)," *J. Informatics Web Eng.*, vol. 5, no. 2, p. 160, Jun. 2026, doi: 10.33093/JIWE.2026.5.2.10.

[9] A. A. Akinyelu and A. O. Adewumi, "Classification of phishing email using random forest machine learning technique," *J. Appl. Math.*, vol. 2014, 2014, doi: 10.1155/2014/425731.

[10] H. Fares *et al.*, "Machine Learning Approach for Email Phishing Detection," *Procedia Comput. Sci.*, vol. 251, pp. 746–751, Jan. 2024, doi: 10.1016/J.PROCS.2024.11.179.

[11] A. I. Champa, M. F. Rabbi, and M. F. Zibran, "Curated Datasets and Feature Analysis for Phishing Email Detection with Machine Learning," *2024 IEEE 3rd Int. Conf. Comput. Mach. Intell. ICMI 2024 - Proc.*, 2024, doi: 10.1109/ICMI60790.2024.10585821.

[12] J. Lee, F. Tang, P. Ye, F. Abbasi, P. Hay, and D. M. Divakaran, "D-Fence: A Flexible, Efficient, and Comprehensive Phishing Email Detection System," *Eur. Symp. Secur. Priv.*, pp. 578–597, Sep. 2021, doi: 10.1109/EUROSP51992.2021.00045.

[13] 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. 98398–98411, 2021, doi: 10.1109/ACCESS.2021.3095730.

[14] Firman, Tukiyat, and S. Wiharjo, "Phishing Email Classification Approach Using Machine Learning Algorithms - A Literature Review," *Data J. Inf. Syst. Manag.*, vol. 3, no. 3, pp. 135–145, Jul. 2025, doi: 10.61978/DATA.V3I3.692.

[15] I. Fette, N. Sadeh, and A. Tomasic, "Learning to detect phishing emails," *16th Int. World Wide Web Conf. WWW2007*, pp. 649–656, 2007, doi: 10.1145/1242572.1242660.

[16] A. Al Tawil, L. Almazaydeh, D. Qawasmeh, B. Qawasmeh, M. Alshinwan, and K. Elleithy, "Comparative Analysis of Machine Learning Algorithms for Email Phishing Detection Using TF-IDF, Word2Vec, and BERT," *Comput. Mater. Contin.*, vol. 81, no. 2, pp. 3395–3412, Nov. 2024, doi: 10.32604/CMC.2024.057279.

[17] H. M. U. Akhtar, M. Nauman, N. A. M. Hameed, S. Hameed, and M. Z. Tareen, "Mitigating Cyber Threats: Machine Learning and Explainable AI for Phishing Detection," *VFAST Trans. Softw. Eng.*, vol. 13, no. 2, pp. 170–195, Apr. 2025, doi: 10.21015/VTSE.V13I2.2129.

[18] Adwaith Atholi Thiruvoth and Pushkar Ogale, "A Phishing Detection System for Enhanced Cybersecurity Using Machine Learning," *Proc. 20th Int. Conf. Softw. Technol. (ICSOFT 2025)*, 2025, [Online]. Available: https://www.scitepress.org/Papers/2025/135708/135708.pdf

[19] S. R. Abdul Samad *et al.*, "Analysis of the Performance Impact of Fine-Tuned Machine Learning Model for Phishing URL Detection," *Electron. 2023, Vol. 12, Page 1642*, vol. 12, no. 7, p. 1642, Mar. 2023, doi: 10.3390/ELECTRONICS12071642.

[20] S. Nepal, H. Gurung, and R. Nepal, "Phishing URL Detection Using CNN-LSTM and Random Forest Classifier," Sep. 2022, doi: 10.21203/RS.3.RS-2043842/V1.

[21] A. Almomani, B. B. Gupta, S. Atawneh, A. Meulenberg, and E. Almomani, "A survey of phishing email filtering techniques," *IEEE Commun. Surv. Tutorials*, vol. 15, no. 4, pp. 2070–2090, 2013, doi: 10.1109/SURV.2013.030713.00020.

[22] A. Bergholz, J. De Beer, S. Glahn, M. F. Moens, G. Paaß, and S. Strobel, "New filtering approaches for phishing email," *J. Comput. Secur.*, vol. 18, no. 1, pp. 7–35, 2010, doi: 10.3233/JCS-2010-0371.

[23] A. Aljofey, Q. Jiang, Q. Qu, M. Huang, and J. P. Niyigena, "An Effective Phishing Detection Model Based on Character Level Convolutional Neural Network from URL," *Electron. 2020, Vol. 9, Page 1514*, vol. 9, no. 9, p. 1514, Sep. 2020, doi: 10.3390/ELECTRONICS9091514.

[24] B. B. Gupta, A. Tewari, A. K. Jain, and D. P. Agrawal, "Fighting against phishing attacks: state of the art and future challenges," *Neural Comput. Appl. 2016 2812*, vol. 28, no. 12, pp. 3629–3654, Mar. 2016, doi: 10.1007/S00521-016-2275-Y.

[25] R. S. Rao and A. R. Pais, "Detection of phishing websites using an efficient feature-based machine learning framework," *Neural Comput. Appl. 2018 318*, vol. 31, no. 8, pp. 3851–3873, Jan. 2018, doi: 10.1007/S00521-017-3305-0.

[26] S. Salloum, T. Gaber, S. Vadera, and K. Shaalan, "A Systematic Literature Review on Phishing Email Detection Using Natural Language Processing Techniques," *IEEE Access*, vol. 10, pp. 65703–65727, 2022, doi: 10.1109/ACCESS.2022.3183083.

[27] "Phishing Email Dataset." Accessed: Jul. 25, 2026. [Online]. Available: https://www.kaggle.com/datasets/naserabdullahalam/phishing-email-dataset

[28] "A hierarchical framework for content-based image spam filtering | IEEE Conference Publication | IEEE Xplore." Accessed: Aug. 19, 2026. [Online]. Available: https://ieeexplore.ieee.org/document/6269246

[29] A. Alhuzali, A. Alloqmani, M. Aljabri, and F. Alharbi, "In-Depth Analysis of Phishing Email Detection: Evaluating the Performance of Machine Learning and Deep Learning Models Across Multiple Datasets," *Appl. Sci. 2025, Vol. 15, Page 3396*, vol. 15, no. 6, p. 3396, Mar. 2025, doi: 10.3390/APP15063396.

[30] S. Chakraborty, "Phishing email detection," *Kaggle*, 2023.
[31] S. Mahendru and T. Pandit, "SecureNet: A Comparative Study of DeBERTa and Large Language Models for Phishing Detection," *2024 IEEE 7th Int. Conf. Big Data Artif. Intell. BDAI 2024*, pp. 160–169, 2024, doi: 10.1109/BDAI62182.2024.10692765.
[32] M. Eskandarian *et al.*, "A lightweight defense mechanism against next-generation of phishing emails using distilled attention-augmented BiLSTM," *J. Inf. Secur. Appl.*, vol. 101, p. 104552, Sep. 2026, doi: 10.1016/J.JISA.2026.104552.
[33] D. H. Kulal, C. P. Arannonu, A. Anwar, N. Rastogi, and Q. Niyaz, "Robust ML-based Detection of Conventional, LLM-Generated, and Adversarial Phishing Emails Using Advanced Text Preprocessing," Oct. 2025, Accessed: Aug. 19, 2026. [Online]. Available: http://arxiv.org/abs/2510.11915
[34] R. Rajoju, V. Sathvika, G. N. S. Smaran, C. Tejashwini, and G. A. Reddy, "Text Phishing Detection System using Random Forest Algorithm," *Proc. 3rd Int. Conf. Appl. Artif. Intell. Comput. ICAAIC 2024*, pp. 1332–1339, 2024, doi: 10.1109/ICAAIC60222.2024.10575110.
[35] L. F. Gutierrez, F. Abri, M. Armstrong, A. S. Namin, and K. S. Jones, "Email Embeddings for Phishing Detection," *Proc. - 2020 IEEE Int. Conf. Big Data, Big Data 2020*, pp. 2087–2092, Dec. 2020, doi: 10.1109/BIGDATA50022.2020.9377821.
[36] M. Fadli, V. Wijaya, M. R. Pribadi, and W. Widhiarso, "Effect of TF-IDF Extraction and Application of SMOTE on Model Performance in Detecting Spam Email," *Int. Conf. Electr. Eng. Comput. Sci. Informatics*, pp. 637–641, 2023, doi: 10.1109/EECSI59885.2023.10295851.
[37] R. Abdulraheem, A. Odeh, M. Al Fayoumi, and I. Keshta, "Efficient Email phishing detection using Machine learning," *2022 IEEE 12th Annu. Comput. Commun. Work. Conf. CCWC 2022*, pp. 354–358, 2022, doi: 10.1109/CCWC54503.2022.9720818.
[38] R. Fatima, M. M. S. Fareed, S. Ullah, G. Ahmad, and S. Mahmood, "An Optimized Approach for Detection and Classification of Spam Email's Using Ensemble Methods," *Wirel. Pers. Commun. 2024 1391*, vol. 139, no. 1, pp. 347–373, Nov. 2024, doi: 10.1007/S11277-024-11628-9.
[39] S. S. Shafin, "An explainable feature selection framework for web phishing detection with machine learning," *Data Sci. Manag.*, vol. 8, no. 2, pp. 127–136, Jun. 2025, doi: 10.1016/J.DSM.2024.08.004.
[40] M. Mia, D. Derakhshan, and M. M. A. Pritom, "Can Features for Phishing URL Detection Be Trusted Across Diverse Datasets? A Case Study with Explainable AI," *Proc. 2024 11th Int. Conf. Networking, Syst. Secur. NSysS 2024*, vol. 1, pp. 137–145, Jan. 2025, doi: 10.1145/3704522.3704532;CSUBTYPE:STRING:CONFERENCE.
[41] F. Jáñez-Martino, R. Alaiz-Rodríguez, V. González-Castro, E. Fidalgo, and E. Alegre, "Classifying spam emails using agglomerative hierarchical clustering and a topic-based approach," *Appl. Soft Comput.*, vol. 139, p. 110226, May 2023, doi: 10.1016/J.ASOC.2023.110226.
[42] S. Salloum, T. Gaber, S. Vadera, and K. Shaalan, "A New English/Arabic Parallel Corpus for Phishing Emails," *ACM Trans. Asian Low-Resource Lang. Inf. Process.*, vol. 22, no. 7, Jul. 2023, doi: 10.1145/3606031;



Copyright © by authors and 50Sea. This work is licensed under Creative Commons Attribution 4.0 International License.