RESEARCH ARTICLE

INTERNATIONAL RESEARCH JOURNAL OF MULTIDISCIPLINARY TECHNOVATION

Asian Research Association

DOI: 10.54392/irjmt2658

EPAD: Email Phishing Attack Detection Powered by Machine Learning and Natural Language Processing

Mansi Harihar [a], Shilpa Deshpande [a,*], Mahendra Deore [a]

[a] Department of Computer Engineering, Cummins College of Engineering for Women, Pune, Maharashtra, India.

* Corresponding Author Email: shilpa.deshpande@cumminscollege.in

DOI: https://doi.org/10.54392/irjmt2658

Received: 13-01-2026; Revised: 10-08-2026; Accepted: 07-09-2026; Published: 15-09-2026

Abstract: Phishing is a form of social engineering attack which takes advantage of human vulnerability to exploit users. Email phishing is an emerging and prevalent form of cyber attack that impersonates a trusted entity. In today's world, email has become a de facto mechanism of communication for individuals and businesses. Phishing attacks on emails are growing constantly. Hence, the need for email phishing attack detection is increasingly becoming crucial. This paper presents an email phishing attack detection (EPAD) system using Machine Learning and Natural Language Processing (NLP) techniques to detect phishing emails. The proposed work makes use of algorithms which include Support Vector Classifier, Naive Bayes, Logistic Regression, Random Forest and eXtreme Gradient Boosting (XGBoost) for the prediction of benign or phishing emails. The EPAD system focuses on content based and textual features in detecting phishing emails. Readability scores, sentiments and attachments related information are also considered for the effective phishing email detection. The EPAD system uses NLP techniques which include Term Frequency-Inverse Document Frequency (TF-IDF) and Word2Vec for capturing the semantics of the email messages. Explainable AI is employed to provide interpretability of results of email phishing detection. Consequently, the EPAD system offers the reasoning about the prediction of phished or benign email and thereby enhances user trust. Experimental outcomes demonstrate that XGBoost with Word2Vec performs optimally in comparison with other algorithms in respect of accuracy, F1-score, Area under Receiver Operating Characteristic Curve (ROC-AUC) and Matthews Correlation Coefficient (MCC) for the detection of phishing emails. A thorough evaluation carried out in terms of ablation study, statistical performance analysis and sensitivity analysis validates the effectiveness of the proposed system.

Keywords: Cyber-Attacks, Email Phishing, Explainable AI, Machine Learning, Natural Language Processing.

1. Introduction

With the digital transformation, it brings several advantages to businesses on one side, whereas on the other side it is expanding to new cyber risks [1] like data breaches, fraud and privacy breaches. Cyber criminals perform attacks [2] like malware attacks, social engineering, and ransomware to steal data, money, and identity and disrupt computer systems. Among all cyber-attacks, phishing indicates one of the most prevalent and harmful attacks. Phishing represents a kind of social engineering attack in which attackers manipulate people into divulging sensitive information [3, 4].

Phishing can take place in various ways like using email, SMS or voice calls. It intends to deceit users into revealing sensitive personal information [5] or downloading malicious software by mimicking trusted and reliable entities. Email phishing is the more common form of phishing where fraudulent email messages look real. In today's world, email serves as a versatile form of communication in various domains such as finance, healthcare, e-commerce and government services. Its significance has been intensified during COVID-19 outbreak by the shift to remote work, enabling cyber criminals to use this situation for sending phishing emails [6]. Since then, attackers have been leveraging increase in remote work as an opportunity to target users with phishing emails, spam and malicious URLs. These emails typically lure users through fake links, urgent and panicking threats or malicious attachments [7]. The fallout often includes stolen credentials, malware software infection, unauthorized access and monetary losses [8]. As per APWG [9], phishing incidents including phishing email attacks have been observed to be continuously evolving.

Hackers exhibit phishing exploiting domain-specific trust and workflow, indicating urgency. Email phishing attacks have been seen taking place increasingly in various real world enterprise applications [10]. In banking, attackers may impersonate trusted

banking authority by sending mails like account suspension, KYC update requests and fake transaction messages. They ask for OTPs, PINs and credentials to visit spoofed banking websites causing financial damage. The educational systems also may face phishing attacks where attackers target students and faculty through emails. These emails sound official and use academic language requesting to visit portals. This leads to compromise student and staff credentials and access to institutional systems. The healthcare domain has patient data called protected health information (PHI). If an attacker impersonates a hospital administrator or insurance provider by sending mails to get data, leakage of this data leads to identity theft which is a risk faced by patients. Consequently, the healthcare provider faces reputation damage and incurs huge fines. The above elaborated email phishing scenarios imply that from the perspective of end users, it is absolutely essential to detect the email phishing attacks for enabling trustworthy email communication.

The necessity of developing a system to detect phishing emails is evident. Researchers have been working on the problem of email phishing detection. Different solutions have been offered. However, in spite of ongoing work, there exist gaps in the research in the area of email phishing, which are elaborated below. Conventional methods rely on blacklist or whitelist methods [11] and rule based approaches [12, 13]. Blacklist is not effective for zero-day attacks where the static list is not updated. Whitelist approaches may flag previously unseen sender and domain. The rule based approaches can result in false positives where benign or legitimate emails are misclassified as phishing attempts. Therefore, these techniques are ineffective against evolving phishing tactics.

Since email has multiple parts, attackers may exploit one or more components of it. Some of the previous approaches [9] either include text or URL part of email only to detect phishing instead of email as whole. Machine Learning (ML) based techniques have also been explored in literature for identifying the phishing emails. Before Natural Language Processing (NLP) gained popularity, content based features have been used by the ML based approaches in [14, 15] which analyze content like sender address, URL and email message. But these approaches do not cover the semantics of the email messages. While NLP methods capture semantic information of email text, these approaches miss other spoofed components like URLs and attachment related information [16]. Researchers use complex algorithms which are black-box in nature to detect phishing emails. These algorithms lead to lack of transparency and interpretability of the results [17].

Given the evolving phishing tactics, detection approaches need to be flexible to handle linguistic patterns such as structure and semantics of contents, obfuscated text, URLs, and attachments found in the phishing email. Thus, these aspects from multiple parts of email need to be considered comprehensively by the approaches for the effective detection of email phishing attacks. Moreover, these approaches also need to provide interpretation of phishing email detection results so as to gain user trust.

To address the above mentioned issues, this paper presents a system called EPAD to detect phishing emails. EPAD system stands for Email Phishing Attack Detection system. As ML with NLP is flexible and adaptive to learn by design, this combination can be more appropriate to analyze the changing patterns of email contents effectively to detect the phishing. Hence, the proposed EPAD system aims to detect the phished emails using ML and NLP techniques. The contributions of the proposed work lie in providing a structured benchmark and integrated feature set and not in a fundamentally new phishing email detection algorithm. These contributions are outlined below.

1. Curation of phishing email dataset, combining benign and phishing emails from multiple sources to make it diverse for comprehensive detection.
2. Detecting phishing emails by incorporating both content based as well as text based features. Readability scores, sentiments and attachment related information are also considered for the effective phishing email detection.
3. Comparative analysis of various ML and NLP text representation techniques based on their underlying accuracies, F1-scores, Area under Receiver Operating Characteristic Curve (ROC-AUC) and Matthews Correlation Coefficient (MCC) for phishing email attack detection.
4. Validation of results using ablation study on features, statistical performance analysis and sensitivity analysis.
5. Interpretation of the results of phishing email detection using Explainable AI (XAI).

2. Related Work

Phishing remains a major cybersecurity challenge. It has been one of the prevalent channels for intruders seeking unauthorized access to systems. Numerous studies have proposed different detection and prevention strategies, employing machine learning, deep learning, and hybrid modeling approaches to counteract phishing attempts [18].

Fuzzy rule based approaches of phishing detection are suggested by authors in [19-21]. However, high computational cost with increasing number of rules remains a challenge in these approaches. Data mining based approach proposed in [22] is limited in testing and validation. A phishing detection model is proposed by

the authors in [23] by applying the knowledge discovery and data mining techniques. In this model, over 16 features are considered from the email header and body. A novel phishing term weight has been used by the authors that capture high-frequency phishing vocabulary. However, the phishing-terms approach depends heavily on vocabulary frequency. Thus, it limits adaptation to evolving phishing language. The approach proposed in [24] makes use of different ML models for phishing detection. However, the approach focuses on detection of phishing URLs only.

Rawal, *et al.* in [14] proposed a system that uses only 9 content based features. These features are then trained on multiple classifiers to detect phishing emails. The authors achieve high accuracy with Support Vector Machine (SVM) and Random Forest (RF). The work in [15] employs Naive Bayes (NB) on real-world Gmail dataset for phishing detection using content based features. An approach is proposed by authors in [25] using 18 hybrid features extracted from header, URL, script and psychological indicator present in the email. Even though the authors in [14, 15, 25] achieve high accuracy, these approaches rely exclusively on small features set and limited dataset sizes.

Subsequent research in the area of email phishing emphasizes improving feature representation through advanced feature selection and optimization techniques. Phishing email detection approach is proposed in [26] to iteratively reduce dimensionality by replacing low-performing features by employing remove and replace technique. When combined with an ensemble of CART and C4.5 decision trees, the approach reduces the feature space from 47 to 11 attributes and achieves a good accuracy. Sonowal, *et al.,* in [27] make use of binary search for email feature selection with 41 content based features reduced to 37 features. This achieves higher accuracy in comparison with other feature selection methods. A technique is introduced in [28] to optimize NB classifier parameters for phishing detection. The work in [28] explains the benefits of metaheuristic optimization. Even though the techniques [26-28] reduce feature list and provide optimization, high computational cost is associated with memory and processing power. They are also prone to the risk of overfitting.

With the exposure to NLP techniques, recent research has focused on detecting emails as benign or phished by treating the problem as categorization of text. An approach proposed by [29] makes use of ML along with NLP and provides the benefit of context awareness. However, the proposed approach has the limitation of data dependency. Bountakas *et al.,* [16] have conducted a systematic ML based comparative study on email phishing detection techniques where different word embeddings have been used. In the work described in [16], when the dataset is balanced, Word2Vec combined with RF performs best, while when it is imbalanced, Word2Vec and Logistic Regression (LR) performs best. The study emphasizes the inadequacy of accuracy as a metric for imbalanced datasets. The experiments performed by Al Tawil *et al.,* in [30] use various NLP based feature extraction techniques and the work achieves good results across all evaluation metrics. However, the work proposed in [16, 30] requires higher computational resources and longer training times.

An approach is proposed in [31] where 26 NLP-based email features such as word count, punctuation usage, and uniqueness are considered. Seventeen models have been tested, and the most produced acceptable results. An approach proposed in [32] generates document embedding for emails using Doc2Vec. For this, the experiment is conducted on various dimensional spaces and achieves good results. Unnithan, *et al.,* in [33] use Term Frequency-Inverse Document Frequency (TF-IDF) features with domain specific email features achieving good performance with LR. The approaches [31-33] are mainly targeting email text, and no other parts of email are considered where obfuscation is possible. The approach proposed in [34] makes use of ML and TF-IDF for detection of phishing emails. Although the approach achieves good accuracy, the details of features are not provided.

The approach suggested by Alhogail and Alsabih in [35] introduces a graph based classifier - Graph Convolutional Networks (GCN) using TF-IDF to compute edges between words in the email. Although the model achieves good accuracy, it does not incorporate other email components such as attachments and URLs. Deep learning based techniques [36, 37] have also been suggested in literature for phishing detection. Deep learning has been used to capture semantic and contextual patterns in email in [38]. In this work, a feed-forward Neural Network proposed with 18 header and body features achieves high accuracy. However, the model lacks comparison with alternative classifiers. Also, it limits the generalizability of its claims. An approach is proposed in [39] for detection of phished mails by combining transformer-based contextual embeddings with sequence modeling through Long Short-Term Memory (LSTM). A transformer-based approach proposed by Patra et al., in [40] uses NLP and focuses on the semantics of the mail. Despite their effectiveness, the models proposed in [39, 40] require large datasets and intensive computational resources. The work in [41] suggests a deep learning based approach using dual layers. Although the approach gives good accuracy, the complexity of model is increased.

An approach based on SVM and Probabilistic Neural Network is proposed in [42] which integrates text and image features present in the email. The proposed approach compares the results with various classifiers like K-Nearest Neighbors (KNN), NB, RF, and LR. However, the scalability of the approach is limited due to

the use of a small dataset. A solution proposed by the Altan *et al.,* in [43] integrates transformer-based email text analysis with structural URL feature evaluation. The proposed approach effectively captures semantic patterns in the email message and processes the URL features using NLP techniques. However, it is constrained by a comparatively small dataset.

Ensemble methods have gained prominence for improving performance by combining multiple models for phishing email detection. Stacking ensemble models are proposed in [44, 45] using multiple classifiers for the classification of emails. An approach combining TF-IDF, SVM, and RF using soft voting, shows the effectiveness of ensemble-based fusion strategies for phishing email detection [46]. An approach proposed in [47] makes use of ensemble with Word2Vec model for phishing email detection. However, approaches [44-47] are computationally costly. The approach proposed in [6] uses ensemble learning- soft voting and stacking for detection of phishing emails. The proposed solution achieves high performance by integrating heterogeneous features. However, the use of highly imbalanced dataset causes the risk of majority class biases, and it also affects the generalization. A federated learning based approach proposed by [48] is scalable in nature. However, it incurs communication overhead in phishing detection.

Recent work explores the area of explainability techniques for providing transparency in the results of email phishing prediction. The authors in [49, 50] employ XAI to provide explanations for the phishing or benign email predictions made by the classification models. However, the approach proposed in [49] lacks the comparison with alternative classifiers. Whereas the approach proposed in [50] focuses only on the email text.

In summary, the above review of related work highlights the significance of addressing the problem of phishing emails in today's world. Table 1 provides the structured comparative analysis of above discussed phishing email detection approaches. While various approaches are proposed in the literature to detect phished emails, most of them depend upon limited features. To assess the authenticity of an email, the sentiments that it conveys, the suspiciousness of attachments and the readability of contents play a significant role. However, most of the reviewed approaches do not consider these aspects in the detection of phishing email. Consideration of both, content based and textual features is crucial in identifying the semantics and the structure of the email message to detect malicious mails. However, most of the existing approaches do not consider the combination of content based features with textual features for detecting phishing mails. From the end user's perspective, understanding the reasoning behind why a particular email is detected as a phished email or a benign one is significant to ensure the trustworthiness of the system's decision. However, this reasoning is not addressed by the majority of the approaches in literature.

The proposed EPAD system intends to address the limitations discussed above in the earlier work. The EPAD system performs the detection of phishing or benign emails using ML and NLP techniques. It makes use of a comprehensive set of email related features in the phishing detection. This feature set includes multiple content based features along with attachment-related information present in email. Various text based features along with readability scores and sentiments of email are also included in the EPAD system. Moreover, the EPAD system offers the reasoning about the prediction of phished or benign email by using SHapley Additive exPlanations (SHAP) as an explain ability technique.

3. Methodology

3.1 Functional Overview of EPAD System

Figure 1 depicts the functional overview of the proposed EPAD system. As shown in the figure, the EPAD system takes an email as input and determines if it is a benign or phished one. Additionally, it provides the reasoning about why an email has been detected as a phishing email or a benign one. The EPAD system incorporates various modules which are described below.

3.1.1 Email Parsing

Within this module, the system performs extraction of various components of the email. An email consists of a header and body. The header contains the metadata, while the body has actual message, HTML, hyperlink and attachment. So, by performing parsing on the email, decomposed components are stored in a structured matrix which is passed on to the next stage.

3.1.2 Content Based Analysis

The system analyzes the email content based on writing style and technical elements. It is assessed directly from the actual content of the email. This module inspects various elements of email such as URL properties, domain related features, suspicious keywords, and attachment types to analyze anomalies often used in phishing emails. This analysis is further elaborated in subsection 3.3.

3.1.3 Text Based Analysis

In this module text based analysis is performed for the email message by employing NLP techniques. Here, the email body is analysed to understand the morphological cues and semantic meaning.

Table 1. Comparative Analysis of Phishing Email Detection Techniques

Reference	Methodology	Dataset	Key Findings	Limitations
Bountakas & Xenakis (2023) [6]	Hybrid Ensemble learning: Soft voting, Stacking	Nazario, SpamAssassin, Enron	High performance	Complex
Rawal *et al.* (2017) [14]	ML – SVM, RF	Public sources	High accuracy	Small feature set and limited dataset size
Yang *et al.* (2019) [25]	SVM with hybrid features	monkey.org, CSDMC2010	Improved detection with good accuracy	Small feature set and limited dataset size
Hota *et al.* (2018) [26]	Ensemble of decision tree algorithms	Public source	High accuracy	Costly
Sonowal (2020) [27]	RF with Binary search Feature selection	Nazario, csmining group	High accuracy with reduced features	Information loss
Chakravorty *et al.* (2026) [29]	ML: NB, LR + NLP: TF-IDF	Kaggle, zenodo	Context aware	Data dependent
Gutiérrez *et al.* (2020) [32]	RF, SVM with Document embedding	Self-created	Semantics capture	Limited dataset size
Unnithan *et al.* (2018) [33]	ML: LR with TF-IDF	IWSPA-AP	Effective performance	Limited features
Alhogail & Alsabih (2021) [35]	ML + NLP: GCN with TF-IDF	CLAIR collection	Good accuracy	Limited to text features
Chinta *et al.* (2025) [39]	ML + Feature Engineering	Sources like SpamAssassin, UCI ML library	High performance	Need large datasets and computationally heavy
Patra *et al.* (2025) [40]	Transformer based, NLP	Nazario, Enron	Strong semantics	Computationally Costly
Rashed & Ozcan (2026) [41]	Deep learning	Email server, SpamAssassin	High accuracy	Complex
Kumar *et al.* (2020) [42]	Hybrid: SVM + NLP, Probabilistic Neural Network	Public sources	Better performance	Small dataset
Altan *et al.* (2025) [43]	Transformer based, NLP	CEAS 08, Nazario, SpamAssassin, Enron, Kaggle	High accuracy	Limited dataset size
Anirudh *et al.* (2024) [44]	Ensemble – SVM, XGBoost, LR	Kaggle	Robust	Resource heavy
Adnan *et al.* (2024) [45]	Ensemble Stacking approach	SpamAssassin, Enron	Better accuracy	Computationally costly
Ahmed & Sumesh (2026) [47]	Ensemble + Word2Vec, XAI	Enron, SpamAssassin, Nazario	Balanced performance	Computational overhead
Yi *et al.* (2026) [48]	Federated learning	Multiple like Nazario, Enron, SpamAssassin, IWSPA-AP	Handles imbalance	Communication overhead

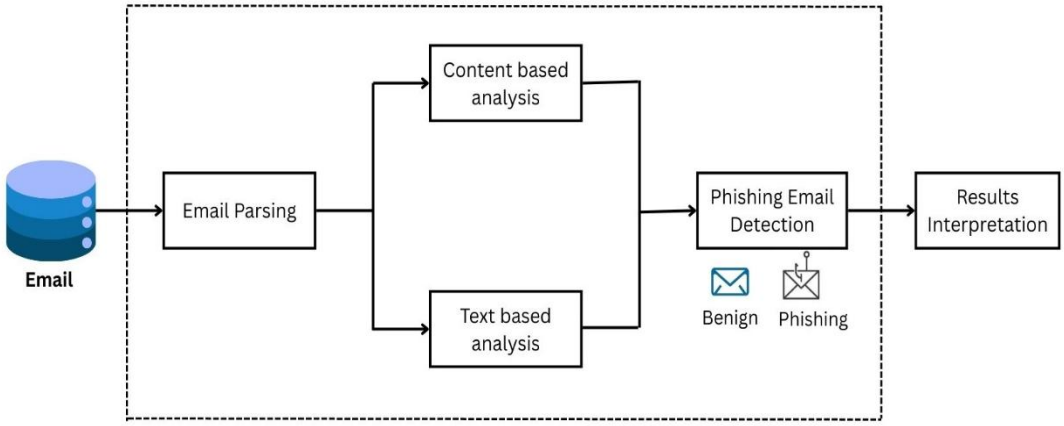

Figure 1. Functional overview of Email Phishing Attack Detection system

To capture these aspects, pre-processing of the contents of the email body has been done which is described in subsection 3.2. Further details of the text based analysis are described in subsection 3.3.

3.1.4 Phishing Email Attack Detection

Based on email content and text analysis, the phishing email detection module learns to evaluate email by finding patterns and indicators that can distinguish between benign and phished ones. The detection process is carried out by categorizing the email using the ML algorithm possessing optimal performance. Here, five algorithms namely, Support Vector Classifier (SVC), NB, LR, RF and eXtreme Gradient Boosting (XGBoost) are considered. The subsequent subsection describes the details of detection algorithms.

3.1.5 Results Interpretation

In this module, the results of phishing detection are interpreted using XAI. XAI is employed to make the EPAD system understandable to humans, revealing why it made specific predictions. Here, SHAP as an XAI technique has been employed which analyses feature importance and provides visualization for the predictions. The details of result interpretation are described in the ensuing subsection. The overall steps of phishing email detection are depicted by Algorithm 1 and are elaborated subsequently in the paper.

Algorithm 1. Email Phishing Detection

Input: Email dataset D

Output: Prediction of benign or phishing emails with reasoning

Begin

Step 1: Pre-process email messages to normalize the text

Step 2: Extract features from emails

a. Content based features: Keyword based indicators like urgency related words and suspicious terms, structural patterns like presence of HTML, links and forms, sentiment, readability and attachment related features

b. Text based features: Use of TF-IDF vectors and Word2Vec embeddings

Step 3: Integrate all the extracted features into a unified feature vector F

Step 4: Split the pre-processed dataset into training, validation and testing sets

Step 5: Train the phishing email detection algorithms with training set and unified feature vector F

Step 6: Evaluate the ability of trained algorithms to predict benign or phishing emails using various evaluation metrics

Step 7: Apply SHAP algorithm to interpret results of phishing email detection by analyzing the importance and impact of features

End

3.2 Data Pre-processing

Data pre-processing is required for textual feature extraction. The cleaning of the text in the email body starts with conversion of it into lowercase to standardize the data. Duplicate email messages are then identified and removed, where only the first occurrence of each distinct message is retained. This step ensures that the dataset does not contain redundant email messages, thus eliminating the possibility of bias in performance. This is followed by removing special characters, stop words and punctuation marks. HTML elements are removed if they appear. Hyperlink is replaced by a fixed string. Afterwards, tokenization is performed using NLTK's word tokenizer. To minimize the vocabulary size, lemmatization has been done for reducing words to their root forms. Lemmatization is performed using WordNet in NLP.

3.3 Feature Extraction

Feature extraction contributes significantly in phishing email detection. The EPAD system makes use of textual as well as content based features in the

identification of phishing mails. Email content inherently contains discriminative phishing information. In this work, content based features are extracted from specific components of email like headers, hyperlink, attachment or suspicious keywords for finding malicious patterns. The content based features which have been employed in this work are summarized in Table 2. Examples of urgency, money, login and threat related words referred in Table 2 are described below.

Table 2. Content based features list with description

Sr. No.	Feature	Description
1.	subject_is_present	This feature checks whether the subject is present or not.
2.	subject_re_present	It checks whether “Re:” is in the subject indicating it is a reply to a previous email.
3.	subject_num_words	Total word count in the subject
4.	subject_num_chars	Total character count in the subject
5.	subject_richness	This is the ratio of total word count to the total number of characters in the subject.
6.	subject_badword_present	It checks for bad words appearing in the subject of phishing emails.
7.	Body_num_chars	Total character count in the body
8.	Body_num_words	Total word count in the body
9.	Body_num_sentences	Total sentence count in the body
10.	Body_richness	This is the ratio of the total word count to the total number of characters in the body.
11.	Body_uppercase_ratio	This is the ratio of the total word count in uppercase and the total word count in the body.
12.	Body_urgency_count	It counts the urgency related words.
13.	Body_money_count	It counts the money related words.
14.	Body_login_count	It counts the login related words.
15.	Body_threat_count	It counts the threat related words.
16.	Body_sentiment	The score depicts the emotional tone of an email message
17.	Body_readability	The score determines the ease of understanding an email message
18.	html_present	It checks whether the HTML structure is present or not in the body.
19.	HasHTMLForms	It checks for HTML form in the email.
20.	HasImageLink	It checks whether a hyperlink is hidden behind an image.
21.	HasScript	It checks for the presence of the scripting code in the body.
22.	NumHyperlinks	Total count of hyperlinks in the email
23.	NumDifferentHREF	Total count for different HREF tags in an email
24.	HasTextHyperlink	This checks whether a hyperlink is hidden behind a human readable text.
25.	HasIPURL	It checks whether IP addresses are contained in the URLs.
26.	NumDots	Maximum number of dots in the URL.
27.	HasPortURL	It checks the presence of port in the URL.
28.	HasHexURL	This checks the presence of hex-encoded characters in the URL.
29.	MaxURLLength	This feature counts maximum URL length.
30.	CheckDomain	This feature checks if the sender's domain is different from hyperlink’s domain.
31.	HasAttachment	It checks the presence of attachment in the email.
32.	NumAttachments	Total count of attachments present in the email
33.	HasImage	Checks if an image file is attached. (for example .jpg, .png,.gif,.tiff)
34.	HasDocument	Checks if a document file is attached. (for example .pdf, .docx, .ppt)
35.	HasCompressed	Checks if a compressed archive is attached. (for example .zip, .rar, .tar, .gz)
36.	HasExecutable	Checks if any attachment has an executable file. (for example .exe, .bat, .bin, .msi, .jar)

- Urgency related words: urgent, immediately, now, expires, action required
- Money related words: money, account, transaction failed, payment, and invoice
- Login related words: login, sign in, reset, verify, authenticate, update security, two-factor
- Threat related words: suspend, block, deactivate, warning, failure, penalty, risk

The proposed system is not limited to only the above mentioned words in the respective categories. However, the above examples are given for the purpose of illustration.

Sentiment and readability score are also considered as part of content based features of an email as shown in Table 2. The compound sentiment score of email contents is calculated using Python's VADER utility from NLTK library because of its effectiveness in handling short messages like in emails. This compound score ranges from minus one to plus one, based on which the sentiment of the email message is depicted as positive, negative or neutral. Phishing email generally has a negative tone causing urgency, fear or threat. The readability score of an email message is calculated using the Flesch Reading Ease algorithm which assigns the scores from 0 to 100 to the email text. A higher readability score means it is easy to understand an email message.

The textual features are extracted using text representation techniques which capture semantic meaning of the email. This helps the EPAD system to effectively detect phishing email not just by the content but by its semantics. In this work, two text representation techniques are employed: TF-IDF [9] which is a frequency based text representation and Word2Vec [9] which is a prediction based word embedding. In the proposed system, TF-IDF transforms email text into numerical feature vectors by quantifying the discriminative importance of each word. TF-IDF computes the occurrence of a word appearing within a specific email and reduces the weight of highly frequent, non-informative words. This increases the weight of rarer and more informative terms. In the proposed system, Word2Vec captures semantic relationships between words of email by mapping words to high dimensional vectors. Words with similar meaning tend to lie in the near vicinity of vector space. The skip-gram architecture of Word2Vec which is used in this work predicts the context words given the target word and is better with rare and infrequent words.

3.4 Phishing Email Detection Algorithms

Phishing email detection distinguishes between benign and phishing mails. This is a binary classification problem. For performing this classification, various algorithms have been explored, as given below.

3.4.1 SVC

SVC is a supervised learning algorithm and is widely used for classification tasks. It functions by finding the optimal hyperplane for separating various classes. It maximizes distance for closest points of two different classes, here benign and phishing classes. It is particularly effective in high-dimensional spaces, such as text feature vectors from emails, where class boundaries can be complex. Also, SVC is robust to outliers. The proposed work uses a radial basis function (rbf) kernel that maps various features to a higher dimensional space. This kernel is used as it handles non-linear boundaries that separate the email classes.

3.4.2 NB

NB is a probabilistic classifier which makes use of Bayes's theorem to classify emails as phished or benign. It assumes that one feature is conditionally independent of the existence of another feature. This classifier calculates posterior probabilities for class based on feature likelihoods. It is fast and efficient and widely used for classification problems. Amongst three NB variants, namely Gaussian NB (GNB), Multinomial NB and Bernoulli NB, GNB model performs better during the experimentation in this work, and hence it is selected for further analysis.

3.4.3 LR

LR is a binary classifier used for the classification of emails as benign and phishing ones. It calculates the probability of input features associated with a specific class by using the sigmoid function. Sigmoid function maps output to probability between 0 and 1. LR is one of the widely used classifiers for classification tasks because of its simplicity and effectiveness with high-dimensional sparse features. In this work, the LR model is trained with 100 iterations so that it can learn properly from the email features. This enables the stable performance of the model in the detection process.

3.4.4 RF

RF works on a collection of decision trees. These decision trees have been constructed by various randomly selected subset features. The output of each decision tree is combined through voting to make the final decision so as to classify email as benign or phishing one. RF model is robust to noise and outliers. It offers high accuracy, reduces overfitting and works well with large datasets. In this work, the RF model is trained with 200 decision trees. This enables the model to better learn diverse phishing email patterns.

3.4.5 XGBoost

XGBoost is a classifier based on the gradient boosting framework. It is an optimized version of

gradient boosting. It builds decision trees in sequential order, where each new tree minimizes the errors of previous ones. It has the ability to capture subtle feature interaction and has built-in regularization that prevents the risk of overfitting. It provides high performance by capturing the complex non-linear patterns of the features on a large dataset. XGBoost is notable for its efficiency, speed and high performance, hence it is used in this work for the classification of emails. In this work, the XGBoost model is trained with 200 decision trees. This enables the model to capture complex phishing indicators in emails.

3.5 Explainability Algorithm for Results Interpretation

The EPAD system uses ML algorithms to detect phishing email attacks. The black-box nature of algorithms inherently lacks trust and interpretability. Hence, by using XAI, the impact of features on the working of the EPAD system has been analyzed. Also, the insights are gained about how phishing and benign emails are predicted. In this work, SHAP as an explainability algorithm has been employed which works with almost every ML algorithm being model-agnostic. SHAP is based on cooperative game theory that allocates SHAP values for all the features to show their contributions to the game. SHAP provides global as well as local explanations. In the context of the EPAD system, global explanation is the summary of feature importance across the email dataset while local explanation depicts which feature influences individual prediction. This dual capability makes SHAP particularly suitable for interpreting the results of phishing email attack detection in this work.

4. Results and Discussion

A prototype for the proposed EPAD system is built using Python and Jupyter notebook with Google Colab for experimentation and evaluation. This prototype makes use of an email dataset which has been curated for training and testing.

4.1 Dataset Curation

For the proposed system, datasets containing all the features required for the comprehensive assessment of emails for the detection of phishing are found not to be readily available. Therefore, for email phishing attack detection, the dataset is constructed by combining multiple publicly available and widely used labeled email corpora [9]. In this work, three distinct raw datasets are employed and merged to form a unified dataset. These datasets include the Enron email dataset, the SpamAssassin corpus and Nazario phishing corpus [9], containing phishing and benign emails. The Enron dataset consists of a collection of real-world corporate emails. This dataset is often used in email classification research because of the large scale authentic data that it contains. The SpamAssassin dataset is frequently used as a benchmark standard as it contains real-world diverse content in the form of bulk emails and legitimate emails. From the Enron dataset, 6000 emails are randomly selected to represent legitimate business communication. To increase the diversity of benign emails, additionally 6635 legitimate emails from SpamAssassin dataset are collected. After combining the selected samples of the Enron and the SpamAssassin datasets and removing duplicates from them, 9185 unique benign emails are obtained. For phishing class, multiple sources are combined to capture different attack patterns. Phishing emails are first collected from Nazario phishing corpus, which is a well-known benchmark dataset in cybersecurity research. From this dataset, to capture the evolution of phishing techniques over time, 3466 emails from the period 2015-2025 and 1879 emails before 2015 are taken into consideration.

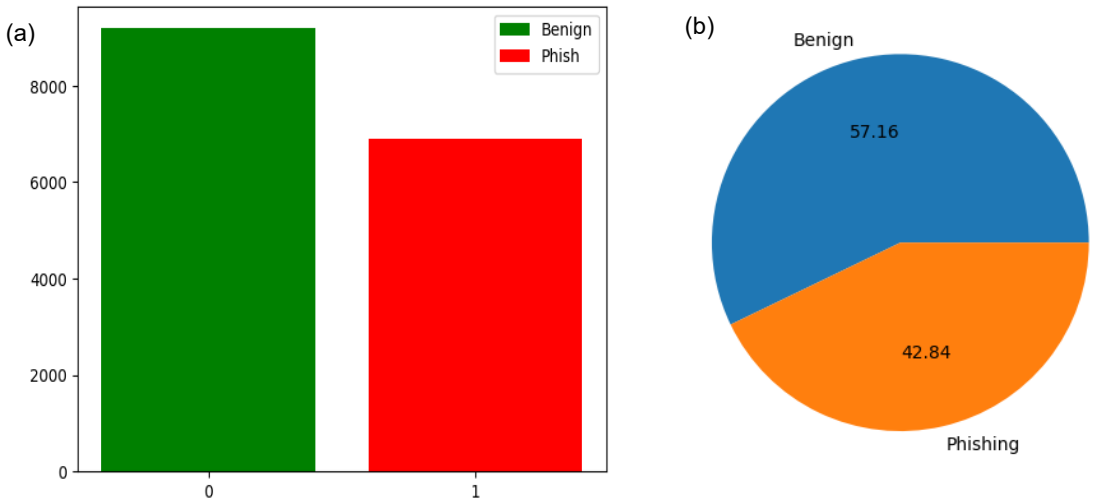

Figure 2. Dataset distribution: **(a)** Count of benign and phishing emails **(b)** Percentage distribution for benign and phishing emails.

In addition, the spam subset of SpamAssassin corpus has been used as a proxy for phishing emails, from which 3797 emails are obtained. After combining these two phishing datasets and performing deduplication, 6885 unique phishing emails are obtained. Thus, the final curated dataset contains total 16070 email samples where 9185 emails are benign and 6885 are phishing emails. This selected count of emails has been graphically shown in Figure 2 (a). The percentage distribution for benign and phishing emails in the dataset has been shown by Figure 2 (b). Figure 2 depicts that the curated dataset is quite balanced in terms of number of benign and phishing emails. This dataset has been split into 80:20 ratio so as to perform the training and testing respectively. Then the 80% portion in the earlier specified ratio has been further split into 70:10 ratio for training and validation respectively. That mean, 87.5% of the original 80% training portion is used for actual training and 12.5% of the original 80% training portion is used for validation.

4.2 Experimental Setup

During experimentation, the default threshold of 0.5 and the random seed of 42 have been used for the phishing email detection models which are described in Section 3.4. These models make use of various hyperparameters. Hyper parameters tuning has been carried out using Grid Search and the dataset split for training, validation and testing as described in section 4.1. During experimentation, various combinations of values of hyper parameters for each of the models are considered and the combination of values which gives the best performance is finalized for each model. The details of model-wise hyper parameters and their values are described below.

a) SVC:

C (Regularization) => {0.1, 1, 10} and kernel => {linear, rbf}

Best parameter values: C => 1 and kernel => rbf

b) NB:

Variants explored: GNB, Multinomial NB and Bernoulli NB

Best performing model: GNB with default values of parameters

c) LR:

C (Inverse regularization strength) => {0.1, 1, 10}, Maximum iterations => {100, 200}

Best parameter values: C => 0.1, Maximum iterations => 100

d) RF:

Number of estimators or decision trees => {100, 200}

Maximum depth => {None, 10, 20}

Best parameter values: Number of estimators => 200, Maximum depth => 20

e) XGBoost:

Number of estimators or decision trees => {100, 200}

Maximum depth => {3, 6} and Learning rate => {0.01, 0.1}

Best parameter values: Number of estimators => 200

Maximum depth => 6, learning rate => 0.1

The configuration of TF-IDF model has been decided by experimenting with different values of the parameter called maximum features. Three distinct values considered for maximum features are 100, 200 and 300. The best performance has been observed with the value of 300 for the maximum features parameter. Hence, TF-IDF vectorization has been performed using 300 maximum features. The minimum document frequency for TF-IDF has been set to 5 for eliminating the noise in the form of rare words or misspelled words.

During experimentation, vector sizes of 100, 200 and 300 have been evaluated for the Word2Vec model. The best performance has been observed for the vector size of 100, indicating sufficient semantic relationship without overfitting. The Word2Vec model has been trained with Skip-gram algorithm, vector size of 100, window size of 5 and minimum frequency of word as 2.

4.3 Evaluation Metrics

To evaluate the effectiveness of different algorithms used for phishing email detection, accuracy, F1-score, ROC-AUC and MCC are employed. These metrics provide insights on the performance of used algorithms and thereby help in selecting the best performing algorithm for the proposed system.

The confusion matrix provides a summary of classification performance including correctly and incorrectly predicted phishing and benign emails. Table 3 shows the confusion matrix, based on which performance metrics are derived.

Table 3. Confusion Matrix

Actual \ Predicted	**0 (Benign)**	**1 (Phishing)**
0 (Benign)	TN	FP
1 (Phishing)	FN	TP

True Negative (TN) denotes benign emails correctly classified as benign ones. True Positive (TP)

represents phishing emails correctly predicted as phishing ones. False Negative (FN) shows phishing emails incorrectly predicted as benign ones. False Positive (FP) depicts benign emails incorrectly predicted as phishing emails.

Accuracy represents the measure of correct prediction of whether it is a benign or a phishing email against the ground truth label. Precision indicates the ratio of count of correct prediction of phishing mails to the complete count of mails classified as phishing. Recall denotes the ratio of count of rightly identified phishing mails to the complete count of real phishing mails. F1-score is a measure obtained by merging precision and recall in proportion. The phishing emails being rightly predicted as phished ones and legitimate emails being rightly predicted as benign ones are equally important. Hence, in this work, instead of taking into account only precision and recall, F1-score as a metric is considered. ROC-AUC represents a measure which quantifies the ability of the model to distinguish between the positive class and the negative class where higher value of it indicates better capability of separation of classes. MCC represents a robust measure in phishing email detection which computes the correlation between actual and predicted classes. It gives a high value only when a model provides good results across all the four categories of confusion matrix.

4.4 Results and Analysis

For phishing email detection, different algorithms are explored as described in Section 3.4 to select the algorithm with the optimal results in terms of accuracy, F1-score, ROC-AUC and MCC. The experimentation is carried out with two distinct text representation techniques viz. TD-IDF and Word2Vec.

4.4.1 Assessment of Performance with TF-IDF

During the first set of experiments, TF-IDF technique is used for text representation and features based on content along with text are used for the detection of phishing email. For this setup, Table 4 depicts the results of different algorithms when applied for email phishing detection. The results from Table 4 demonstrate that the XGBoost algorithm achieves the best overall performance in terms of accuracy, F1-score, ROC-AUC and MCC. The confusion matrix of XGBoost model shows very low false positives and false negatives, indicating a balanced and reliable classifier. The results further imply that RF and SVC also perform competitively. RF provides slightly better recall value than SVC while SVC achieves higher precision, indicating fewer false positives. The results of LR reflect the baseline performance in prediction. GNB model illustrates relatively lower performance as it assumes the independence of features, whereas TF-IDF based features and content based features are correlated, which leads to higher false positives and false negatives. Overall, the results in Table 4 imply that when the XGBoost model with TF-IDF and content based features is applied for detection of phishing emails, it results in minimum error and the rate of rightly predicting the email as benign or phishing will be the highest.

4.4.2 Assessment of Performance with Word2Vec

During the second set of experiments, Word2Vec technique is used for text representation and features based on content along with text are used for the detection of phishing email. For this setup, Table 5 depicts the results of different algorithms when applied for email phishing detection. The results from Table 5 illustrate that the XGBoost algorithm attains the best overall performance in terms of accuracy, F1-score, ROC-AUC and MCC. SVC depicts closely following well-balanced performance with respect to accuracy and F1-score. RF model also performs well considering the accuracy and it maintains a good balance between precision and recall. LR model continues to provide the baseline performance in prediction. GNB model by characteristic gives the lowest performance. Overall, the results in Table 5 signify that when the XGBoost model with Word2Vec and content based features is used for detection of phishing emails, it results in minimum error and the rate of rightly predicting the email as benign or phishing will be the highest. This is clearly indicated by the confusion matrix of XGBoost model which shows very low false positives and false negatives.

4.4.3 Ablation Study

To systematically evaluate the effect of different categories of features on phishing email detection, an ablation study has been carried out using XGBoost, the best performing model. During this set of experiments, particular feature categories are chosen to be included or excluded to assess their impact on the performance of the model. Following are the categories of features used for the evaluation.

f1: Only content based features

f2: Only text-based features with TF-IDF vectors

f3: Only text-based features with Word2Vec embeddings

f4: Content based features excluding readability

f5: Content based features excluding sentiment

f6: Content based features excluding attachment-related features

f7: Full feature set – content based features and text-based features with TF-IDF vectors

f8: Full feature set – content based features and text-based features with Word2Vec embeddings

Table 6 shows the results of phishing email detection using XGBoost with different feature categories.

Table 4. Comparative results for phishing email detection with TF-IDF

Model	Accuracy	Precision	Recall	F1-score	ROC-AUC	MCC	Confusion matrix
SVC	0.9813	0.9895	0.9665	0.9779	0.9968	0.9619	[1823 14] [46 1331]
GNB	0.9175	0.8943	0.9157	0.9049	0.9573	0.8323	[1688 149] [116 1261]
LR	0.9726	0.9721	0.9636	0.9679	0.9953	0.9440	[1799 38] [50 1327]]
RF	0.9813	0.9845	0.9716	0.9780	0.9974	0.9618	[1816 21] [39 1338]
XGBoost	0.9844	0.9867	0.9767	0.9817	0.9976	0.9682	[1819 18] [32 1345]

Table 5. Comparative results for phishing email detection with Word2Vec

Model	Accuracy	Precision	Recall	F1-score	ROC-AUC	MCC	Confusion matrix
SVC	0.9856	0.9868	0.9796	0.9832	0.9973	0.9707	[1819 18] [28 1349]]
GNB	0.9156	0.9381	0.8598	0.8973	0.9687	0.8282	[1759 78] [193 1184]
LR	0.9744	0.9757	0.9644	0.9700	0.9949	0.9478	[1804 33] [49 1328]]
RF	0.9810	0.9852	0.9702	0.9776	0.9976	0.9612	[1817 20] [41 1336]]
XGBoost	0.9866	0.9904	0.9782	0.9842	0.9977	0.9726	[1824 13] [30 1347]]

Table 6. Comparative results of phishing email detection for XGBoost with different feature categories

Feature category	Accuracy	Precision	Recall	F1-score	ROC-AUC	MCC	Confusion matrix
f1	0.9713	0.9776	0.9549	0.9662	0.9942	0.9415	[1807 30] [62 1315]
f2	0.9467	0.9533	0.9208	0.9368	0.9887	0.8912	[1775 62] [109 1268]
f3	0.9732	0.9680	0.9694	0.9687	0.9950	0.9453	[1793 44] [42 1335]
f4	0.9691	0.9754	0.9520	0.9636	0.9939	0.9371	[1804 33] [66 1311]
f5	0.9688	0.9747	0.9520	0.9632	0.9932	0.9364	[1803 34] [66 1311]
f6	0.9713	0.9776	0.9549	0.9662	0.9942	0.9415	[1807 30] [62 1315]
f7	0.9844	0.9867	0.9767	0.9817	0.9976	0.9682	[1819 18] [32 1345]
f8	0.9866	0.9904	0.9782	0.9842	0.9977	0.9726	[1824 13] [30 1347]

The results in Table 6 show a clear trend concerning the importance of feature categories and the correlation amongst them. When only content based features (f1) are used, the model achieves high performance in respect of accuracy, F1-score, ROC-AUC and MCC. This indicates the discriminative capability of structural and metadata related features. Whereas, when only text based features are used with TF-IDF vectors (f2), performance lowers down clearly, indicating that TF-IDF vectors solely may not be able to capture deeper contextual relationships in the text. On the other hand, use of text-only features with Word2Vec embeddings (f3) provides considerably better performance in the form of higher values of accuracy, F1-score, ROC-AUC and MCC.

This underlines the effectiveness of semantic representations in capturing contextual and syntactic patterns applicable to phishing detection. Further, as seen by the results in Table 6, excluding readability feature (f4) causes a decrease in performance, implying that writing complexity and styles matter significantly in identifying phishing mails. On similar lines, exclusion of sentiment feature (f5) causes a slight degradation of performance. This signifies that emotional tone which is frequently related with urgency in phishing emails provides a useful measure in phishing detection. Elimination of attachment related features (f6) also affects performance, indicating their importance in detection of suspicious behaviour associated with malicious attachments.

When all the feature categories are combined, a significant enhancement in performance is observed. Using content based features and text based features with TF-IDF vectors (f7) results into high values for all the evaluation metrics. This demonstrates the advantage of combining diverse feature types into integrated representation. The overall best performance is depicted by the full feature set with Word2Vec embeddings (f8) giving the highest values for all the evaluation metrics. This implies that the results of combining content based features and text-based features with Word2Vec embeddings indicate a promising internal benchmark and this combination makes it possible for the model to capture structural as well as contextual characteristics more efficiently.

Overall, the ablation study clearly shows that no single feature category is sufficient to achieve optimal performance in isolation. Instead content based and textual features complement each other, whereas readability, sentiment and attachment related features provide important incremental improvements.

4.4.4 Statistical Performance Analysis across multiple runs

To assess the robustness and stability of phishing email detection models, experiments are repeated by employing five distinct random seeds, viz. 42, 52, 62, 72, and 82. The top three models as demonstrated by the results in Table 4 and Table 5 are considered for the repetitive experiments. In the first set of repetitive experiments, content based features and text-based features with TF-IDF vectors are considered. Whereas, in the second set of repetitive experiments, content based features and text-based features with Word2Vec embeddings are considered. The results of statistical performance analysis are reported as mean ± standard deviation across all evaluation metrics for top three models in both the experiments. Here, mean implies the average value of the respective evaluation metric and standard deviation indicates the measure of its stability.

The results in Table 7 show that when the content based features are combined with TF-IDF features, all the models demonstrate consistently high performance with very low standard deviation values, indicating strong stability across different runs. Among them, XGBoost achieves the best overall results with regard to accuracy, F1-score, ROC-AUC and MCC, reflecting both high predictive capability and reliable behavior across varying initializations. RF shows comparable but slightly lower performance while maintaining stable results across all metrics. SVC achieves the highest precision, indicating fewer false positives, however its relatively lower recall suggests a tendency to miss some phishing instances, which slightly reduces its overall F1-score.

The results in Table 8 show further enhanced performance and stability when content based features are combined with Word2Vec embeddings. The standard deviation values are generally lower than those observed in Table 7. Again, XGBoost attains the best overall results with highest values of accuracy, F1-score, ROC-AUC and MCC. With Word2Vec representation, SVC also achieves competitive accuracy, F1-score and MCC and performs better than RF. whereas, the results of RF indicate that it may be less effective in utilizing semantic embeddings.

Overall, the consistently low standard deviation values across the two feature representations confirm that models are highly reproducible and not sensitive to random seed variations.

Table 7. Results of Statistical Analysis with TF-IDF

Model	Accuracy	Precision	Recall	F1-score	ROC-AUC	MCC
XGBoost	0.9829 ± 0.0022	0.9852 ± 0.0012	0.9747 ± 0.0043	0.9799 ± 0.0026	0.9975 ± 0.0007	0.9651 ± 0.0045
RF	0.9813 ± 0.0011	0.9834 ± 0.0021	0.9727 ± 0.0028	0.9780 ± 0.0013	0.9976 ± 0.0005	0.9618 ± 0.0023
SVC	0.9795 ± 0.0028	0.9874 ± 0.0027	0.9644 ± 0.0054	0.9757 ± 0.0034	0.9968 ± 0.0009	0.9582 ± 0.0057

Table 8. Results of Statistical Analysis with Word2Vec

Model	Accuracy	Precision	Recall	F1-score	ROC-AUC	MCC
XGBoost	0.9861 ± 0.0011	0.9870 ± 0.0005	0.9804 ± 0.0031	0.9837 ± 0.0014	0.9984 ± 0.0003	0.9715 ± 0.0023
RF	0.9808 ± 0.0030	0.9831 ± 0.0029	0.9720 ± 0.0044	0.9775 ± 0.0035	0.9981 ± 0.0005	0.9609 ± 0.0061
SVC	0.9846 ± 0.0017	0.9861 ± 0.0019	0.9779 ± 0.0037	0.9820 ± 0.0020	0.9981 ± 0.0005	0.9686 ± 0.0035

The results in Table 7 and Table 8 confirm XGBoost as the most reliable model and the model attains the highest performance across multiple runs. The results also highlight that use of Word2Vec embeddings over TF-IDF vectors provides consistent improvement in performance in terms of all the evaluation metrics.

4.4.5 Threshold Sensitivity Analysis

To analyze the effect of decision threshold selection on classification behavior, experiments are conducted using distinct probability thresholds of 0.3, 0.5, and 0.7. The effect of changes in threshold is assessed using confusion matrices for XGBoost being the best performing model, with both the combinations, viz. content based features with TF-IDF and content based features with Word2Vec.

The results in Table 9 show that for XGBoost with TF-IDF, if threshold is decreased to 0.3, it increases the number of samples predicted as positive. This results in more true positives, lesser false negatives and increase in false positives. This implies that recall is emphasized over precision where the model concentrates on detecting the phishing emails at the cost of increase in the false alarms. When the threshold is set to default value of 0.5, it decreases false positives with reasonable increase in false negatives. Thus the model demonstrates a better balancing between precision and recall. When the threshold is increased to 0.7, false positives are considerably decreased but false negatives are further increased. This implies that even if precision is enhanced, it adversely affects the recall.

The results in Table 9 depict a similar pattern with slightly enhanced performance for XGBoost with Word2Vec embeddings. When the threshold is set to 0.3, the model attains the highest true positives, minimum false negatives and maintains a comparatively small number of false positives. This implies that the model with Word2Vec reflects a better sensitivity as compared to the one with TF-IDF. When the threshold is set to default value of 0.5, a more balanced performance in terms of precision and recall is depicted by the model with reasonably smaller false positives and false negatives. When the threshold is increased to 0.7, false positives are considerably decreased but false negatives are further increased. This again indicates that the precision is emphasized as compared to the recall.

Overall, with the threshold value of 0.5, the results demonstrate the balanced performance with no considerable increase in false positives or false negatives. Thus the threshold of 0.5 is identified to be the most suitable one in this work which attains an effective trade-off between precision and recall.

4.4.6 Comparison of Results with Previous Approaches

A comparison is made for the proposed EPAD system's performance with the other [25, 27, 32, 33, 35, 44] approaches on email phishing detection. Table 10 shows the comparison of these approaches based on accuracy and F1-score. Table 10 indicates that each of the email phishing detection approaches makes use of distinct models for the detection. Also, as described in Table 1, these approaches make use of different datasets for the purpose of testing. The results in Table 10 illustrate that the proposed EPAD system which includes XGBoost with Word2Vec model, achieves competitive results within its own benchmark in terms of accuracy and F1-score. Also, unlike the other email phishing detection approaches, EPAD system includes content based and textual features, readability and sentiment scores, attachment related attributes along with the results interpretation using explainable AI.

4.4.7 Interpretation of Results of phishing email detection using Explainability Algorithm

From the experimental results, it is observed that XGBoost with Word2Vec as well as with TF-IDF demonstrate better performance in comparison with the other models of email phishing detection. Although Word2Vec is suitable to capture semantic relationships among the words, TF-IDF is more suitable when interpretation of the decisions is required for the end-users of the system, where users can directly understand the reasoning behind the phishing email prediction.

Table 9. Results of phishing email detection for XGBoost with different threshold values

Threshold / Mode	**0.3**	**0.5**	**0.7**
XGBoost with TF-IDF	[1808 29] [23 1354]	[1819 18] [32 1345]	[1828 9] [47 1330]
XGBoost with Word2Vec	[1815 22] [19 1358]	[1824 13] [30 1347]	[1828 9] [31 1346]

Table 10. Comparison of results with previous approaches

Reference	**Technique**	**Accuracy (%)**	**F1-score (%)**	**Use of features - Readability, Sentiment, Attachment**	**Feature extraction techniques**	**Interpretability of results**
Yang *et al.* (2019) [25]	SVM	95	95.2	Not considered	Content based	Not considered
Sonowal (2020) [27]	RF	97.41	97.78	Readability scores are considered	Content based	Not considered
Gutiérrez *et al.* (2020) [32]	RF	91.6	90	Not Considered	Text based (Doc2Vec)	Not considered
Unnithan *et al.* (2018) [33]	LR	97	98	Not considered	Text based (TF-IDF) and domain features	Not considered
Alhogail & Alsabih (2021) [35]	GCN	98.2	98.55	Not considered	Text Based (TF-IDF)	Not considered
Anirudh *et al.* (2024) [44]	SVM, XGBoost, RF	97.92	98.32	Not Considered	Text based (TF-IDF)	Not considered
Proposed System: EPAD	XGBoost with Word2Vec	98.66	98.42	Readability, Sentiment, Attachment attributes are considered from the email	Content based and Text based (TF-IDF and Word2vec)	Results are interpreted using Explainable AI - SHAP

This is because it provides direct inspection of discriminative terms contributing to phishing classification decisions that are easily understandable, which is not possible with Word2Vec embedding. Hence, the XGBoost and TF-IDF offers the right combination to apply an explainability algorithm for the interpretation of email phishing results. Thus, the explainability algorithm - SHAP is applied on the XGBoost model, where text is represented using TF-IDF and features based on content along with text are considered. SHAP makes phishing attack detection interpretable when black box model like XGBoost has been used. It provides insights on the features which help to make the final decision. SHAP algorithm when applied to the phishing email detection model, provides various types of graphical visualizations to understand and interpret the behaviour of the system.

Figure 3 demonstrates the summary of results of SHAP using a beeswarm plot for phishing email detection model. It provides a global view of feature importance and the manner in which individual feature values influence the model's decisions. Here, ranking of features is done based on corresponding mean absolute SHAP values. The features at the top are the most influential features, indicating their overall importance in the decision making process. The maximum number of displayed features is set to 30 for better interpretability because of the large size of the feature set considered for the dataset. The x-axis shows SHAP values which quantify the impact of features on the prediction. By default, SHAP values are computed in log-odds space. The binary classes are encoded as class 1 for phishing and class 0 for benign.

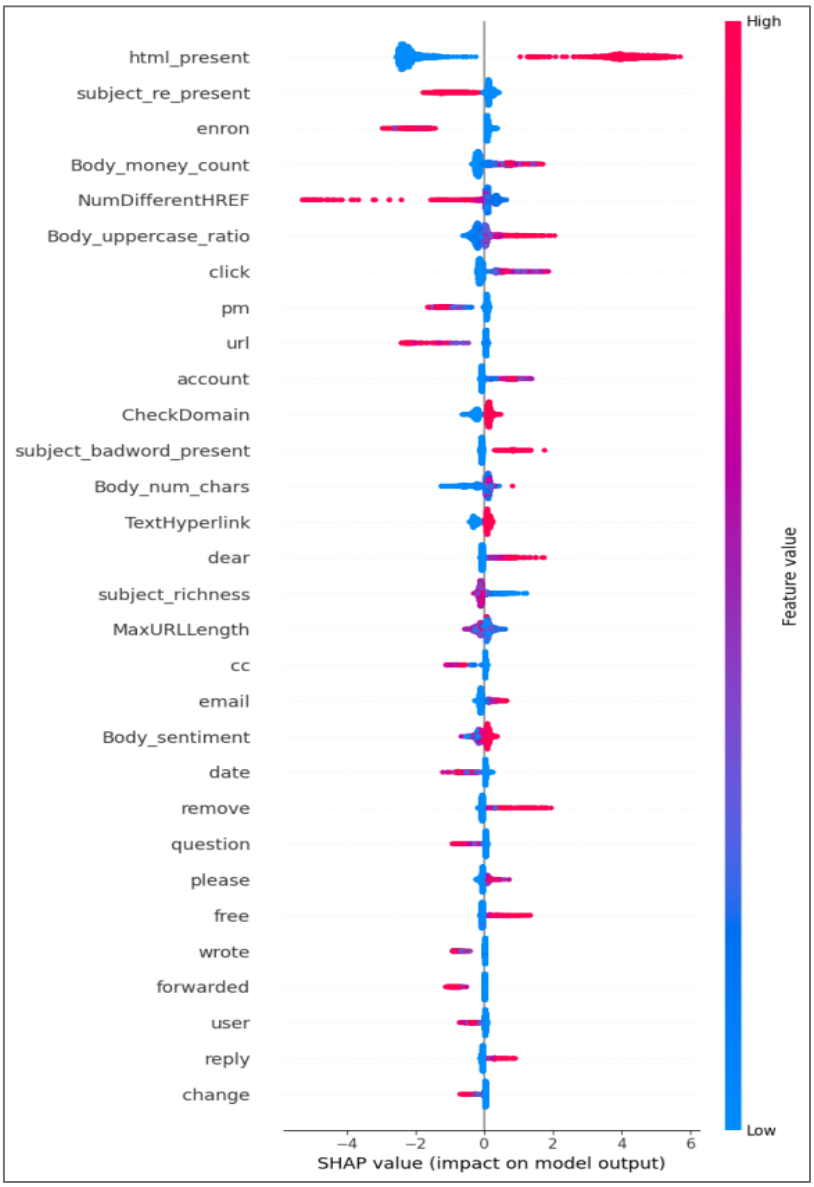

Figure 3. Summary of Results of SHAP Explainability Algorithm for XGBoost-TF-IDF model

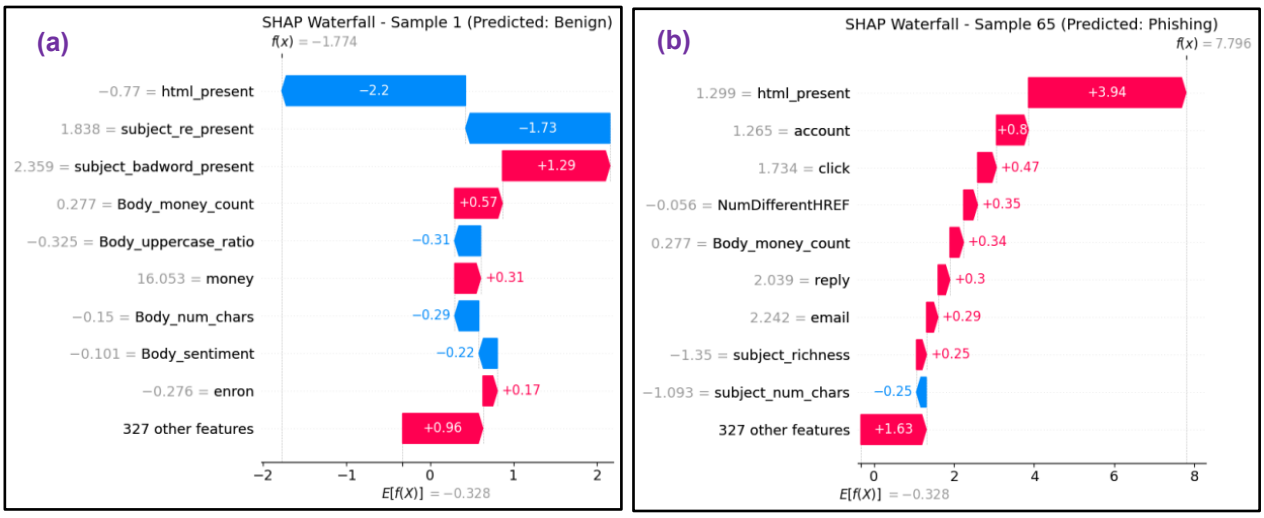

Figure 4(a). SHAP waterfall plot for benign prediction, **(b)** SHAP waterfall plot for phishing prediction

The positive SHAP values drive towards phishing prediction whereas the negative SHAP values drive to benign predictions. Each point shows the actual value of the feature using the color gradient with high value shown as red while low value as blue. This gradient helps to identify whether higher or lower value of a feature increases likelihood of phishing. The spread of SHAP values across both positive and negative regions for several features shows the non-linear feature interaction, demonstrating that their impact varies across different email contexts. From the plot it implies that html_present, subject_re_present, Body_money_count, NumDifferentHREF and Body_uppercase_ratio along with interaction-related words such as click, url and account have the strongest influence on the model's decision. Also, the CheckDomain feature captures domain mismatches, which further contributes to phishing detection. Content based features such as html_present, Body_uppercase_ratio and subject_badword_present positively influence phishing predictions, suggesting that emails which excessively use HTML formatting, and uppercase text or suspicious words in the subject are more prone to get classified as phishing emails. Similarly, the presence of monetary terms denoted by the Body_money_count feature increases the chances of phishing predictions. On the other hand, the presence of "Re:" in the subject line represented by subject_re_present feature generally drives towards benign predictions, indicating a kind of legitimate email communication. This plot highlights the importance of content as well as text based features for phishing detection.

SHAP waterfall plot illustrates an explanation for individual predictions made by the model. Figure 4 (a) and Figure 4 (b) represent the SHAP waterfall plots for individual predictions made on email samples. Here features are delimited to show top 10 features that drive the phishing email predictions. The waterfall plot starts by average model output E[f(x)] as base value which represents the average predictions over the dataset. The colored bars show the SHAP magnitude and direction of the feature. Blue bars indicate negative SHAP values supporting benign email classification. Red bars indicate positive SHAP values supporting phishing email classification. f(x) represents the final model's output for specific instances. Final prediction (f(x)) is obtained by adding all feature contributions that is E[f(x)] and SHAP values for all features.

Figure 4(a) depicts the SHAP waterfall plot for a benign email prediction made by the phishing detection model for the selected instance. The waterfall plot reveals that certain features have a dominant influence on prediction. Features like html_present, subject_re_present, Body_uppercase_ratio, Body_num_chars and Body_sentiment exhibit strong negative contribution, indicating that they contribute towards the benign class. Whereas, the features like subject_badword_present and Body_money_count exhibit positive contributions, showing that they contribute towards phishing class. In spite of the presence of some phishing-indicative signals, the strong negative influence of features outweighs these effects, resulting in a final benign classification. Overall by the summation of the feature contribution, the output is f(x) = -1.774 compared to the base value E[f(x)] = -0.328, reflecting a prediction as benign email.

Figure 4(b) depicts the SHAP waterfall plot for a phishing email prediction made by the phishing detection model for the selected instance. The plot shows that html_present feature makes the highest positive contribution in the prediction. Also, features like NumDifferentHREF, Body_money_count, subject_richness and words like account, click, reply and email contribute positively towards phishing classification. The bars corresponding to all these features are red in color and have positive values. The subject_num_chars feature provides a slight negative contribution giving a notion of being benign. The combined effect of the remaining 327 features supports a phishing classification. After summing all contributions, the output is f(x) = 7.796 compared to the base value E[f(x)] = -0.328, which clearly indicates that the email is classified as phishing.

The waterfall plots shown in Figure 4 (a) and Figure 4 (b) depict that content based and text based features both help in making decisions. Predictions made by the model are explained to end users, highlighting its ability to interpret and distinguish phishing and benign mails based on contribution of features.

4.5 Discussion

The proposed EPAD system has been validated with different ML models and two text representation techniques viz. TF-IDF and Word2Vec to identify models with promising results within internal benchmarks. Content based as well as textual features are considered during the evaluation. During experimentation, three distinct and benchmark datasets are considered to include the variety of samples. Also the corpus age includes the period before 2015 as well as the period 2015-2025 to cover both, the earlier as well as the recent patterns of emails.

Moreover, ablation study is carried out to identify the impact of individual category of features on the phishing email detection. Statistical analysis over multiple runs is performed to evaluate the reproducibility and stability of models. Sensitivity analysis with different threshold values is also carried out to further assess the robustness of the XGBoost model with TF-IDF and Word2Vec. From the SHAP based explainability analysis, it is observed that content based and textual features are equally important in detecting phishing emails.

Although the proposed work has included above all dimensions, there are few limitations of the study. The work is limited only to the internal validation of results. The external validation to assess the dynamic patterns of email such as attachment heavy campaigns in real life application domains is also needed. Also, the current work does not consider the testing of situations like possible parsing failure modes and practical deployment trade-offs.

5. Conclusion

This paper presents a system using ML algorithms along with NLP strategies for phishing detection which classifies an email as benign or phishing one. The contributions of the proposed EPAD system lie in providing a structured benchmark and integrated feature set and not in a fundamentally new phishing email detection algorithm. The EPAD system focuses on both, content based and textual features by analyzing email structure. Moreover, the use of key aspects which include readability scores, sentiment polarity and attachments related attributes are the unique features of EPAD system which contribute in the effective detection of phishing email. For the purpose of experimentation, a dataset of emails is curated with a comprehensive set of features. The EPAD system also offers the interpretation of results of phishing mail detection by using SHAP as an explainability algorithm.

In phishing mail detection, comparative evaluation of various ML algorithms is made with TF-IDF and Word2Vec text representation techniques to identify the algorithm with better performance. Experimental results have demonstrated that for phishing email detection, the performance of XGBoost model is better as compared to other algorithms concerning accuracy, F1-score, ROC-AUC and MCC. From the results it is further depicted that the performance of both, XGBoost with TF-IDF and XGBoost with Word2Vec are at par. The EPAD system which includes XGBoost with Word2Vec model, achieves competitive results within its own benchmark in terms of accuracy (98.66%) and F1-score (98.42%).

The ablation study on features has shown that the full feature set that is content based and textual features with Word2Vec embeddings depicts the overall best performance and the results indicate a promising internal benchmark. Evaluations in the form of statistical performance analysis and threshold sensitivity analysis have indicated the robustness and reproducibility of the models. Interpretation of results of phishing email detection using explainability algorithm - SHAP has demonstrated that features based on content and text, both are important and they contribute equally for efficiently detecting phishing emails.

This study motivates further directions as the future work. In future, the external validation of the system can be carried out, which is not considered in the current work. Enhancing the dataset further and experimenting with different combinations of training-testing splits can be taken up. Cross-corpus validation, uncertainty reporting and multilingual phishing email detection can also be considered as next important steps.

References

[1] S. Saeed, N.Z. Jhanjhi, M.A. Khan, D.K. Yadav, Digital transformation and cybersecurity challenges. Frontiers in Computer Science, 7, (2025) 1631362. https://doi.org/10.3389/fcomp.2025.1631362

[2] B. Alotaibi, Cybersecurity attacks and detection methods in web 3.0 technology: A review. Sensors, 25(2), (2025) 342. https://doi.org/10.3390/s25020342

[3] I. Naseer, The role of artificial intelligence in detecting and preventing cyber and phishing attacks. European Journal of Engineering Science and Technology, 11, (2024) 82-86.

[4] A. Jadhav, P.R. Chandre, Survey and comparative analysis of phishing detection techniques: current trends challenges and future directions. IAES International Journal of Artificial Intelligence (IJ-AI), 14(2), (2025) 853-866. http://doi.org/10.11591/ijai.v14.i2.pp853-866

[5] M. Abdolrazzagh-Nezhad, N. Langarib, Phishing detection techniques: a review. Data Science: Journal of Computing and Applied Informatics, 9(1), (2025) 32-46. https://doi.org/10.32734/jocai.v9.i1-19904

[6] P. Bountakas, C. Xenakis. Helphed: Hybrid Ensemble Learning Phishing Email Detection. Journal of network and computer applications, 210, (2023) 103545. https://dx.doi.org/10.2139/ssrn.4147334

[7] D. Sturman, J.C. Auton, B.W. Morrison, Security awareness, decision style, knowledge, and phishing email detection: Moderated mediation analyses. Computers & Security, 148, (2025) 104129. https://doi.org/10.1016/j.cose.2024.104129

[8] F. Carroll, J.A. Adejobi, R. Montasari, How good are we at detecting a phishing attack? Investigating the evolving phishing attack email and why it continues to successfully deceive society. SN Computer Science, 3(2), (2022) 170. https://doi.org/10.1007/s42979-022-01069-1

[9] S. Salloum, T. Gaber, S. Vadera, K. Shaalan. A Systematic Literature Review on Phishing Email Detection Using Natural Language Processing Techniques. IEEE Access, 10, (2022) 65703-65727. https://doi.org/10.1109/ACCESS.2022.3183083

[10] D. Hillman, Y. Harel, E. Toch, Evaluating

organizational phishing awareness training on an enterprise scale. Computers & Security, 132, (2023) 103364. https://doi.org/10.1016/j.cose.2023.103364

[11] M. Alanezi, Phishing Detection Methods: A Review. Applied Sciences and Technology, 3(9), (2021). https://doi.org/10.47577/technium.v3i9.4973

[12] M. Khonji, Y. Iraqi, A. Jones, Phishing detection: a literature survey. IEEE Communications Surveys & Tutorials, 15(4), (2013) 2091-2121. https://doi.org/10.1109/SURV.2013.032213.00009

[13] S. Paliwa, D. Anand, S. Khan, Application of Rule based Fuzzy Inference System in Prediction of Internet Phishing. International Journal of Computer Applications, 148(14), (2016) 30-37. https://doi.org/10.5120/ijca2016911334

[14] S. Rawal, B. Rawal, A. Shaheen, S. Malik. Phishing detection in e-mails using machine learning. International Journal of Applied Information Systems, 12(7), (2017) 21-24. https://doi.org/10.5120/ijais2017451713

[15] S.B. Rathod, T.M. Pattewar, (2015) Content based spam detection in email using Bayesian classifier. International Conference on Communications and Signal Processing (ICCSP), IEEE, India. https://doi.org/10.1109/ICCSP.2015.7322709

[16] P. Bountakas, K. Koutroumpouchos, C. Xenakis. A comparison of natural language processing and machine learning methods for phishing email detection. In Proceedings of the 16th International Conference on Availability, Reliability and Security, 127, (2021) 1-12. https://doi.org/10.1145/3465481.3469205

[17] S. Kavya, D. Sumathi. Staying ahead of phishers: a review of recent advances and emerging methodologies in phishing detection. Artificial Intelligence Review, 58(2), (2024) 50. https://doi.org/10.1007/s10462-024-11055-z

[18] D. Rathee, S. Mann. Detection of E-mail phishing attacks–using machine learning and deep Learning. International Journal of Computer Applications, 183(1), (2022) 1-7. https://doi.org/10.5120/ijca2022921868

[19] M. Moghimi, Varjani, A.Y. New rule-based phishing detection method. Expert systems with applications, 53, (2016) 231-242.

[20] M. Almseidin, M. Alkasassbeh, M. Alzubi, J. Al-Sawwa, Cyber-phishing website detection using fuzzy rule interpolation. Cryptography, 6(2), (2022) 24. https://doi.org/10.3390/cryptography6020024

[21] S.A. Khan, K. Iqbal, N. Mohammad, R. Akbar, S.S.A. Ali, A.A. Siddiqui, A novel fuzzy-logic-based multi-criteria metric for performance evaluation of spam email detection algorithms. Applied Sciences, 12(14), (2022) 7043. https://doi.org/10.3390/app12147043

[22] S. Sentürk, E. Yerli, İ. Sogukpınar, (2017) Email phishing detection and prevention by using data mining techniques. International Conference on Computer Science and Engineering (UBMK), IEEE, Turkey. https://doi.org/10.1109/UBMK.2017.8093510

[23] A. Yasin, A. Abuhasan, (2016) an intelligent classification model for phishing email detection. arXiv preprint arXiv:1608.02196. https://doi.org/10.48550/arXiv.1608.02196

[24] A. Karim, M. Shahroz, K. Mustofa, S.B. Belhaouari, S.R.K. Joga, Phishing detection system through hybrid machine learning based on URL. Ieee Access, 11, (2023) 36805-36822. https://doi.org/10.1109/ACCESS.2023.3252366

[25] Z. Yang, C. Qiao, W. Kan, J. Qiu, (2019) Phishing email detection based on hybrid features. In IOP Conference series: earth and environmental science, 252(4), 042051. https://doi.org/10.1088/1755-1315/252/4/042051

[26] H.S. Hota, A.K. Shrivas, R. Hota. An ensemble model for detecting phishing attack with proposed remove-replace feature selection technique. Procedia computer science, 132, (2018) 900-907. https://doi.org/10.1016/j.procs.2018.05.103

[27] G. Sonowal, Phishing email detection based on binary search feature selection. SN Computer Science 1(4) (2020) 191. https://doi.org/10.1007/s42979-020-00194-z

[28] K. Agarwal, T. Kumar, (2018) Email spam detection using integrated approach of Naïve Bayes and particle swarm optimization. In 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), IEEE, India. https://doi.org/10.1109/ICCONS.2018.8662957

[29] A. Chakravorty, M. Price, N. Elsayed, Z. ElSayed, (2026) Context-Aware Phishing Email Detection Using Machine Learning and NLP. arXiv preprint arXiv:2603.27326. https://doi.org/10.48550/arXiv.2603.27326

[30] A. Al Tawil, L. Almazaydeh, D.S. Qawasmeh, B. Qawasmeh, M. Alshinwan, K. Elleithy, Comparative Analysis of Machine Learning Algorithms for Email Phishing Detection Using TF-IDF, Word2Vec, and BERT. Computers Materiais Continua, 81(2), (2024) 3395-3412. http://dx.doi.org/10.32604/cmc.2024.057279

[31] G. Egozi, R. Verma, (2018) Phishing email detection using robust nlp techniques. IEEE International Conference on Data Mining Workshops (ICDMW), IEEE, Singapore. https://doi.org/10.1109/ICDMW.2018.00009

[32] L.F. Gutiérrez, F. Abri, M. Armstrong, A.S. Namin, K.S. Jones, (2020) Email embeddings for phishing detection. In 2020 ieee international

conference on big data (big data), IEEE, USA. https://doi.org/10.1109/BigData50022.2020.9377821

[33] N.A. Unnithan, N.B. Harikrishnan, S. Akarsh, R. Vinayakumar, K.P. Soman. Machine learning based phishing e-mail detection. Security-CEN@ Amrita (2018) 65-69.

[34] N.B. Harikrishnan, Vinayakumar, R., Soman, K.P.A machine learning approach towards phishing email detection. In Proceedings of the anti-phishing pilot at ACM international workshop on security and privacy analytics (IWSPA AP), 2013, (2018) 455-468.

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

[36] K. Thakur, M.L. Ali, M.A. Obaidat, A.A. Kamruzzaman, systematic review on deep-learning-based phishing email detection. Electronics, 12(21), (2023) 4545. https://doi.org/10.3390/electronics12214545

[37] Y. Korkmaz, Fraud E-mail detection using intelligent algorithms: Comparison of traditional approaches with deep learning techniques. Information Processing & Management, 63(2), (2026) 104416. https://doi.org/10.1016/j.ipm.2025.104416

[38] N.G. Jameel, L.E. George, Detection of phishing emails using feed forward neural network. International Journal of Computer Applications, 77(7), (2013) 10-15. https://doi.org/10.5120/13405-1057

[39] P. C. R. Chinta, C. S. Moore, L. M. Karaka, M. Sakuru, V. Bodepudi, S. R. Maka. Building an Intelligent Phishing Email Detection System Using Machine Learning and Feature Engineering. European Journal of Applied Science, Engineering and Technology 3(2), (2025) 41-54. https://doi.org/10.59324/ejaset.2025.3(2).04

[40] C. Patra, D. Giri, S. Nandi, A.K. Das, M.J. Alenazi, Phishing email detection using vector similarity search leveraging transformer-based word embedding. Computers and Electrical Engineering, 124, (2025) 110403. https://doi.org/10.59324/ejaset.2025.3(2).04

[41] S. Rashed, C.A. Ozcan, Novel Dual-Layer Deep Learning Architecture for Phishing and Spam Email Detection. Electronics, 15(3), (2026) 630. https://doi.org/10.3390/electronics15030630

[42] A. Kumar, J.M. Chatterjee, V.G. Díaz, A novel hybrid approach of SVM combined with NLP and probabilistic neural network for email phishing. International Journal of Electrical and Computer Engineering, 10(1), (2020) 486.

[43] I. Altan, A. Bachir, Y. Parbhulkar, A.M. Rizvi, M. Farazi, (2025) Dual-Path Phishing Detection: Integrating Transformer-Based NLP with Structural URL Analysis. In 2025 IEEE/ACS 22nd International Conference on Computer Systems and Applications (AICCSA), IEEE, Qatar. https://doi.org/10.1109/aiccsa66935.2025.11315219

[44] S. Anirudh, P.R. Nishant, S. Baitha, K.D. Kumar. An ensemble classification model for phishing mail detection. Procedia Computer Science, 233, (2024) 970-978. https://doi.org/10.1016/j.procs.2024.03.286

[45] M. Adnan, M. O. Imam, M. F. Javed, I. Murtza. Improving spam email classification accuracy using ensemble techniques: a stacking approach. International Journal of Information Security 23(1) (2024) 505-517. https://doi.org/10.1007/s10207-023-00756-1

[46] A. Mathur, A.S. Dubey, P.S. Mahara, U.D. Gandhi. Unified approach integrating Machine Learning algorithms for detection of e-mail phishing attacks. In 2024 International Conference on Computing, Sciences and Communications (ICCSC), IEEE, India. https://doi.org/10.1109/ICCSC62048.2024.10830374

[47] R. Ahmed, E.P. Sumesh, An Intelligent Phishing Email Detection System Using Ensemble Methods and Explainable AI. Knowledge-Based Systems, (2026) 115388. https://doi.org/10.1016/j.knosys.2026.115388

[48] X. Yi, Li, L., Zhang, J., Jia, Z. Heterogeneous federated learning for imbalanced phishing email detection. Cybersecurity, 9(1), (2026) 10. https://doi.org/10.1186/s42400-025-00420-2

[49] B. Lim, R. Huerta, A. Sotelo, A. Quintela, P. Kumar. (2025) EXPLICATE: Enhancing Phishing Detection through Explainable AI and LLM-Powered Interpretability. arXiv preprint arXiv:2503.20796. https://doi.org/10.48550/arXiv.2503.20796

[50] A. Alzahrani, Explainable AI-based Framework for Efficient Detection of Spam from Text using an Enhanced Ensemble Technique. Engineering, Technology & Applied Science Research 14(4), (2024) 15596-15601. https://doi.org/10.48084/etasr.7901

Authors Contribution Statement

Mansi Harihar: Conceptualization, Methodology, Investigation, Data curation, Writing – Original Draft. Shilpa Deshpande: Investigation, Validation, Writing – Original Draft, Review and Editing. Mahendra Deore: Writing – Review and Editing. All authors read and approved the final version of the manuscript.

Funding

The authors declare that no funds, grants or any other support were received during the preparation of this manuscript.

Competing Interests

The authors declare that there are no conflicts of interest regarding the publication of this manuscript.

Data Availability

The data supporting the findings of this study can be obtained from the corresponding author upon reasonable request.

Has this article screened for similarity?

Yes

About the License