International Journal of Science and Research Archive

eISSN: 2582-8185

Cross Ref DOI: 10.30574/ijsra

Journal homepage: https://ijsra.net/

(Research Article)

Stacked ensemble improvement of phishing Email corpus detection based on frequency-based count vector embedding

Olayemi Olasehinde [1], Olayemi Olufunke Catherine [2] and Peter Adetola Adetunji [3, *]

[1] *Department of Computing and Engineering, University of Huddersfield, UK.*
[2] *Department of Computing and Games, Teesside University, Middlesborough, UK.*

International Journal of Science and Research Archive, 2024,13(02), 3774-3788

Publication history: Received on 18 November 2024; revised on 26 December 2024; accepted on 28 December 2024

Article DOI: https://doi.org/10.30574/ijsra.2024.13.2.1830

Abstract

Email users are at risk from phishing attacks, which utilize a combination of technological and social engineering techniques to obtain sensitive information from targets and cause significant financial loss. It is the fastest-rising online crime for stealing personal and financial data. In this work, natural language processing was applied to process an unstructured email corpus and convert it to a word vector matrix suitable to build machine learning models implemented using the Python programming language. The test corpus was evaluated using the four base models, and the results indicate that the random forest model had the highest accuracy (92.71%), closely followed by the logistic regression model (89.01%), the Naive Bayes recoded model (83.52%), and the KNN model (79.95%) with the lowest accuracy. A notable improvement in classification accuracy and a decrease in the false alarm rate observed by all base models were demonstrated by the stacked ensemble evaluation of the base model predictions, which yielded an accuracy of 97.14%. It recorded a classification improvement of 21.5%, 5.4%, 16.3%, and 9.1% over the KNN, RF, NB, and LR models, respectively, and a drop in false alarm rate by 79.0%, 36.0%, 76.4%, and 64.0% over the KNN, RF, NB, and LR models, respectively. The implementation of this approach on the mail server to filter incoming phished emails.

Keywords: Identity theft; Corpus Embedding; Phishing Detection; Meta-Learners

1. Introduction

Electronic mail (Email) is one of the most effective and easy methods of sending messages (Mails) over the internet. It is the most widely used of all the internet components, its advantages of being the cheapest and fastest methods of sending message and its ability to attach files (documents, video, audio files) with the transferred messages gives it an edge over other methods. Virtually all businesses, individuals, private and government sectors have adopted Email as a medium of corporate communication within and outside their organization, Email communication plays an important role in everybody's life. Nowadays email usage is experiencing tremendous growth compared to the olden days. According to Sara and Quoc (2019) Nearly 4.8 billion persons were using email as at 2017 and this number is have risen to 5.6 billion in 2021. But the main problem with email has been phishing mails which posed serious security challenges to both individuals and organizations that use email as platform for their communication.

The notion of phishing originated in the mid-1990s when hackers started using false identities to obtain America Online (AOL) accounts (Zahra et al 2022). Phishing emails are a cyberattack in which the attacker attempts to deceive people into disclosing sensitive information, including login passwords, bank account information, or personal information (Das et al., 2020). Phishing combines political science, technical systems, social psychology, and security. According to the APWG, phishing is "a criminal mechanism that uses technical deception as well as social engineering to steal consumers' financial account credentials and personal identity data" (APWG, 2020). A recent poll (Proofpoint, 2020) found that nearly 90% of businesses experienced targeted phishing attempts in 2019. This suggests that phishing

[*] Corresponding author: Adetola P. A

attacks are happening more frequently. They dealt with spear-phishing attacks in 88% of cases, voice phishing (also called Vishing) in 83%, social media attacks in 86%, SMS/text phishing (also called Smishing) in 84% of cases, and malicious USB drops in 81% of cases.

A recent report of the APWG Phishing Activity Trends (APWG, 2023), for the second quarter of the 2023, found that, 1,286,208 phishing attack cases were reported in Q2 of 2023. This is the third-highest quarterly total in the history of the organization. Phishing activity overall decreased despite this high number. There was a notable surge in the intensity of business email hack assaults, with an average 57% increase in wire transfer demands over the previous quarter. Compared to $187,053 in Q1, the average demand increased to $293,359. Approximately 23.5% of all phishing assaults were directed towards the banking industry, making it the most targeted industry overall. Online payment providers were also the target of 5.8% of assaults. More and more people are falling victim to voice-mail phishing, or vishing. Vishing is the practice of attackers tricking people into divulging private information by means of voice communications. This demonstrates how fraudsters are always changing their strategies to take advantage of weaknesses.

Phishing emails posed serious security risk that can cause individuals and businesses to experience several issues. The following are just a few horrifying effects of phished emails:

- Identity Theft: Phishing emails commonly attempt to trick users into divulging private information, including passwords, usernames, and bank account information. Attackers can use this data for illegal activity, such as identity theft, or unauthorized account access. Ramanatha and Wechesler (2012) define identity theft as impersonating a person’s identity to steal and use their personal information.
- Financial Loss: Victims are likely to suffer financial losses whenever phishing attacks occur. Attackers can utilize credentials they have stolen to access bank accounts, financial cards, or financial accounts. This could result in unauthorized activities and financial losses. The huge financial impact of phishing assaults was demonstrated in 2020 when the FBI revealed that losses resulting from these attacks totaled $4.2 billion.
- Data Breach: Having access to critical organization information or personal data through targeted phishing assaults can cause data breaches, which could have negative effects on one's reputation, legal action, fines, and regulatory penalties.
- Reputation Damage: Individual and organization's reputation may suffer if they fall prey to a phishing attempt. If confidential information is compromised, there could be long-term repercussions because customers, clients, or partners might lose trust.

Phishing is a cyber-attack that involves the use of psychological tricks by hackers to seduce victims into divulging critical information and valuable secrets that might be used to infiltrate their systems. The theft of private information from victims using technological and social engineering techniques is a crime (Manning & Aron 2015). The attack victims have suffered significant financial losses as a result of this major threat to personal information security. The fastest-rising online crime is identity theft and the theft of private financial information.

The downturn in the global economy, according to Lungu & Tabusca (2010), reflects the rise in phishing cyberattacks that have been occurring recently. Phishing is comparable to traditional fishing, however, instead of using a bait to capture a fish, the online strategy involves the phisher sending out multiple emails to as many recipients as possible, trying to persuade them to click on the embedded link and "catch" the bait (Al-Momani and Gupta 2013).

Another method employed is tricking the user by informing them that their user details has changed on their corporate account, and for them to log-in to review the changes. Once they click on an obfuscated link, they are re-directed to the malicious site, which gathers their details, and then redirects them to the corporate site. As far as the user is concerned, they had just put in the incorrect details but had just given away his confidential credentials.

Machine learning (ML) is a branch of artificial intelligence (AI) that focuses on developing algorithms and techniques that enable computers to learn from and make predictions or decisions based on data without being explicitly programmed. ML plays a crucial role in countering cyberattacks by leveraging algorithms to analyze data, detect patterns, and make predictions. It offers various applications in cybersecurity, including anomaly detection, intrusion detection systems (IDS), malware detection, phishing detection, user behavior analytics (UBA), threat intelligence, predictive analytics, adversarial machine learning, and automated response systems. It empowers organizations to enhance their security posture by detecting and mitigating threats more effectively, ultimately safeguarding their digital assets and infrastructure.

The Meta Model Tree (MMT) is a meta-learner category of ML. Meta-learners utilize predictions from multiple base learners to make final decisions or predictions, operating at a higher level of abstraction compared to traditional learning algorithms.

Frequency-Based Count Vector embedding (FBCVE) is a technique used in natural language processing (NLP) to convert textual data into numerical vectors. It provides a simple and effective way to process text data for machine learning applications. It involves representing each document in a corpus as a vector, where each element corresponds to the frequency of a specific word in the document. FBCVE creates a vocabulary of unique words from the corpus and counts the occurrences of each word in each document. These numerical representations enable the application of machine learning algorithms for tasks such as classification or clustering. Despite its full dependency on feature engineering and over-reliance on linguistic patterns, It has the potential to address changing cybersecurity risks and offers advantages like as efficiency, interpretability, versatility, and exhibiting continuing improvements in text representation approaches.

This research is set to answer the following questions

- How well do different ML algorithms perform in detecting phishing and legitimate emails, and what are their respective strengths and weaknesses?
- How effective are ensemble learning methods, such as the MMT algorithm, at improving the detection rate of phishing emails?
- What role does the FBCVE play in applying ML techniques to the real-time detection of phishing emails?

The answers to these study questions serve as a great reference for future research and offer important insights into how well machine learning algorithms, ensemble approaches, and natural language processing techniques work to identify phishing. The rest of the paper is organized as follows; Section 2 provides an overview of the theoretical framework, discussing the relevant literature and theoretical concepts that underpin the study. Section 3 outlines the research methodology employed in this study, detailing the data collection methods, sample selection criteria, and analytical techniques used for data analysis. In Section 4, the empirical findings are presented and discussed, including the key results and their implications. This section also includes any relevant tables, figures, or charts to support the findings. Section 5 offers a discussion of the results in the context of the existing literature, highlighting the contributions of this study and addressing any limitations or areas for future research. Finally, Section 6 concludes the paper by summarizing the main findings, reiterating their significance, and suggesting potential avenues for further exploration.

2. Literature Review

The detection and mitigation of phishing emails involves identifying and quarantine fraudulent messages that aim to trick recipients into disclosing private information, including bank account information or login passwords. (Al-Qahtani and Cresci, 2022). To keep ahead of phishing strategies as they evolve, research in this area is imperative. Though their efficacy is limited by certain issues, earlier attempts to employ different detection approaches have helped to produce strong detection systems. Heuristic-based filtering is less effective against complex attacks because it relies on established parameters (Bhadani, 2023). signature-based detection is limited to known patterns, and it cannot effectively combat evolving threats (sano et al., 2023). Behavioral analysis relies on user behavior, which cannot always reliably identify phishing attempts, machine learning, and natural language processing requires labeled data. The obfuscation of URLs makes it extremely difficult to detect phished websites based on their URLs. (Balogun et al., 2021)

AS effective, ensemble methods are, they require significant computational resources and may introduce complexity into detection systems. User education, though important, depends on user awareness, which can vary widely. It is essential to recognize these limitations and employ a combination of techniques for effective phishing detection. Effective phishing email detection often involves a combination of these approaches to provide comprehensive protection against phishing threats. (Olasehinde, 2019)

According to KeepnetLABS (2018) and Crane (2019), phishing takes use of both technology flaws and personal psychological and behavioral characteristics, putting everyone at risk. According to studies, when people believe an email is legitimate, they are more likely to click on the link (Furnell, 2007). Individuals and corporations must develop efficient procedures for detecting and reducing these dangers in light of the surge in phishing attempts in recent times. The phishing email ML model assigns emails to either legitimate or phishing categories using machine learning algorithms (Ham). The primary drivers of people's response to phishing assaults, according to a 2017 PhishMe analysis, are curiosity and urgency. That's why fighting these ever-present risks requires the development of effective detection techniques.

Basnet and Doleck (2015) conducted a comparison phished email detection check among seven ML techniques and showed that Random Forest performs the best while SVM performs the worst. Rawal et al., (2017) proposes a content based ML Phished email detection system, email content were preprocessed and converted into the form suitable for ML, five ML algorithms; SVM, Random Forest, Logistic, Naive Bayes and Voted Perceptron were used to trained extracted features and evaluated via ten folds cross validation techniques Maximum accuracy of 99. 87% was achieved by Random Forest model.

Ebubekir et al. (2017) developed a technique based on NLP for identifying phishing attacks that originate from URLs. The researchers propose a technique that leverages ML algorithms and NLP techniques to detect perceptual similarities to detect phishing assaults. With an astounding success rate of 97.2%, the experimental testing phase demonstrates the effectiveness of the RF algorithm in detecting and preventing phishing assaults. To be more precise, the research adds to the continuing efforts to develop trustworthy methods for phishing attack detection by utilizing NLP and ML technology.

Cohen et al. (2018) introduced a novel set of general descriptive features to improve the detection of phishing emails using ML techniques. These features are directly extracted from the email content, making them independent and suitable for real-time systems as they don't rely on internet access or additional tools. The features encompass all components of the email, including the header, body, and attachments.

The authors utilized a dataset consisting of 33,142 emails, comprising 38.73% malicious and 61.27% benign emails. They applied feature selection techniques to identify the 30 most significant features out of 100 initially extracted features using filter, wrapper, and embedded methods. Among nine commonly used machine learning classification algorithms, Random Forest (RF) achieved the highest detection accuracy of 92.9%, a true positive rate (TPR) of 94.7%, and a false positive rate (FPR) of 0.03%.

Text or document classification is a task in the field of NLP, where a given text is labeled and categorized into specified classes or categories. The objective is to automatically categorize textual data according to predetermined criteria, taking into consideration its content. It is supervised ML approach that learns from a given labeled text document and used the knowledge gained to classify or predict unseen text document to the right label. Text classification has its uses in several areas such as subject classification in news articles, sentiment analysis in social media, and spam detection in emails.

Olasehinde, (2019) carried out a researcher on Text Analysis and ML Approach to Phished Email Detection, three ML approaches were employed in the work to identify phished emails on a standard examined phished email and ham corpora: Naive Bayes, K-Nearest Neighbor, and Support Vector Machine. The work involved text mining of phished and ham emails. Based on the results, Naive Bayes was shown to have the highest classification accuracy (99.5%) compared to SVM (98.6%) and KNN (96.9%).

Zamir et al.,(2020) presented a phishing detection approach that enhances robustness and performance by utilizing several machine learning algorithms. The system extracts feature from websites, encompassing URL and HTML content, which are subsequently utilized for model training. A dataset containing both legitimate and phishing websites was used for the experiment. Findings showed that phishing websites could be detected with high accuracy with the random forest and support vector machine methods yielding the best outcomes. However, the approach lacks generalization to new, unseen datasets, and is limited to only website phishing detection.

Valecha et al. (2021) presented a technique that uses persuasion cues—more especially, gain and loss cues—to improve the detection of phishing emails. They developed three machine learning models: one that combined gain and loss persuasion cues, another that used relevant gain persuasion cues, and a third that used loss persuasion. These models were compared against a baseline model that did not consider persuasion cues. The Results showed that the models with relevant persuasion cues performed about 5–20% better in terms of F-score than the baseline model. This study demonstrates how well persuasion cues may be included in anti-phishing techniques to enhance phishing email detection and prevention.

A systematic literature review (SLR) on email phishing detection for individual and organizational users was provided by Muhammad et al. (2023). It highlighted the main difficulties in detecting email phishing, such as the ever-changing strategies employed by aggressors and the shortcomings of the available detection techniques. Additionally, it looked into how different phishing email difficulties affected users in private and in organizations. It provided insightful information about the unique difficulties faced by individual and corporate users in their attempts to fend off email phishing scams.

This research project aims to optimize phishing email detection by combining natural language processing (NLP), machine learning (ML), and ensemble learning optimization approaches. The NLP techniques preprocess unstructured email corpus, converting them into structured vectors suitable for ML algorithms. These ML algorithms build various models for detecting phished emails. Subsequently, a stacked ensemble combines the predictive power of each ML model's predictions. By leveraging NLP, ML, and ensemble learning, the study seeks to enhance the accuracy and effectiveness of phishing email detection systems, thereby improving organizations' security against email-based cyber threats.

3. Materials and Method

The System Architecture of the Stacked Ensemble Approach to Phishing Email Cyber Security Improved Detection with Multiple Model Tree Meta (MMT) Algorithm is shown in Fig. 1, it consists of three different stages; the text (corpus) pre-processing stage handles the cleaning, preparation and conversion of the unstructured email corpus to the structured form suitable for ML analysis. The second stage involves the building of the base predictive models, the three-base algorithm; KNN, RF, Naïve Bayes and LR were trained and evaluated with the training set using ten folds cross validation to generate the base predictions. The predictions of the four base models were trained with the MMT meta-algorithm to build the MMT stacked ensemble model. The last stage involves the evaluation of the built models; the pre-processed testing dataset was used to evaluate the four base models. Their predictions were used to evaluate the MMT stacked ensemble model via ten folds cross validation to produce the final predictions. Python language was used for the implementation of the models.

3.1. Dataset Description

The datasets used in this study are the Ham public mail corpus from Spam Assassin Project APWG (2013) and the Fraudulent e-mails, which are phished emails corpora [15]. The fraudulent e-mails contain criminally fraudulent information, typically with the intent of persuading the individual receiving them to give the sender a substantial amount of money. This dataset, which spans the years 1998 through 2007, consists of 4075 fraudulent emails. There are a total of 5500 corpora for training out of the 2,500 fake corpus and the 3000 Ham corpus that make up the training dataset. The other 2940 corpus from the Ham dataset and the remaining 1575 corpus from the fraudulent sample make up the testing dataset. The composition of the two email corpora employed in this study are depicted in Table 1. The study simply uses the email's main content, or text, leaving out the header details such as sender email, subject, CC, and BCC.

Table 1 Composition of Training and Testing Datasets

	Training	**Testing**	**Total Fraudulent and Ham Emails**
Fraudulent (Phished) Email	2500 (61.34%)	1575 (38.66%)	4075
Ham (Non-Phished) Email	3000 (50.51%)	2940 (49.49%)	5940
Total Training and Test Dataset	5500 (54.92%)	4515 (45.08%)	

3.2. Text (Email Corpus) Pre-processing

One of the most important steps in getting unstructured text data ready for ML tasks is text pre-processing. It entails preparing unformatted email corpus and transforming it into a suitable format for analysis. It is essential to guarantee the efficacy of subsequent ML models. An email corpus is a group of emails used as a dataset for analysis, research, and ML, among other uses. It is essentially an organized or unorganized collection of emails collected from various sources. It is used as the basis for a variety of activities, such as linguistic analysis and the creation and assessment of algorithms and models connected to emails.

Text preprocessing aims to produce a clear, consistent representation text data while preserving the crucial details needed for additional modeling or analysis. It involves the following stages.

3.3. Tokenization

Tokenization is the act of breaking down a text document into smaller pieces (tokens), like words or phrases, so that computers can analyze and process it more easily. In many NLP tasks, it is a fundamental step in making machines capable of understanding and interacting with natural language. Tokens are used to represent a range of language elements, like phrases, sentences, and words. The type of tokenization approach used depends on the task at hand and the level of granularity. In this work, words are the appropriate level of token granularity.

In the statement, "Click the link below to receive a $10,000 voucher," The terms "click," "the," "link," "below," "to," "receive," "a," "voucher," "of," and "$10,000." become distinct words after they have been tokenized. Tokens are components that are used in later language-related analysis. Words that are not separated by spaces and compound words require a higher level of tokenization.

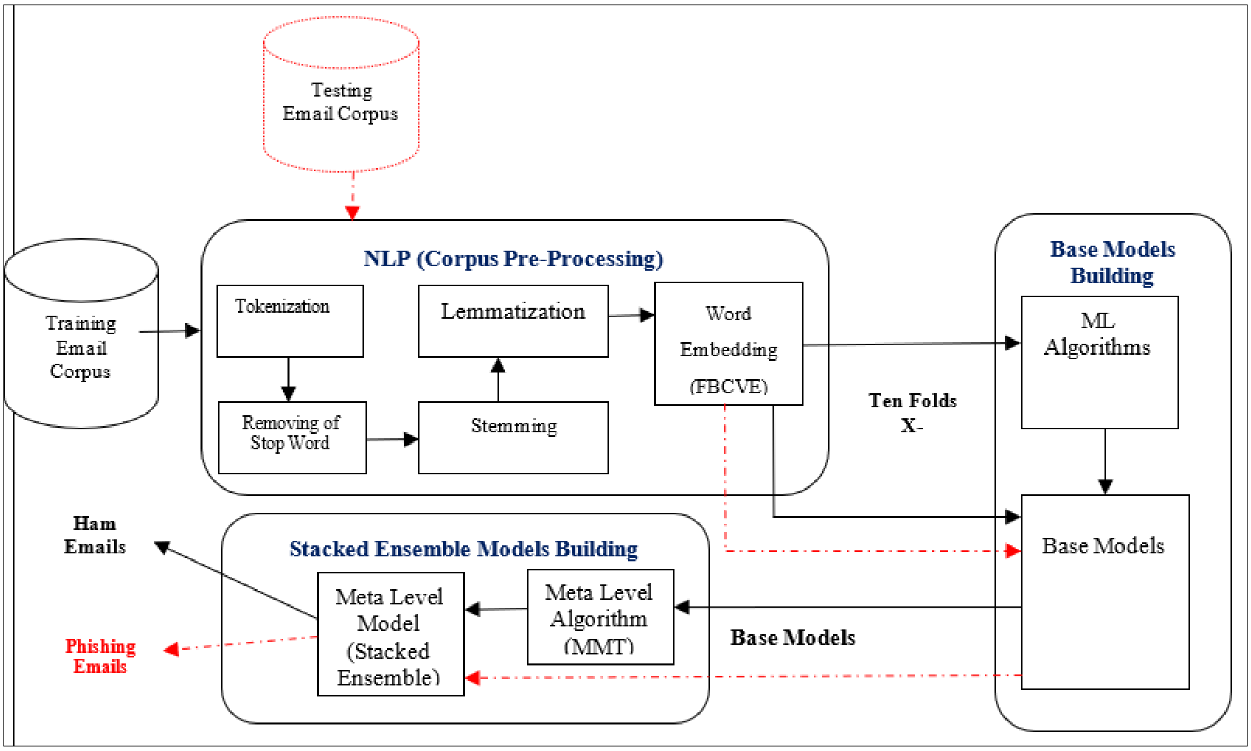


Figure 1 Architecture of Ensemble Approach to Phishing Email Cyber Security Improved Detection

3.4. Removal of Stop Words

Stop words are frequently used words regarded as having minimal significance in text analysis. These words are commonly seen in many documents, although they have little meaning. The idea behind removing stop words is to focus on the more meaningful words that contribute to the understanding of the content and the identification of patterns. Removing stop words will make the text data representation informative

The decision of removing stop words may vary depending on the particular task and the data's properties. Stop words may not always need to be removed because they may be important in some situations (such as when performing specific information retrieval tasks). The purposes of the study or the requirements of the ML model being employed frequently determine if stop words are to be removed or not.

3.5. Words Embedding

This is a NLP technique used to represent a continuous vector space words (tokens) as vectors. In order to maintain the semantic links between words, it entails mapping words from a high-dimensional discrete space (vocabulary) into a lower-dimensional continuous space. Word vectors or word embeddings are continuous vector representations that are useful for a variety of NLP tasks because they represent semantic meanings and relationships. Embedding converts tokens into numbers; it maps a token (word) using a dictionary to a vector, in this work, Frequency Based Count Vector embedding (FBCVE) was employed for the implementation of the email corpus. FBCVE is a technique used in NLP and

ML to represent text data as numerical vectors. The purpose of this technique is to generate a feature vector for every document by measuring the frequency of words in the content.

Consider a Corpus C_i of n Tokens [T_1,T_2.....T_n], the N tokens (unique combination of all n tokens in each corpus C_i) will form our dictionary and the size of the Count Vector matrix M will be given by C X N. Each row in the matrix M contains the frequency of tokens in Corpus C_i.

For example, let corpus C_1 be: "access will be denied", corpus C_2 be "account will be locked.". Text pre-processing will turn the corpora to : corpus C_1 ["access" "will" "be" "denied"], corpus C_2 ["account" "will" "be" "locked"] The dictionary will create a list of unique tokens (words) from the two corpus of the form dictionary =["access" "account" "will" "be" "locked" "denied"] Table 2 shows the Vector count matrix representation of Corpus C_1 and C_2 python function; make dictionary (preprocessed training dataset) created 2425 unique tokens from the 5500 training corpus, vector count matrix has 5500 rows which denotes the 5500 dataset files and 2426 columns denote 2425 most frequent words in the dictionary and the corpus label class as illustrated Table 3.

Table 2 Word Embedded of Corpus C_1 and C_2

	Account	**Access**	**will**	**be**	**denied**	**Locked**
Corpus C_1	0	1	1	1	1	0
Corpus C_2	1	0	1	1	0	1

Table 3 Vector Matrix of the preprocessed Training Set Corpus

	Token1	**Token2**	**.....**	**Token2425**	**Class label**
Corpus C_1					1- Phished
.					0 – Ham
Corpus C_{5500}					1- Phished

3.6. Stacked Ensemble Framework

Stacked Ensemble is a supervised two levels learning approach used to improve ML predictions by combining predictions of several ML models. This approach allows the prediction of an enhanced prediction accuracy model compared to the individual base models which their predictions were combined. Stacking consists of two levels, which are base learner as level 0 and stacking model learner as level 1. Base learners (level 0) were built from the training of diver's ML algorithms with the training dataset, the predictions of the evaluation of the different base learners are used to train the meta learner to build the meta model, prediction of the meta model form the final prediction of the stacked framework.

3.7. Base learners

Each base learner is used to independently trained the dataset and their predictions are used to train Meta leaner to make a final prediction. Four ML Algorithms; K Nearest Neighbor, Random Forest, Naive Bayes and Decision Tree were adapted to build the base models. K-Nearest Neighbor (KNN). KNN refers to distance based classification model capable of handling both binary and multiclass labels classification, it is an instance based learner that does less work during the training and more work during classification and prediction, model evaluation with KNN is very computational and expensive, each instance to be classified are compared against all instances in the training set in terms of their Euclidean distance, label of the closest neighbor are returned as the label of instance being classified. Random Forest (RF). A random forest is an ensemble decision tree classification *algorithm*. Each decision tree is independent of the other in their individual predictions, RF uses bagging and feature randomness when building each individual *tree* to try to create an uncorrelated *forest* of trees whose prediction by committee is more accurate than that of any individual *tree*. RF performs well with large datasets and can handle both binary and multi-class label problems. Naïve Bayes (NB). NB is a probabilistic predictive algorithm, based on independence and probability (Bayes theorem). Given a class label, NB classifier adopts the idea that the existence of a certain feature of an object is unrelated to the existence of any other feature. It treats all features as independent of one another, it is scalable and does not require huge instances of dataset to build an efficient model. Logistic Regression (LR). LR is a ML algorithm; it is a simple algorithm that models the probability of the class label against each of the independent features' attribute. Like NB, it adopts the idea that the existence of a certain feature of an object is unrelated to the existence of any other feature. it treats all features as independent of one another, size of training set affects it performance and is a binary classifier

3.8. Meta Learner

The Meta Model Tree (MMT) is a ML algorithm that is particularly useful for classification problems. It is a type of meta-algorithm that combines base predictions with linear regression functions at the leaves. MMT algorithm combines the strengths of regression and classification to provide a potent ensemble model that can handle challenges involving both regression and / or classification. When these two methods are combined, flexibility and accuracy are increased, particularly in situations when traditional base models s would not perform well. Meta-learners stacking combines of predictions from the base learners as features for a higher-level model. (Olasehinde *et al.,* 2018)

During training, MMT constructs a tree structure atop the base learners, where each node represents a base learner, and the leaves represent their predictions. The algorithm learns to assign weights to the predictions of different base learners to optimize overall performance. MMT is beneficial when dealing with heterogeneous data or when individual base learners perform differently across various parts of the feature space. By amalgamating predictions from multiple base learners, the meta-learner achieves superior generalization performance compared to any single base learner.

The algorithm of MMT is depicted in fig 2, the predictions of each of base leaner's form the attributes of the level two dataset while it label is the original label from the base training (level one) dataset. The M5' algorithm was then induced on the derived dataset for phished and not phished dataset to build phished and not phished regression function f. These two functions are then applied on the prediction of a new test instance, the function with the highest is refunded as the classified label.

Fig. 2 depicted the virtualization of figure 2. The Meta-Model Trees (MMT) algorithm utilizes a two-step process for classification, particularly in phishing detection: Base learners are trained on the original dataset. Predictions from these base learner's form attributes of a new dataset, and the original labels are retained. M5's algorithm was induced to build regression functions for phishing and non-phishing instances using the derived dataset. The base learner predictions of a new instance under examination are fed into regression functions, and the function with the highest value is selected for classification.

4. Result

The four models prediction are used to train the Meta Model tree (MMT) algorithm to build the MMT stacked ensemble model, Table 4 shows confusion matrix of the evaluation of all the models on the test dataset and their performance, , Random forest model recorded the highest correct classification of 4186 corpus and 329 incorrectly classified corpus with accuracy of 92.71%, closely followed by Logistic Regression model with 4019 correctly classified corpus and 496 incorrectly classified corpus with accuracy of 89.01%, Naive Bayes recoded 3771 and 744 correctly and incorrectly classified corpus respectively with accuracy of 83.52%, KNN recorded the least performance of 3610 and 905 for correctly and incorrectly classified corpus respectively with accuracy of 79.95%, the highest accuracy of 92.71% recorded by RF can be linked to the fact that, RF is an Ensemble several decision trees. The stacked Ensemble model shows an improvement over the RF with 4386 and 129 of correctly and incorrectly classification respectively with accuracy of 97.14%. Fig.4 shows the accuracy of evaluation of each classification model on the test dataset.

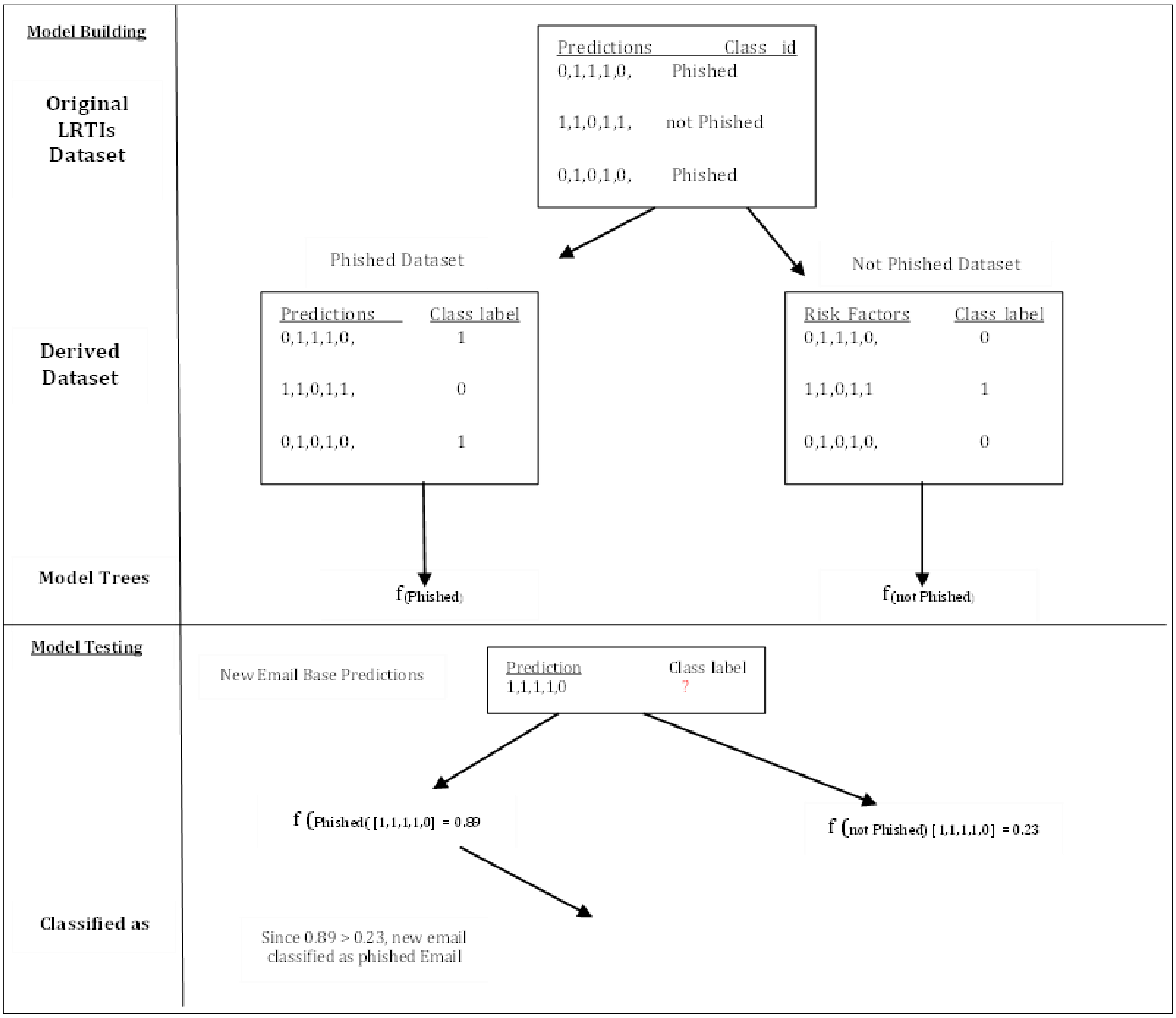
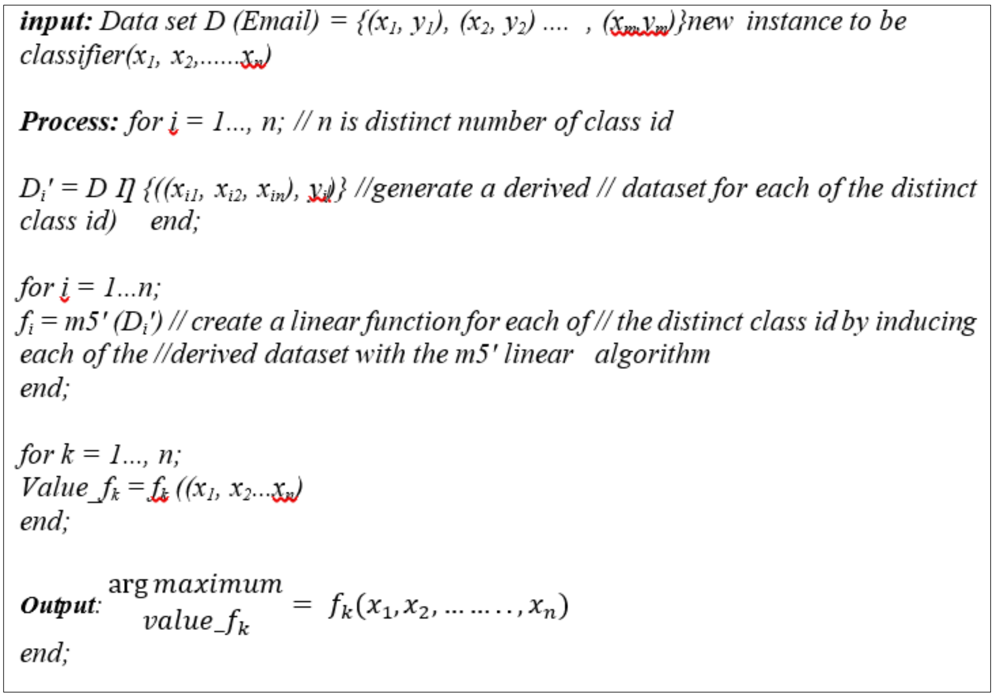


Figure 2 The operations of Meta Model Trees Algorithm



```
input: Data set D (Email) = {(x_1, y_1), (x_2, y_2) ....  , (x_m y_m)}new  instance to be classifier(x_1, x_2,......x_n)

Process: for i = 1..., n; // n is distinct number of class id

D_i' = D Ŋ {((x_i1, x_i2, x_in), y_i)} //generate a derived // dataset for each of the distinct class id)    end;

for i = 1...n;
f_i = m5' (D_i') // create a linear function for each of // the distinct class id by inducing each of the //derived dataset with the m5' linear   algorithm
end;

for k = 1..., n;
Value_f_k = f_k ((x_1, x_2...x_n)
end;

Output: $\underset{value\_f_k}{\arg maximum} = f_k(x_1, x_2, \ldots\ldots.., x_n)$
end;
```

Figure 3 Multiple Model Trees algorithm

Table 4 Classification Models Confusion Matrix and Predictive Performance

	Base Models Predictions Performances				**Meta Model Predictions Performance**			
Classification Models	Confusion Matrix		Accuracy %	False Rate %	Confusion Matrix		Accuracy %	False Rate %
K- Nearest Neighbor	TP = 1315	FN = 645	79.95	10.18				
	FP = 260	TN = 2295						
Random Forest	TP = 1482	FN = 236	92.71	3.32				
	FP = 93	TN = 2704						
Naive Bayes	TP = 1332	FN = 501	83.52	9.06				
	FP = 243	TN = 2439						
Logistic Regression	TP = 1412	FN = 333	89.01	5.88				
	FP = 163	TN = 2607						
Meta Model Trees Stacked Ensemble					TP = 1512	FN = 66	97.14	2.14
					FP = 63	TN = 2874		

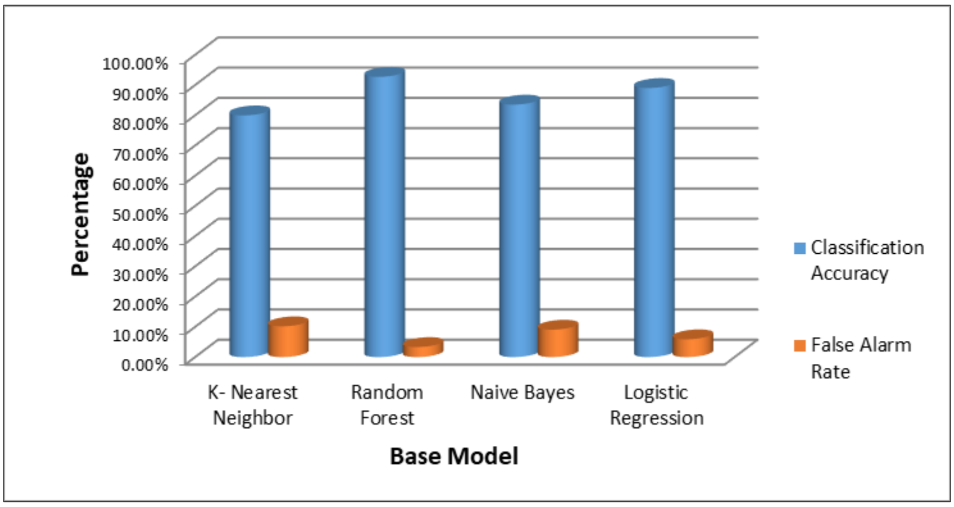


Figure 4 Classification Performances of Base and Stacked Ensemble Models

Table 5 and Table 6 shows the performance improvement of the MMT stacked ensemble model over the base models, from Table 5, MMT stacked ensemble model recorded classification improvement of 21.50%, 5.36%, 16.31% and 9.13% over KNN, RF, NB and LR models respectively as shown in Fig.5, from Table 6, MMT stacked ensemble model recorded false alarm rate reduction of 78.98%, 35.54%, 76.38% and 63.61% over KNN, RF, NB and LR models respectively as shown in Fig.6

Table 5 Classification Accuracy Performance Improvement of MMT Stacked Ensemble Prediction over the Base Models Predictions

Base Models	**Base Model Accuracies (%)**	**MMT Stacked Ensemble Accuracy (%)**	**MMT Stacked Ensemble Accuracy Improvement (%)**
K- Nearest Neighbor	79.95	97.14	21.50%
Random Forest	92.71		5.36%
Naive Bayes	83.52		16.31%
Logistic Regression	89.01		9.13%

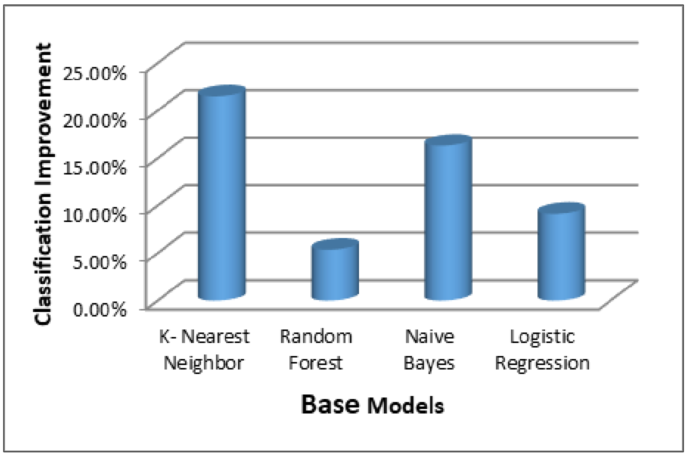


Figure 5 Classification Accuracy improvement of Stacked Ensemble Model Over Base Models

Table 6 False Alarm Rate Improvement of MMT Stacked Ensemble Model Prediction over the Base Models Predictions

Base Models	**Base Model False Alarm Rate (%)**	**MMT Stacked False Alarm Rate (%)**	**MMT Stacked Ensemble False Alarm Rate Improvement (%))**
K- Nearest Neighbour	10.18	2.14	78.98%
Random Forest	3.32		35.54%
Naive Bayes	9.06		76.38%
Logistic Regression	5.88		63.61%

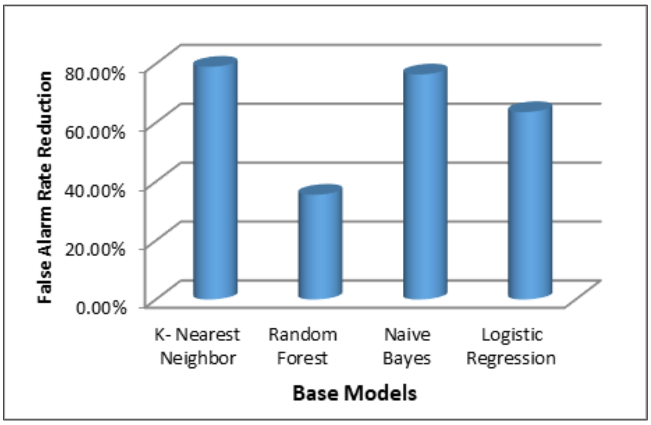


Figure 6 False Alarm Rate Reduction of Stacked Ensemble Model over Base Models

5. Discussion

This study provides a detailed overview regarding the use of frequency-based count vector embedding and stacked ensemble methods, particularly the Meta Model Tree (MMT) algorithm, in text classification tasks. and ML.

The study employs frequency-based count vector embedding to transform textual data into numerical representations, enhancing predictive capabilities by capturing semantic information crucial for classification. The integration of count vector embedding into ensemble learning highlights the synergy between feature engineering techniques and ensemble methods. While embedding captures semantic richness, ensemble methods leverage multiple base models to navigate complex decision boundaries, emphasizing the importance of a holistic approach in model development.

Results from the study illustrate the superiority of stacked ensemble methods, especially with MMT, in achieving higher classification accuracy compared to individual models across various algorithms. The MMT algorithm contributes significantly to the improved performance of the stacked ensemble model, achieving higher accuracy and lower false alarm rates. While ensemble methods offer superior predictive performance, they often sacrifice interpretability. However, the MMT algorithm provides a degree of interpretability by revealing the decision-making process at the meta-model level.

The successful application of count vector embedding has significant implications for real-world applications such as sentiment analysis and document categorization. The study's findings highlight the efficacy of stacked ensemble methods, especially when augmented with sophisticated meta-modeling techniques like MMTs, in improving classification accuracy and robustness. These insights contribute to advancing machine learning by providing a deeper understanding of ensemble methods for complex classification tasks and paving the way for further research and innovation in the field.

Overall, the study underscores the potency of stacked ensemble methods and feature engineering techniques in text classification tasks, offering valuable insights for both theoretical understanding and practical application in various domains.

6. Conclusion

This research employed a robust stacked ensemble approach to enhance the detection of phishing emails, focusing on the integration of multiple base models and a sophisticated meta-learner, the Meta Model Tree (MMT). The systematic methodology outlined in the Materials and Methods section ensured a comprehensive approach, from dataset preparation through text pre-processing to the implementation of the ensemble framework.

The dataset, composed of fraudulent (phished) emails and legitimate (ham) public mail, provided a diverse and representative set for training and testing. The careful preprocessing of an unstructured email corpus, including tokenization, noise elimination, and lexicon normalization, played a pivotal role in transforming raw data into a format suitable for ML analysis. The adoption of techniques such as stemming and lemmatization aimed at reducing lexical variations further contributed to the effectiveness of the models.

A careful selection of algorithms with different strengths is reflected in the selection of four different base models: K Nearest Neighbor, Random Forest, Naïve Bayes, and Logistic Regression. These base models might be combined with the later Meta Model Tree meta-learner implementation to create a stacked ensemble model that could outperform individual models.

Upon evaluating the models on the testing dataset, the results showed some noteworthy performances, with Random Forest being the basic model with the highest accuracy. But the MMT algorithm-driven stacked ensemble model outperformed all standalone models, with a 97.14% accuracy rate. This represents a significant advancement and shows how well the ensemble approach utilizes the capabilities of several models.

Furthermore, the comparison of false alarm rates highlighted the Stacked Ensemble Model's ability to significantly reduce misclassifications, showcasing its robustness in distinguishing between phishing and legitimate emails. The improvements over base models in terms of accuracy and false alarm rates are quantified, emphasizing the practical significance of the ensemble strategy.

In summary, this research not only presents a systematic and detailed methodology for phishing email detection but also underscores the substantial benefits of leveraging a stacked ensemble approach. The results demonstrate the synergistic power of combining diverse models, ultimately enhancing the accuracy and reliability of phishing email detection systems. This research contributes to the ongoing efforts to strengthen cyber-security measures and showcases the potential of ensemble techniques for addressing complex classification challenges

Compliance with ethical standards

Disclosure of conflict of interest

No conflict of interest to be disclosed.

References

[1] Al-Momani, A., Gupta, B. B., Atawneh, S., Meulenberg, A., and Al-Momani, E. 2013. A survey of phishing email filtering techniques. Communications Surveys & Tutorials, IEEE, 15 (4), 2070-2090.

[2] Al-Qahtani AF, Cresci S. The COVID-19 scamdemic: A survey of phishing attacks and their countermeasures during COVID-19. IET Inf Secur. 2022 Sep;16(5):324-345. doi: 10.1049/ise2.12073. Epub 2022 Jul 4. PMID: 35942004; PMCID: PMC9349804.

[3] Anti-Phishing Working Group, 2006. Phishing Activity Trends Report, Retrieved 11th, November 2023 from http://www.antiphishing.org/reports/ apwg_report_mar_06.pdf

[4] APWG (2020). APWG phishing attack trends reports. 2020 anti-phishing work. Group, Inc Available at: https://apwg.org/trendsreports/ (Accessed September 20, 2023).

[5] APWG (2023). APWG phishing attack trends reports. 2023 anti-phishing work. Group, Inc Available at: apwg_trends_report_q2_2023PDF (docs.apwg.org) (Accessed January 23m 2024)

[6] Balogun AO, Adewole KS, Raheem MO, Akande ON, Usman-Hamza FE, Mabayoje MA, Akintola AG, Asaju-Gbolagade AW, Jimoh MK, Jimoh RG, Adeyemo VE. Improving the phishing website detection using empirical

analysis of Function Tree and its variants. Heliyon. 2021 Jun 29;7(7):e07437. doi: 10.1016/j.heliyon.2021.e07437. PMID: 34278030; PMCID: PMC8264617.

[7] Basnet R. B., Doleck T. 2015. Towards developing a tool to detect phishing urls: A ML approach," in Computational Intelligence & Communication Technology (CICT), 2015 IEEE International Conference on, pp. 220–223, IEEE.

[8] Bhadani, D. A. 2023). Heuristic-based Phishing Site Detection (Doctoral dissertation, California State University, Northridge).

[9] Cohen, A., Nissim, N., & Elovici, Y. (2018). Novel set of general descriptive features for enhanced detection of malicious emails using machine learning methods. Expert Syst. Appl., 110, 143-169.

[10] Crane, C. (2019). The dirty dozen: the 12 most costly phishing attack examples. Available at: https://www.thesslstore.com/blog/the-dirty-dozen-the-12-most-costly-phishing-attack- (accessed August 2, 2023).

[11] Das, A., Baki, S., El Aassal, A., Verma, R., & Dunbar, A. (2020). SoK: A Comprehensive Reexamination of Phishing Research From the Security Perspective. IEEE Communications Surveys & Tutorials, 22(1), 671–708. https://doi.org/10.1109/COMST.2019.2957750

[12] Ebubekir, B. Diri, B. Sahingoz O.K.. NLP based phishing attack detection from URLs, in International Conference on Intelligent Systems Design and Applications, Springer, Cham, 608-618 (2017)

[13] Furnell, S. (2007). An assessment of website password practices. Comput. Secur. 26, 445–451. doi:10.1016/j.cose.2007.09.001

[14] Jain, A., Richariya, V. 2011. Implementing a Web Browser with Phishing Detection Techniques. arXiv preprint arXiv:1110.0360.

[15] Keepnet LABS (2018). Statistical analysis of 126,000 phishing simulations carried out in 128 companies around the world. USA, France. Available at: www.keepnetlabs.com.

[16] Lungu, I., & Tabusca, A. 2010. Optimizing anti-phishing solutions based on user awareness, education and the use of the latest web security solutions. Informatica Economica, 14(2), 27

[17] Maimon O. and Rokach L. 2004. Ensemble of Decision Trees for Mining Manufacturing Data Sets, Machine Engineering, 4:(1-2) 56-76.

[18] Manning, R., and Aaron, G. 2015. Phishing Activity Trends Report. Anti-Phishing Work Group, Tech. Rep. 1st -3rd Quarter

[19] Muhammad N., Syeda W. Z., Muhammad N. A. Ali A., Saman R., Waqas A. 2023. Phishing Attack, Its Detections and Prevention Techniques. International Journal of Wireless Security and Networks. 1(2): 13–25

[20] Olasehinde O. O., Alese B.K., Adetunmbi A.. O. 2018. A ML Approach for Information System Security. IJCSIS. 16(12)

[21] Olasehinde O. O. 2019. Text Analysis and ML Approach to Phished Email Detection, International Journal of Computer Application, 0975-8887, 182(36)

[22] Omar, A. R., Taie, S., & Shaheen, M. E. (2023). From Phishing Behavior Analysis and Feature Selection to Enhance Prediction Rate in Phishing Detection. International Journal of Advanced Computer Science and Applications, 14(5).

[23] Proofpoint (2020). 2020 state of the phish. Available at: https://www.proofpoint.com/sites/default/files/gtd-pfpt-us-tr-state-of-the-phish-2020.pdf. (Accessed 19th January 2024.)

[24] PhishMe (2016). Q1 2016 malware review. Available at: WWW.PHISHME.COM.

[25] Ramanathan, V. Wechsler, H. 2012. phishGILLNET - Phishing Detection Methodology using Probabilistic Latent Semantic analysis, AdaBoost, and cotraining. EURASIP Journal on Information Security, 1-22

[26] Rawal, S., Rawal, B., Shaheen, A., Malik, S. 2017. Phishing Detection in E-mails using ML. International Journal of Applied Information Systems. 12. 21-24. 10.5120/ijais2017451713.

[27] Sara R., Quoc H. 2015. Email statistics report, 2011-2015. Retrieved May, 222nd October, 2023 http://fliphtml5.com/uteh/jwtn.

[28] Soni, J., Sirigineedi, S., Vutukuru, K. S., Sirigineedi, S. C., Prabakar, N., & Upadhyay, H. 2023. Learning-Based Model for Phishing Attack Detection. In Artificial Intelligence in Cyber Security: Theories and Applications (pp. 113-124). Cham: Springer International Publishing.

[29] Valecha, R., Mandaokar, P., & Rao, H. R. 2021. Phishing email detection using persuasion cues. IEEE transactions on Dependable and secure computing, 19(2), 747-756.

[30] Zahra SW, Arshad A, Nadeem M, Riaz S, Dutta AK, Alzaid Z, Almotairi S, et al. 2022. Development of Security Rules and Mechanisms to Protect Data from Assaults. Appl Sci. 2022; 12(24): 12578.

[31] Zamir, A., Khan, H. U., Iqbal, T., Yousaf, N., Aslam, F., Anjum, A., & Hamdani, M. (2020). Phishing web site detection using diverse machine learning algorithms. The Electronic Library, 38(1), 65-80.