International Journal of Computer Applications (0975 - 8887)
Volume 187 - No.24, July 2025

Machine Learning-based Detection of Spear Phishing Emails

Md. Siam Ansary
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology
141 & 142, Love Road, Tejgaon Industrial Area
Dhaka-1208, Bangladesh

ABSTRACT

Phishing attacks are a serious risk to people, companies, and even whole systems. Therefore, it is critical to identify phishing emails. By identifying such emails, people may be shielded from identity theft and financial crime by preventing unwanted access to private information. In addition, it can aid in preventing monetary losses, misappropriation of private and business identities, preservation of credibility and trust, and compromise of networks and systems. Phishing threat detection and mitigation add to general cyber security awareness. It is essential to maintaining the privacy of sensitive information. In this study, we have observed how efficiently machine learning (ML) models can identify phishing emails. From evaluation of the models, it has been ascertained that the ML models can classify the spam emails very proficiently.

General Terms

Computer Science, Artificial Intelligence, Cyber Security

Keywords

Phising, Spear Phising, Machine Learning, Security

1. INTRODUCTION

A specific person is the target of spear phishing emails, a sort of targeted cyberattack in which the sender tailors their misleading emails with the intention of deceiving the receiver into disclosing sensitive information, including bank account information or login passwords. Personalised and more advanced than standard phishing emails are these attacks.

Spear phishing emails are customised for particular people or companies. To make the emails seem more genuine, the attackers acquire details about the victim, including their relationships, activities, and place of employment. The chances of success are increased by this focused strategy. Attackers frequently assume the identity of someone the target is familiar with and trusts, like a manager, coworker, or reputable organisation. The likelihood that the recipient will fall for the fraud increases when the attackers generate a false sense of trust by using well-known names, titles, or trademarks. Emails with spear phishing content usually pertain to current events, initiatives, or particular problems, and are therefore relevant to the intended recipient. This strengthens the email's persuasiveness and raises the possibility that the receiver will perform the intended action, such opening a dangerous attachment or clicking on a malicious link. Phishing emails frequently include dangerous attachments or viruses. Malware may install on the victim's device as a result of opening an attachment or visiting a link. This malware might be created to spy on activities, steal confidential data, or help launch more assaults on the targeted system. When spear phishing attacks are successful, private information may be accessed or stolen, resulting in data breaches. Serious ramifications could result from this, such as monetary losses, harm to one's reputation, and legal issues. An attacker may utilise a compromised account that a victim of spear phishing has obtained to launch additional attacks from within the organisation. This sideways motion may result in a more significant security vulnerability.

Machine learning (ML) can be a powerful tool for spear phishing detection. Through the use of sophisticated algorithms to examine patterns, spot abnormalities, and identify traits connected to phishing attempts, ML can be a vital tool.

Email content analysis is possible with NLP approaches. Phishing signals include strange wording, requests for sensitive information, and inconsistent terminology use, all of which can be recognised by machine learning models.

Spear phishing detection accuracy can be improved overall by using ensemble methods, such as merging the predictions of several machine learning models. Integrating the output of several algorithms that each specialise in identifying distinct phishing attack elements may be necessary to accomplish this.

2. LITERATURE REVIEW

The application of machine learning to the identification of phishing emails has been the subject of scientific investigations and studies. In this regard, machine learning and cybersecurity are fields that are developing quickly.

According to the research of Ding et al.[1], spear phishing emails are more sophisticated, targeted, and dangerous than phishing emails because they are directed towards a specific person or organisation. 417 spear phishing emails and 13916 non-spear phishing emails, comprising initial benign and phishing emails, made up the research dataset. Security professionals manually found and validated these spear phishing emails. The researchers employed for machine learning methods, Random Forest, Decision

Tree, Logistic Regression, and Support Vector Machine, and used modified synthetic minority oversampling technique by K-means cluster to lessen the influence of unbalanced data. Maximum recall of 95.56%, precision of 98.85%, and F1-score of 97.16% were attained in the study.

By using features selection to identify the strongly correlated features with the class label, Odeh et al [2]. presented a detection model. To identify the features that were significantly associated, independent significance features were used in the features selection step. The suggested model then employed a technique called adaptive boosting, which makes use of several classifiers, to raise the model's accuracy. The methodology yielded a remarkably elevated forecast precision of almost 99%.

In one study [3], a multilayered stacked ensemble learning technique was applied to detect phishing attacks. This technique comprised estimators at various layers, with the predictions of the current layer's estimators being provided as input to the subsequent layer. The experimental results showed that, with an accuracy ranging from 96.79% to 98.90%, the suggested model performed significantly well when tested on various datasets. The multilayer ensemble learning paradigm consisted of three layers. Multi Layer Perceptron (MLP), K Nearest Neighbor (KNN), Random Forest (RF), Logistic Regression (LR), and eXtreme Gradient Boosting (XGB) were used in the first layer. XGB was present in the last layer, while RF, MLP, and XGB were used in the second layer.

A multilayer stacked ensemble learning model with boosting as its foundation was presented in a different study [4]. This model selected the pertinent features for the classification using a hybrid feature selection strategy. The eXtreme Gradient Goosting (XGBoost), Light Gradient Boosting Machine (LGBM), CatBoosting (CatB), and AdaBoost (AdaB) models were the four estimators in the first layer. XGB is the meta-learner in the final layer, whereas CatB, AdaB, and XGB are the three estimators in the second layer. According to the results of the experimental investigation, the accuracy of the suggested model ranged from 96.16 to 98.95% across several datasets when feature selection was not used, and from 96.18% to 98.80% when feature selection was used. Three boosting models—XGB, CatB, and LGBM—were used in this study's suggested feature selection method. These models were applied to the datasets under evaluation in order to extract pertinent attributes. Next, utilising XGB, CatB, and LGBM, the feature importance for each feature was calculated independently. Random Under Sampling (RUS) and Random Over Sampling (ROS) were employed to address imbalanced datasets.

According to Raj and Jothi [5], the likelihood of phishing attempts can be greatly decreased by identifying rogue websites. They explored with various machine learning models in their study to test phishing detection techniques. In order to identify fraudulent websites, eight machine learning classification approaches that are currently in use were tested: extreme gradient boosting, random forest, adaptive boosting, decision trees, K-nearest neighbours, support vector machines, logistic regression, and Naive Bayes. Based on the data, it was found that XGboost had the highest accuracy (96.71%), followed by AdaBoost and random forest. The researchers also experimented with various combinations of the top three classifiers and noticed that XGboost-Random Forest hybrid methods generated the best results. With 97.07% accuracy, the hybrid model identified the websites as phishing or authentic. Thirty percent of the dataset was used for testing and seventy percent was used for training the models.

Samad and Gani[6] worked on using the multi-layer method to lessen the impact of spear phishing. They used SVM classifier and Random forest classifier to do sentiment analysis on emails, including both the email content and attachments, in order to determine whether or not they are spam. The former demonstrated 96 percent accuracy, while the latter offered 97.66 percent accuracy. Using machine learning methods like SVM and Random-forest, the researchers conducted a sentiment analysis of the email content and attachment in the first tier. In order to verify the legitimacy of the attachments, the sender and recipient sides computed hash values in the second tier; in the case of uncorrupted files, these hash values stayed constant. The hash values for corrupted files also changed at the same time. Using topic modelling and LDA, it was possible to identify the research dataset's main topics—topics that hackers exploit for spear phishing.

Atmojo et al. [7] developed a cosine-similarity-based model for spear phishing activity detection. The goal of the study was to find host behaviour that matched the spear phishing activity knowledge database in terms of activity pattern. The goal of the study was to create a new detection model based on network traffic data analysis, which captures the email sending communication process via a computer network, in order to identify spear phishing operations. The suggested model has a 90.45% detection accuracy when it came to spear phishing activities. The network traffic used in this investigation was actual activity data that was captured with the Wireshark application and a network activity recording programme.

Cazares et al. [8] investigated the mental model people use to determine whether or not an email is real in an effort to increase efficiency while identifying phishing assaults using natural language processing. In particular, the method relied on mouse movements and feedback vectorization, which were acquired during participant interaction in a phishing detection test. The findings made it possible to determine that people's decisions were based on URL analysis in their mental models of phishing and legitimation. The phishing model could present a more diverse and expansive set of features for each indication, potentially leading to new problems and opportunities for future research and development in this area. For the investigation, a six-layer Long Short-Term Memory (LSTM) model has been employed. With a square error of 0.05 and an accuracy of 99.38%, the experiment was successful.

Dewis and Viana [9] developed a system that combined natural language processing with a hybrid machine learning technique to identify spam and phishing emails. The model was put through an experiment to make sure it was effective, and it was able to obtain an average accuracy of 99% with the LSTM model for text-based datasets. Additionally, for datasets with a numerical basis, the method demonstrated an average accuracy of 94% when paired with the MLP model. 30% of each dataset was used for testing and 70% was used for training. According to the research, processing data using only tokenization and TF-IDF was the optimal NLP strategy.

3. METHODOLOGY

3.1 Dataset

We have done an experiment with different Machine Learning models to observe how efficiently the models can identify a phising email. We have collected three datasets from the Kaggle [10] platform. The Datasets are

—Spam Email

—Spam Classification for Basic NLP

—Email Spam Dataset

The first dataset has 4825 ham emails and 747 spam emails; the second dataset has 3900 ham emails whereas 1896 emails are spam. In the last dataset, 1896 emails are spam while the number of ham emails is 4150.
We merged all three datasets into one. Then, we dropped any empty rows that exited. After that, we identified the duplicate entries and they were eliminated. We observed that the total number of ham emails became 12293 where as 3782 emails were spam ones in the dataset. To make the dataset balanced, we applied random undersampling approach. By randomly deleting samples from the majority class, random undersampling equalises the number of examples in both classes and mitigates the likelihood of bias towards the majority class. The objective is to give the model a more representative training set and to balance the distribution of classes. And thus the number of ham emails became same as the number of spam emails in our dataset.

3.2 Pre-Processing

After preparing the balanced dataset, some pre-processing have been done on the dataset for cleaning the data from raw email texts. First, all the email texts were transformed into lowercase. Then, tokenization of the text was done. Then, specials characters, stopwords, punctuations were eliminated. Later, stemming of the text was done.

3.3 Vectorization

After the data has been cleaned through pre-processing, the email texts are converted into numerical vectors. We have experimented with two vectorization techniques. They are Count Vectorizer and TF-IDF Vectorizer. The vectorization process is a significant feature extraction step.

—Count Vectorizer: A feature extraction method used in text mining and natural language processing (NLP) is Count Vectorizer. Tokenization is the initial step in this vectorization procedure, when the text is divided into individual words or concepts. The Count Vectorizer counts the frequency of each term for every document in the collection. The end product is a sparse matrix in which every row denotes a document and every column denotes a distinct phrase throughout the whole dataset. Each document is then given a numerical vector representation based on the phrase frequencies. The input for machine learning algorithms is this vector.

—TF-IDF Vectorizer: Term frequency-inverse document frequency, or TF-IDF In text mining and natural language processing, vectorizers are frequently utilised as feature extraction techniques. The TF-IDF is a statistical measure that quantifies a word's significance to a document inside a corpus or collection. The product of two variables, TF and IDF, is the TF-IDF value. The term frequency (TF) component counts the number of times a phrase appears in a document. It is computed as the ratio of a term's total number of occurrences to its frequency of appearances in a given document. A term's uniqueness or rarity across the corpus of texts is determined by the Inverse Document Frequency (IDF) component. The logarithm of the ratio between the total number of documents and the number of documents that contain the term is used to compute it.

While Count Vectorizer focuses on the frequency of terms in a document, TF-IDF Vectorizer considers the terms' significance within the context of the entire dataset, offering a more nuanced representation that frequently improves performance in a variety of text analysis applications.

3.4 Application of ML Models

After vectorization is completed, we split the dataset into two portions for training and testing different ML models for observing their efficiency in detecting phishing emails. 70% of the dataset is used for training the models whereas the rest 30% data is used for testing purposes.
We have applied a number of ML models. They are mentioned below.

—Support Vector Classifier: The Support Vector Classifier (SVC) is a member of the discriminative classifier family, whose goal is to identify the optimum hyperplane in the feature space to divide the various classes. When the classes in the dataset can be divided linearly, SVC performs well. It looks for a hyperplane in the feature space that maximum divides the classes. Support vectors are the data points that lie closest to the decision boundary (hyperplane) and have the biggest influence on the location of the hyperplane. These are important elements to consider while defining the ideal decision limit. The distance between the closest data point from either class and the decision boundary is known as the margin. By maximising this margin, the model seeks to improve robustness and generalisation.

—K Nearest Neighbor Classifier: A straightforward and efficient method for supervised machine learning is the K Nearest Neighbour (KNN) classifier. Being a non-parametric lazy learning technique, no assumptions are made regarding the distribution of the underlying data, and no explicit model is built during the training phase. Rather, it retains the training set. The method finds the K nearest data points in the training set given a new, unknown data point by using a distance metric. In KNN, the distance measure used to assess how similar two data points are is crucial. When calculating the distances between each training point and the new point during the prediction phase, KNN incurs its largest computing cost.

—Naive Bayes Classifier: The Naive Bayes classifier is a probabilistic ML technique based on Bayes' theorem. It's a straightforward classification algorithm that works well. The naïve assumption in naïve Bayes is that features are conditionally independent given the class label. The algorithm's name comes from this supposition, which makes the computation easier. The probabilities required for the Bayes theorem are estimated from the training data while a Naive Bayes classifier is being trained. Class probabilities and conditional probabilities of characteristics given the class must be calculated in order to do this. The Naive Bayes classifier predicts the class with the highest probability given a new instance with features by calculating the probability of each class given the features.

—Decision Tree Classifier: Commonly used in supervised machine learning is the Decision Tree (DT) Classifier. With each core node representing a quality or attribute, decision rules represented by branches, and the expected result represented by each leaf node, the model resembles a tree. The nodes of a decision tree represent decisions made on the basis of qualities in a hierarchical structure. Each node, representing a class label or a numerical value, is arranged as follows: the roots are the top nodes, and the leaves are the bottom nodes. Question marks about the incoming data are represented as decision nodes in the tree. Until a leaf node is reached, which provides the final judgement or prognosis, these inquiries lead to subsequent nodes

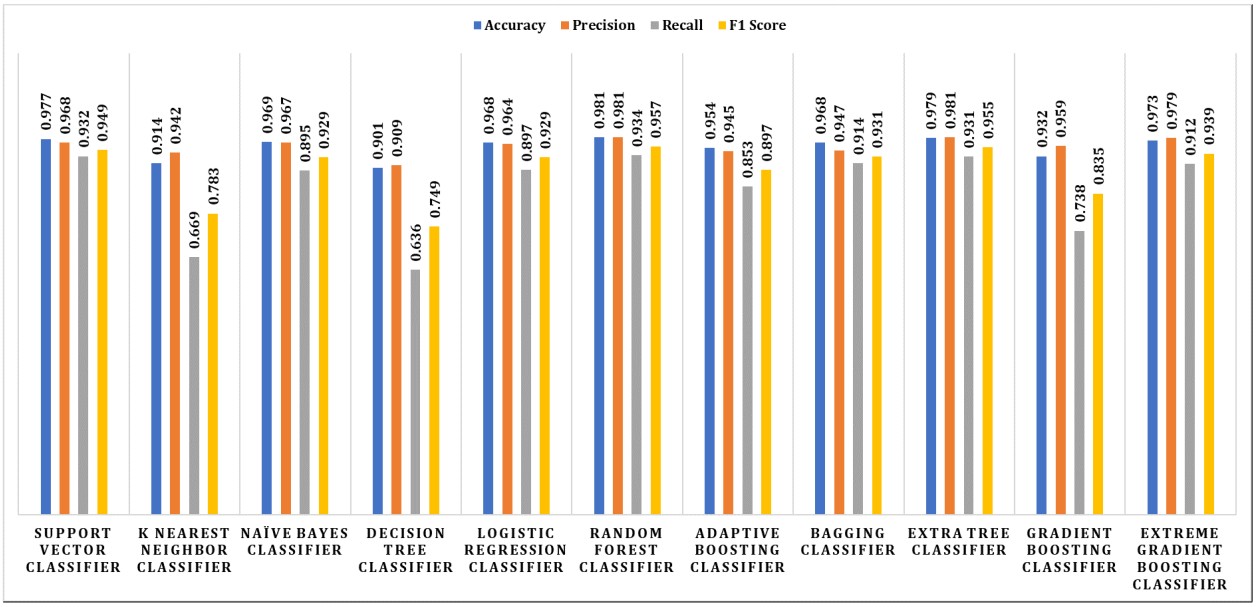

Fig. 1. Performance of ML Models with TF-IDF Vectorizer

in the structure. The approach uses metrics like information gain or Gini impurity to choose the right split for each node.

—Logistic Regression Classifier: One statistical technique for solving classification difficulties is logistic regression. It represents the likelihood that a specific instance is a member of a given class. Any real-valued number can be mapped into the range [0, 1] using the logistic function, which makes it appropriate for binary classification. The likelihood that an instance is a member of the positive class is predicted by the logistic regression model. The instance is classed as positive class if the anticipated probability is greater than or equal to a threshold; otherwise, it is labelled as negative class. The hyperplane that divides instances of various classes in the feature space is known as the decision boundary. The decision boundary for binary classification is the point at which the threshold and the anticipated probability are identical. Maximum likelihood estimation is used to estimate the parameters of a logistic regression.

—Random Forest Classifier: The classification method Random Forest (RF) Classifier generates a large number of decision trees during training and outputs the class mode. The training dataset is randomly picked with replacement to provide many subsets. One of these subgroups is then used to train the forest's decision trees. In a decision tree, a random subset of features is taken into account instead of all features for each split. This increases the trees' diversity and unpredictability. In classification challenges, the final prediction is assigned to the class suggested by the majority of trees. Overfitting is less common in Random Forests than in individual decision trees. The diversity produced by training on various data and feature subsets facilitates generalisation to previously unobserved material.

—Adaptive Boosting Classifier: An ensemble learning technique called Adaptive Boosting (AdaBoost) aggregates the predictions of multiple weak learners to produce a robust and precise predictive model. It gives more weight to events that previous models misclassified, enabling later models to concentrate on the errors. Every data point in the training set has a weight assigned to it, and this weight is modified at each iteration according on how well previously trained models performed. Erroneously classified points carry a higher weight, making them stand out more in later versions. Weak learners receive instruction in a step-by-step fashion, with each new student fixing the errors of the previous ones. The weighted average of the forecasts made by each weak learner makes up the final projection.

—Bagging Classifier: A member of the bagging techniques class of algorithms is the ensemble learning algorithm known as the Bagging Classifier. It operates by independently training several base classifiers on various subsets of the training set, then merging the predictions to minimise overfitting and increase overall accuracy. When there is diversity in the base classifiers and a range of base learner types, bagging works well. By using a technique known as bootstrap sampling, numerous random subsets, or samples, of the training data are created for bagging. A data point may be chosen more than once or not at all inside a particular subset when using bootstrap sampling, which uses random sampling with replacement. Using a basic classifier, the Bagging Classifier trains several instances of the classifier on various bootstrap samples. The heterogeneity brought about by the various training subsets is what allows the base classifiers to be diverse. Because the underlying classifiers can be trained independently and concurrently, bagging is appropriate for distributed and parallel computing. Voting is used in classification tasks to aggregate the predictions of the base classifiers, with the majority vote determining the final prediction.

—Extra Tree Classifier: Within the family of tree-based models is the ensemble learning algorithm known as the Extra Trees Classifier. Extra Trees choose splits for each feature at each node at random, whereas Random Forests build numerous decision trees and choose the optimal split at each node. This is a major distinction between the two types of forests. The Extra Trees Classifier, like Random Forests, is an ensemble

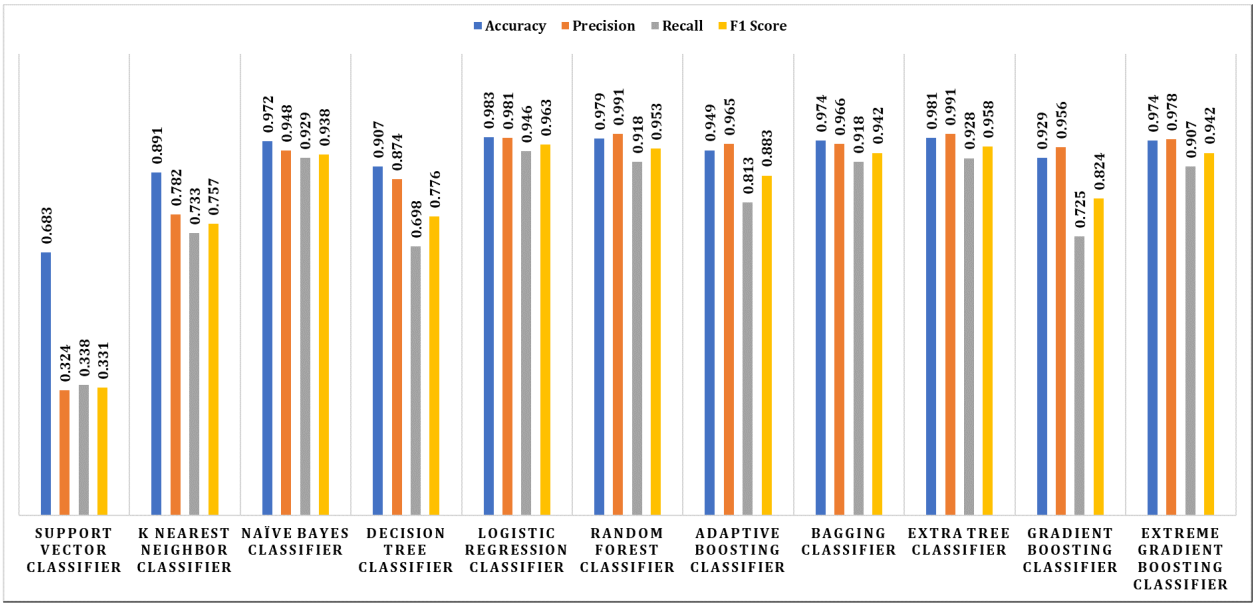

Fig. 2. Performance of ML Models with Count Vectorizer

learning technique that enhances performance and generalisation by constructing several decision trees and combining their predictions. Rather than evaluating every feature at each decision tree node to determine the optimal split, Extra Trees selects a subset of features at random and selects the best split from this subset. By adding additional randomness, this lessens overfitting. In addition to random feature selection, Extra Trees also utilises random thresholds for splitting nodes. As a result, the diversity of the trees in the ensemble is further increased.

—Gradient Boosting Classifier: Gradient Boosting Classifier (GBC) is an ensemble learning approach that combines the predictions of multiple weak learners to generate a powerful predictive model. The method builds a group of incompetent students gradually. Collectively, the incoming students fix the errors made by the current students. The strategy minimises a loss function by using gradient descent optimisation. The model parameters are adjusted to reduce the loss when the loss gradient is computed in relation to the ensemble forecast. Gradient Boosting is focused with reducing the residuals, or mistakes, of the previous ensemble's predictions. Eventually, the total error is decreased as more trees are trained to forecast the residuals. The contribution of each tree to the ensemble is controlled by a shrinkage parameter, also called the learning rate. More trees are needed for the same level of complexity when learning at a slower rate, however this can increase the robustness of the model.

—Extreme Gradient Boosting Classifier: Extreme Gradient Boosting (XGBoost) is a popular and potent machine learning algorithm that belongs to the boosting subclass of ensemble learning techniques. With an emphasis on flexibility, scalability, and computing economy, it is an expansion of the gradient boosting framework. This method involves gradually adding weak learners to the model in order to fix mistakes produced by the previous model. To prevent overfitting, XGBoost includes L1 (Lasso) and L2 (Ridge) regularisation in the objective function. This enhances generalisation to new data and keeps the model from getting overly complicated. By cutting off branches that don't significantly increase prediction power, the method uses pruning to regulate the size of the decision trees.

3.5 Performance Evaluation of ML Models

For evaluation of the ML models, we have used for evaluation metrics. They are Accuracy, Precision, Recall and F1 Score. As the percentage of correctly predicted occurrences to total instances, accuracy indicates the model's overall preciseness. Precision gauges how well the model predicts the good outcomes. It is defined as the proportion of accurately anticipated positive observations to all positive predictions. Recall quantifies the model's capacity to record every instance of success. It is the proportion of all actual positive observations to all correctly predicted positive observations. The harmonic mean of recall and precision is the F1 score. It offers a balance between recall and precision. It is especially helpful in cases where the distribution of classes is not uniform.

In Figure 1, the performances of the models are illustrated when the TF-IDF Vectorizer is employed for vectorization. The Random Forest classifier has outperformed others overall in this scenario. In Figure 2, the performances of the models are illustrated when the the Count Vectorizer is used for vectorization process. The Logistic Regression Classifier, Random Forest Classifier and the Extreme Tree Classifier have performed significantly better than others overall in this scenario.

4. CONCLUSION AND FUTURE WORKS

In our research work, we have experimented to see how effectively ML models can identify a phishing email. If emails are correctly categorized, harms can be prevented. We have made use of different datasets, used two vectorization process and different ML models. The models have demonstrated a notable ability to distinguish phishing emails. We intend to apply more ML models,

pre-processing and vectorization techniques, and datasets in our future experiments.

5. REFERENCES

[1] Ding, Xiong, Baoxu Liu, Zhengwei Jiang, Qiuyun Wang, and Liling Xin. "Spear phishing emails detection based on machine learning." In *2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pp. 354-359. IEEE, 2021.

[2] Odeh, Ammar, Ismail Keshta, and Eman Abdelfattah. "PHIBOOST-a novel phishing detection model using Adaptive boosting approach." *Jordanian Journal of Computers and Information Technology (JJCIT)* 7, no. 01 (2021).

[3] Kalabarige, Lakshmana Rao, Routhu Srinivasa Rao, Ajith Abraham, and Lubna Abdelkareim Gabralla. "Multilayer stacked ensemble learning model to detect phishing websites." *IEEE Access* 10 (2022): 79543-79552.

[4] Kalabarige, Lakshmana Rao, Routhu Srinivasa Rao, Alwyn R. Pais, and Lubna Abdelkareim Gabralla. "A Boosting based Hybrid Feature Selection and Multi-layer Stacked Ensemble Learning Model to detect phishing websites." *IEEE Access* (2023).

[5] Raj, Mukta Mithra, and J. Angel Arul Jothi. "Hybrid Approach for Phishing Website Detection Using Classification Algorithms." *ParadigmPlus* 3, no. 3 (2022): 16-29.

[6] Samad, Dadvandipour, and Ganie Aadil Gani. "Analyzing and predicting spear-phishing using machine learning methods." *Multidiszciplináris Tudományok* 10, no. 4 (2020): 262-273.

[7] Atmojo, Yohanes Priyo, I. Made Darma Susila, Muhammad Riza Hilmi, Erma Sulistyo Rini, Lilis Yuningsih, and Dandy Pramana Hostiadi. "A New Approach for Spear phishing Detection." In *2021 3rd East Indonesia Conference on Computer and Information Technology (EIConCIT)*, pp. 49-54. IEEE, 2021.

[8] Cazares, María Fernanda, Roberto Andrade, Gustavo Navas, Walter Fuertes, and Jhonathan Herrera. "Characterizing phishing attacks using natural language processing." In *2021 Fifth World Conference on Smart Trends in Systems Security and Sustainability (WorldS4)*, pp. 224-229. IEEE, 2021.

[9] Dewis, M., and T. Viana. "Phish responder: A Hybrid machine learning approach to detect phishing and spam emails," Applied System Innovation" 5, 73." (2022).

[10] "Kaggle", [Online] Available: https://www.kaggle.com/ , Last Accessed on: 10 June 2025.