Enhanced Detection of Phishing Emails: A Machine Learning Approach for Cybersecurity

Anirudh S.
Department of Computer Science and Engineering
Amrita School of Computing, Bengaluru
Amrita Vishwa Vidyapeetham, India
bl.en.u4cse21020@bl.students.amrita.edu

P. Radha Nishant
Department of Computer Science and Engineering
Amrita School of Computing, Bengaluru
Amrita Vishwa Vidyapeetham, India
bl.en.u4cse21161@bl.students.amrita.edu

Sanjay Baitha
Department of Computer Science and Engineering
Amrita School of Computing, Bengaluru
Amrita Vishwa Vidyapeetham, India
bl.en.u4cse21185@bl.students.amrita.edu

***Abstract*—Phishing emails, disguised as legitimate communication, pose a significant threat in today's digital landscape. These deceptive emails attempt to extract sensitive information or induce malicious actions from recipients, making them a potent tool for cybercriminals. This research delves into the critical challenge of phishing email detection using the Kaggle dataset and advanced classification techniques. By thoroughly analyzing email attributes, this study not only identifies these malicious emails but also highlights the vital importance of countering phishing threats. The findings not only enhance our understanding of these deceptive tactics but also offer effective strategies to bolster email security, ensuring a safer online environment for individuals and organizations.**

***Index Terms*—phishing emails, legitimate communication, advanced classification techniques, email security.**

I. Introduction

One cannot stress the value of cybersecurity in the digital age particularly when it comes to email. One of the most popular tools for both personal and professional lives is email, but it also poses a lot of vulnerabilities. Sensitive information is frequently sent by email hence protecting it against breaches is necessary to maintain integrity and privacy. This reliance on email is a challenge as phishing attacks [1] happen a lot. The goal of these sophisticated cyber attacks is to steal sensitive information like logins and financial details by using deceptive emails that look like legitimate sources. Phishing techniques [2] are constantly evolving which make it a serious threat. In order to detect and prevent these attacks cybercriminals are constantly adapting their strategies to evade standard security protocols. Thus, a dynamic and robust approach is necessary to improve mail security An advanced machine learning approach [3] is used to classify mails into 'Phishing' or 'Safe'. Initially the dataset is cleaned and converted to the required format using TF-IDF vectorization. Support vector machines are employed for its ability in managing high dimensional dataset and XGBoost is also used for scalability and efficiency. The model is then integrated using a meta-learner which makes the final predictive decisions. This paper also incorporates exploratory data analysis to understand the characteristics of mails.

II. Literature Review

Pankaj et al. [4] talk about how to combat email phishing using a machine learning based approach. They extracted features from the emails using Natural Language Processing and gave them the labels as non harmful (benign) or harmful (phishing). They applied the comparative analysis on various models including SupportVector Machine (SVM), Naive Bayes classifier, Haphazard Woodland (Random forest), Logistic reversion, and Voted Perceptron. It is found that the random forest method gave the best result to find phished emails. However the major issues to this approach is that the dataset used is not representative of the real world.

Weina et al. [5] detection of the phished emails using a proposed Machine learning hybrid model approach. They use a proposed Cuckoo Search Support Vector Machine (CS-SVM) to extract the 23 features and train the dataset collected by Jose Nazario and Calo project and compared with the baseline SVM model. It is found that the proposed model fared better than the baseline model. There is still scope to optimize the model and run it on the distributed platform and check the influence of other sample data.

Mustafa et al. [6] propose the detection of phishing emails as a problem which can be solved using classification to categorize emails as phished or not. They compared the effectiveness of three machine learning methods, Naive Bayes classifier, Random Forests classifier, and Support Vector Machine(SVM). It is found that the Support Vector Machine gave the best result for detecting phishing emails. It provides valuable insights into the features that are most indicative of phishing emails, which can be used to develop more effective anti-phishing strategies. J.Ramprasath et al. [7] focus to explore the use of deep learning algorithms for the identification and protection against phishing email attacks. They proposed a model for identifying Spam and Ham emails using a Recurrent Neural Network (RNN) model with Long

Short-Term Memory (LSTM) cells. They compare the performance of their proposed technique with the Support Vector Machine (SVM) and Constrained k-Nearest Neighbor (CkNN) classifier. It is found that the RNN algorithm gave an effective approach for identifying Spam and Ham emails. However the major issue to this approach is that using Recurrent Neural Networks (RNNs) for spam and ham email identification can be computationally expensive and requires a large volume of training data to attain high accuracy. Lingampally et al. [8] aim to detect phished email using multiple machine learning and deep learning approaches. They have extracted 11 attributes and they had taken LA-BEL as a target attribute and removed of its imbalance using tokenization, lemmatization and vectorization, and implemented using machine learning models and deep learning including Linear Regression, Logistic Regression, Random Forest, XG Boosting, Decision Tree classifier, Support Vector Machine and K-Nearest Neighbors (k-NN) classifier. They developed a model using Random Forest that achieved highly accurate results in distinguishing between genuine and fraudulent emails, showcasing its effectiveness in email detection. They plan to use a few filtering and blocking strategies in their next work to stop phishing emails. Nishant et al. [9] talk about how to detect email phishing using a machine learning and deep learning approach.They applied the comparative analysis on various models including Logistic Regression, Naive Bayes, Decision Tree Classifier, Support Vector Machine(SVM), Long Short-Term Memory (LSTM), Bidirectional LSTM (Bi-LSTM), and Convolutional Neural Networks (CNN).The deep learning techniques like LSTM, Bi-LSTM, and CNN were found to be the best model to detect phishing email. The only drawback is that loss in testing scores to the training scores is higher for CNN. Sikha et al. [10] aim to classify phishing email using machine learning and deep learning approaches. They conducted a comparison of deep learning classifiers such as Long Short Term Memory (LSTM), Convolutional Neural Networks (CNN), and Word Embedding, and machine learning classifiers such as Naïve Bayes, Support Vector Machines (SVM), and Decision Trees. It is found that the Word Embedding model gave the best result to classifying phishing and non-phishing emails. Deep learning models, particularly LSTM and CNN, took longer training times compared to some traditional machine learning algorithms. Md Fazale et al. [11] detecting phishing emails using machine learning and natural language processing approaches. The comparative analysis of these models such as Logistic Regression (LR),K-Nearest Neighbors (KNN),AdaBoost(AB),Multinomial Naive Bayes (MNB),Gradient Boosting (GB),Random Forest (RF). It is found that the random forest method gave the best result to detect phishing emails. The issues for the models are training time constraint, and the dependency on the specific 'subject' feature. Md Abdullah Al Ahasan et. Al. [12] talks about the identification of phishing email using Optimized fuzzy multi-criteria decision making (OFMCDM) and Improved Random Forest (IRD). They created the dataset from real life data collected from PhishTank which has around 10,000 URLs. They used OFMCDM to extract the features and IRD for training the model. They conducted a comparative analysis with other models that include Naïve Bayes (NB), Logistic Regression (LR), K-Nearest neighbor (KNN), and Decision Tree. It was found that the OFMCDM/IRD has is more efficient than other models. Currently, they have extracted the features that have already been retrieved from the papers, they have done research, and the code is inefficient because it takes a long time to execute Ishwarya et. al. [13] discusses the detection of phishing emails using probabilistic classifiers like naïve bayes, k-Nearest Neighbor (KNN), Single Vector Machine (SVM) and Random Forest (RF). The Features are extracted as input and output manually without using any algorithms. The Dataset is divided in the ratio of 70:30 and trained on the 70 data with all the algorithms mentioned earlier and compared. It was observed that the Naïve Bayes comes as better algorithm than others with better accuracy. Since the existing dataset does not reflect the application of emails in real-time, it might not be adequate. Multiple email URLs will be utilized to combat this and resolve the problem. Sadia Parvin Ripa et. al. [14] talks about the detection of phishing urls using different machine learning approaches. The Dataset used for phishing email is a mix of github and Kaggle which contains around 9499 entries They extracted 5 features manually without use of any algorithms. They trained the dataset using XGBoost and compared it with other models like KNN, Random Forest, Decision Tree and others. It was observed that XGboost has better accuracy compared to other algorithms. It is a collection of real-time browser data, particularly emails from a few research studies, was utilized as the dataset in this paper. Framework will be added to connect to the browser in order to resolve this problem Sami Smadi et. Al. [15] discusses the six different classification algorithms used to classify emails as phishing or legitimate including the C4.5, Naive Bayes, SVM, Linear Regression and K-Nearest Neighbour. The model uses the J48 algorithm to express the structure of the data and ten-fold cross validation is used to test and train the model. The model used 23 hybrid features to demonstrate the results to classify the emails. Of all the classification algorithms Random forest gave high accuracy rate and true positive rates. The main drawback of the model is the selection of emails for the test case is done manually which may be causing the inconsistency for few metrics. Prasanta Kumar Sahoo et. al. [16] talks about the use of data mining algorithms to detect the phishing attacks. They applied text mining to extract the features from the dataset like personal, official, financial and others. They split the dataset into train and test in the ratio 60:40 and trained the dataset using a naïve bayes model. They took Gemini ai which is an ai solution for evaluating ml solutions, which took a parameter "Page Load Time" to evaluate the model. More the page load time, less the efficiency. From the observations from different browsers, it was observed that the detection of phishing websites is high. Isredza Rahmi A Hamidet et. al. [17] discusses the implementation of phishing email detection using hybrid feature selection approach based on content-based and behavior-based. They extracted 7 features using hybrid feature selection and trained the model using data mining algorithms. They took 3 different datasets from various sources which was performed using WEKA (Waikato Environment for Knowledge Analysis) which consists of around 3000 phishing emails that were collected from November 2004 to November 2005. They conducted a comparative analysis on various algorithms using two types of data split (60:40 and 70:30). For 60:40, Bayes Net attained highest accuracy and for 70:30, Random forest attained the highest accuracy. Mahmoud Khonji et. Al. [18] discusses an URL approach to detect the phishing emails that redirect to a website which is malicious. It uses lexical URL analysis approach to identify the website that has malicious virus. Using wrapper and best-fit methods, 47 features are extracted. They collected two types of datasets in which one dataset contains 4,116 emails and the other dataset contains 4150 emails. The features are divided into 6 subsets, in which the feature subset 3-A comes better efficiently than other feature subsets as it has better precision and recall. They trained the dataset using Random Forest Classification. Gal Egozi et. al. [19] talks about the detection of phishing emails using robust nlp techniques. They extracted around 26 features. They tested around 17 models to check the better algorithms, in which 14 algorithms gave acceptable results. The dataset used is collected from IWSPA competition which has around 9000 entries which was collected from Wikileaks, SpamAssassin, Nazario Phishing corpora and others. Using Single machine learning algorithm, Bernoulli's Naïve bayes, Decision tree, Linear kernel SVM, Gaussian Naïve bayes, all attained the same accuracy irrespective of weighted or unweighted. Using ensembled method, The Logistic regression model comes as a better algorithm with better accuracy. The process of obtaining the features has a limit and is inefficient. A new method will be utilized to extract more features and provide greater accuracy in order to increase efficiency.

In this stacking model, SVM and XGBoost have been opted as primary classifiers due to their proficiency in identifying complex patterns in the email dataset. The SVM is particularly adept at managing the extensive array of words derived from text-to-numerical conversion. On the other hand, XGBoost is valuable for its cautious approach in making predictions, thus enhancing the model's accuracy. For the final layer, Logistic Regression is being employed. This model is less complex and effectively consolidates the insights from SVM and XGBoost, thereby finalizing the classification of emails. This combination is strategically beneficial as it combines different analytical strengths, resulting in a precise email classification system.

III. Methodology

A. Support Vector Machine

Support Vector machines (SVMs) [20] are a set of supervised learning algorithms that are known for their effectiveness in classification and regression tasks. SVMs are used in a wide range of applications from image recognition to biometrics. SVMs are based on the concept of finding the optimal hyperplane which is a decision boundary that maximizes the margin between two classes of data points. This hyperplane is usually visualized as a line in two-dimensional space or a plane in higher dimensions which separates the two classes ensuring accurate classification. SVMs can work really well with high dimensional data in the real world. Most data exist in multiple dimensions and SVMs overcome this by employing kernel functions that project the data into a higher dimensional space where linear separation becomes possible. SVM is also not influenced by outliers unlike other models as it considers the dataset closes to the hyperplane which results in more accurate and reliable prediction Unfortunately SVMs can be computationally expensive especially for complex datasets and require careful hyperparameter tuning to achieve optimal performance . It's vital to select the appropriate kernel function and tune its hyperparameters otherwise it would affect the models ability to handle complex datasets.Machine learning problems are nevertheless often solved using SVMs, regardless of challenges faced.

B. Xgboost

A popular machine learning algorithm called XGBoost[21] is utilized for producing incredibly precise results because it creates a single, reliable model by merging many decision trees which makes this technique incredibly efficient.The total strength of XGBoost is increased as each tree in the sequence learns from the mistakes made by the trees before it. Additionally, a regularization approach is used to assist prevent overfitting and guarantee that the model performs well when applied to fresh, untested data. XGBoost is well known for its efficiency and speed and it performs exceptionally well with large and complex datasets because of these features, XGBoost is preferred among data scientists especially for predictive modeling jobs like identifying emails as spam or not.

C. Logistic Regression

Logistic Regression is a statistical method that allows us to classify things into two distinct groups. It tries to predict the likelihood that an input will be classified as either '0' or '1'. The logistic function is a mathematical curve that resembles the letter 'S' and is utilized in this algorithm. It can take any real number between 0 and 1 as input. In logistic regression, we use a logistic function to model the probability of the positive class by applying it to a linear combination of the input features. The model utilizes the features you provide to make an educated guess, represented as a logit which is the logarithm of the probability of the event occurring. The logistic function transforms the data into a probability. Part of this process involves finding the best coefficients for the features. Typically, this is accomplished using a method such as maximum likelihood estimation.The algorithm then searches for coefficients that increase the likelihood of observing the specified set of outcomes. Typically, optimization methods like gradient descent are used for this purpose.

Logistic Regression is an optimal choice as a meta-learner in ensemble methods, particularly when it comes to stacking models. Stacking is a technique that involves combining multiple base models, each using a different algorithm. These models make individual predictions, which are then combined by another model called the meta-learner. In this scenario, Logistic Regression is frequently selected as the meta-learner because it can interpret the outputs of the base models in terms of probabilities or class membership. The model uses these outputs as features to make predictions about what will happen in the end. Logistic Regression is considered a good meta-learner due to its simplicity and ability to provide probabilistic outcomes. This approach reduces the chances of overfitting when combining predictions from complex base models. This method is effective for binary classification tasks because it leverages the strengths of Logistic Regression. Logistic Regression provides calibrated probabilities and is easy to interpret, which is crucial for understanding how predictions are made by different base learners

D. Proposed Method

Data is organized using the Pandas library which is important for efficient data management. Patterns within the text are identified and corrected through regular expressions which plays a crucial step in data cleansing. The data is cleaned up by taking out unnecessary words that don't add much meaning to a sentence which are called stop words . A special list from a tool called Scikit-learn is used to find and remove these words. To simplify the text further, a tool from the nltk library called PorterStemmer is used to break down words to their root form. This process is called stemming and helps the computer to understand different forms of the same word as one single word. The refined text is converted into a numerical format using the TfidfVectorizer, with a maximum feature limit set to maintain computational efficiency. At the same time the categorical variables are numerically encoded through Label Encoding, facilitating their use in algorithmic analysis. Data visualization is undertaken using Matplotlib and Seaborn, producing a series of plots that reveal insights into email lengths and the distribution of email types. These visualizations also explore the correlation between email lengths and word counts, providing a deeper understanding of the dataset. In the detection of phishing emails, a stacking ensemble approach is utilized integrating the strengths of SVM and XGBoost models which act as base learners and Logistic Regression as the final decision-maker. The dataset is partitioned dedicating 75% to training and the remaining 25% to testing which ensures the model is well-equipped to identify phishing threats.The effectiveness of this methodology is measured by the accuracy,precision,recall and F1 -score which gauges the model's predictive performance. This proposed methodology is designed to enhance email security through accurate identification of phishing activities.

E. Architecture

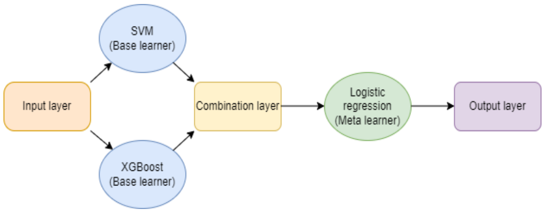

Fig. 1. System Architecture.

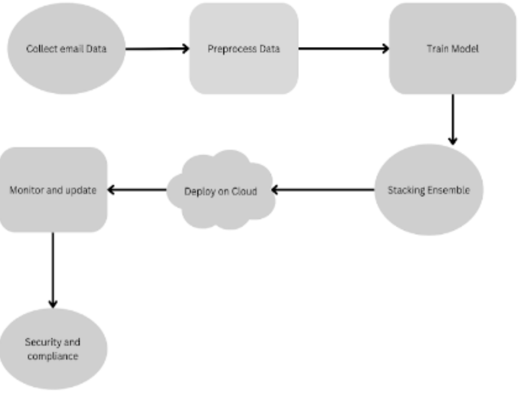

Fig. 2. Real Life Implementation.

IV. Result Analysis

A. Dataset

The dataset we're examining includes a total of 18,650 emails, each tagged under one of two categories in the 'Email Type' column. These categories are 'Safe Email' for regular, harmless messages, and 'Phishing Email' for emails that might be trying to deceive the recipient. Out of the total, 11,322 emails are marked as 'Safe Email', showing they are typical non-malicious communications. The remaining 7,328 are identified as 'Phishing Email', indicating they could be potential threats .An important point to note in this dataset is the presence of some missing values. Specifically, there are 16 emails where the 'Email Text' column, which is essential for understanding the content and for any machine learning work, is empty. This lack of data is a critical issue and needs to be taken care of in the early stages of data processing. Addressing this will help ensure that any analysis or machine learning model built using this dataset is accurate and trustworthy.

B. Data Visualization

Data visualization plays a critical role in the interpretation and analysis of complex datasets. In our phishing email detection project, three types of plots provide valuable insights: Line Plot for Email Length: This visualization showcases the variation in the length of emails within the dataset. It helps in identifying any outliers or anomalies, such as emails with exceptionally long content which may warrant further investigation. Bar Chart for Class Distribution: The bar chart reveals the distribution between safe and phishing emails. Understanding this balance is crucial for evaluating the dataset's bias and ensuring that our machine learning models are not skewed towards one class. Scatter Plot of Email Length vs.

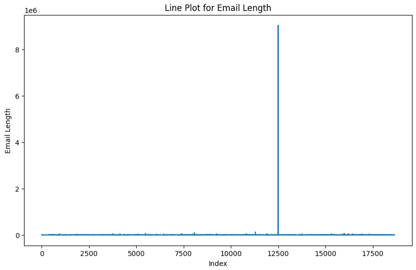

Fig. 3. Example of a figure caption.

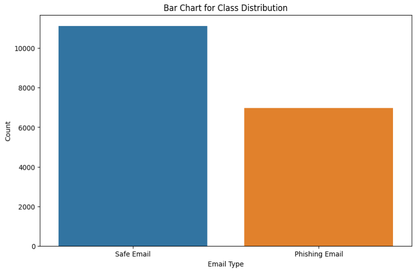

Fig. 4. Example of a figure caption.

Word Count: By plotting email length against word count on a logarithmic scale, we can observe patterns and correlations between the two metrics. This scatter plot helps to understand the relationship between email verbosity and wordiness, which could be indicative of either safe or phishing emails.

C. Metrices

We have various metrics that tell us the efficiency and performance of the program. But mainly, we use five metrics to evaluate the efficiency of the code. They are:

1) 1)Accuracy
2) 2)Precision
3) 3) Recall
4) 4) F1-Score

An impressive 97.92 accuracy is achieved with the stacking ensemble model. In terms of identifying both types of emails, this model has been pretty accurate; it's identified 2754 true positives (legitimate phishing emails) and 1677 true negatives (genuine safe emails). It only misclassifies a small number of emails, with 44 false positives (safe emails incorrectly labeled as phishing) and 50 false negatives (phishing emails missed and labeled as safe). 98.43 of the emails it labeled as phishing were correctly identified by the model. In the dataset, almost all of the phishing emails were found by the model, with a recall rate of 98.22. Finally, the F1 Score, at 98.32, is a balanced metric that takes both precision and recall into account, and this shows how accurate the model is at detecting safe and phishing emails.

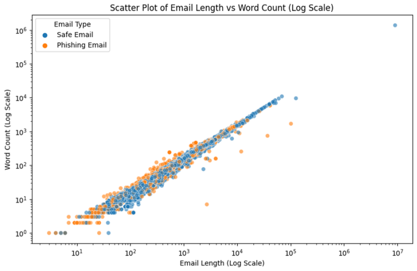

Fig. 5. Example of a figure caption.

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

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

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

$$\text{F1-Score} = \frac{2*Precision*Recall}{Precision+Recall}$$

$$\text{Specificity} = \frac{TN}{TN+FP}$$

Fig. 6. Metrices

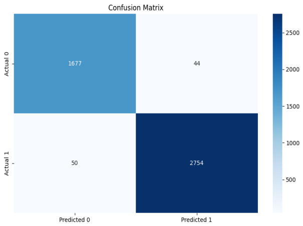

Fig. 7. Confusion Matrix for the proposed method

In comparison with the proposed model, SVM obtained an accuracy of 96.74 which is slightly low.It received a precision score of 98.05, which is still very good for correctly identifying phishing in the email.It also received a recall score of 96.52, which shows that the false negative effect the model a lot more and an overall F1-score of 97.28. It classified 1790 non-phishing emails and 2721 phishing emails correctly. It classified 54 emails as non-phishing emails and 98 phishing emails incorrectly.

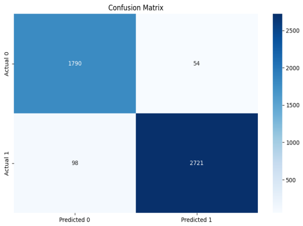

Fig. 8. Confusion Matrix for SVM

XGBoost obtained an accuracy of 96.05 which is slightly lower than any other model. It received a precision score of 97.75 which is still good for correctly identifying phishing in the email. It also received a recall score of 95.67 which shows that the false negative affected the model a lot more than any other model, and an overall F1-score of 96.70. It classified 1782 non-phishing emails and 2697 phishing emails correctly. It classified 122 phishing emails and 62 emails as non-phishing emails incorrectly.

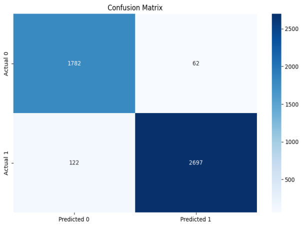

Fig. 9. Confusion Matrix for XGBoost

Random Forest obtained an accuracy of 97.06 which is slightly low. It received a precision score of 97.92 which is still good for correctly identifying phishing in the email. It also received a recall score of 97.33 which shows that the false negative affected the model a lot more than the proposed method, and an overall F1-score of 97.62. It classified 1663 non-phishing emails and 2729 phishing emails correctly. It classified 58 emails as non-phishing emails and 75 phishing emails incorrectly.

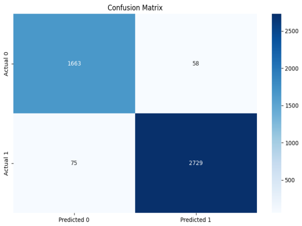

Fig. 10. Confusion Matrix for Random Forest

Multinomial Naive Bayes (MNB) obtained accuracy of 96.22 which is slightly lower than the proposed method. It received a precision score of 96.36, which is also worse than any other method, identifying phishing in the email. It also received a recall score of 97.36 which shows that the false negative affects the MNB a lot less compared to the proposed method, and an overall F1-score of 96.96. It classified 1624 non-phishing emails and 2730 phishing emails correctly. It classified 97 emails as non-phishing emails and 74 phishing emails incorrectly.

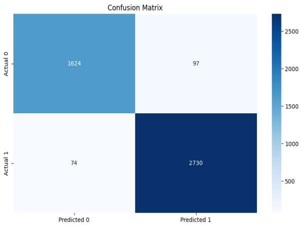

Fig. 11. Confusion matrix for Multinomial Naive Bayes

V. Conclusion

The model that is used in this paper produces remarkable performance metrics—97.92 accuracy, 98.43 precision, 98.22 recall, 98.32 F1 score, and 98.21 specificity. Using XGBoost, along with features extracted like word count and email length, significantly contributed to improving the model's accuracy. Comparative analysis with Naive Bayes, SVM, Random Forest, and Neural Network gives us the result that Xgboost is superior to others. However, there is a concern regarding the code's computational time, that makes our priority as future work. The future work emphasizes refining the model's performance and accuracy reflecting a proactive pursuit of advancing predictive capabilities in email classification.

Model	Accuracy	Precision	Recall	F1-Score
SVM	96.74	98.05	96.52	97.28
XGBoost	96.05	97.75	95.67	96.70
Random Forest	97.06	97.92	97.33	97.62
Multinomial Naive Bayes	96.22	96.57	97.36	96.96
Proposed method	97.92	98.43	98.22	98.32

References

[1] Hiransha M, Unnithan NA, Vinayakumar R, Soman K, Verma AD (2018) "Deep learning-based phishing e-mail detection" *1st AntiPhishing Shared Pilot 4th ACM Int. Workshop Secur. Privacy Anal.(IWSPA)*, 1-5.

[2] Harikrishnan NB, Vinayakumar R, Soman KP (2018) "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)*, 455-468

[3] Unnithan NA, Harikrishnan NB, Vinayakumar R, Soman KP, Sundarakrishna S (2018) "Detecting phishing E-mail using machine learning techniques." *In Proc. 1st anti-phishing shared task pilot 4th acm iwspa co-located 8th acm conf. data appl. secur. privacy (codaspy)*, 51-54.

[4] Saraswat P, Solanki MS. (2022) "Phishing Detection in E-mails using Machine Learning". *In 2022 2nd International Conference on Technological Advancements in Computational Sciences (ICTACS)*, 420-424.

[5] Niu W, Zhang X, Yang G, Ma Z, Zhuo Z (2017). "Phishing emails detection using CS-SVM". *In 2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC)*, 1054-1059

[6] Al Fayoumi M, Odeh A, Keshta I, Aboshgifa A, AlHajahjeh T, Abdulraheem R (2022). "Email phishing detection based on naïve Bayes, Random Forests, and SVM classifications: A comparative study". *In 2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)*, 0007-0011.

[7] Ramprasath J, Priyanka S, Manudev R, Gokul M (2023). "Identification and mitigation of phishing email attacks using deep learning". *In 2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, 466-470

[8] Shalini L, Manvi SS, Gowda NC, Manasa KN (2022). "Detection of Phishing Emails using Machine Learning and Deep Learning". *In 2022 7th International Conference on Communication and Electronics Systems (ICCES)*, 1237-1243.

[9] Paradkar NS (2023). "Phishing Email's Detection Using Machine Learning and Deep Learning". *In 2023 3rd International Conference on Advances in Computing, Communication, Embedded and Secure Systems (ACCESS)*, 160-162.

[10] Bagui S, Nandi D, Bagui S, White RJ (2019) "Classifying phishing email using machine learning and deep learning". *In 2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*, 1-2.

[11] Rabbi MF, Champa AI, Zibran MF (2023) "Phishy? Detecting Phishing Emails Using ML and NLP". *In 2023 IEEE/ACIS 21st International Conference on Software Engineering Research, Management and Applications (SERA)*, 77-83.

[12] Almomani A, Gupta BB, Atawneh S, Meulenberg A, Almomani E (2013) "*A survey of phishing email filtering techniques". IEEE communications surveys tutorials* **15(4)**: 2070-2090.

[13] Ishwarya et al. Separation of Phishing Emails Using Probabilistic Classifiers.

[14] Somesha M et al. Classification of Phishing Email Using Word Embedding and Machine Learning Techniques

[15] Kang et al. A new hybrid ensemble feature selection framework for machine learning-based phishing detection system

[16] Molly et al. A Hybrid Machine Learning Approach to Detect Phishing and Spam Emails

[17] Md Abdullah et al. A Phishing Website Detection Model based on Optimized Fuzzy Multi-Criteria Decision-Making and Improved Random Forest

[18] Isredza et al. Using Feature Selection and Classification Scheme for Automating Phishing Email Detection

[19] Mahmoud et al. Lexical URL Analysis for Discriminating Phishing and Legitimate E-Mail Messages

[20] Akhil, V. M., K. J. Chandan, Deepa Itagi, and K. R. Prakash. "Analyses of different methods of writing using SVM classifier." In 2021 2nd International Conference on Communication, Computing and Industry 4.0 (C2I4), pp. 1-5. IEEE, 2021.

[21] K. Laxmikant, R. Bhuvaneswari and B. Natarajan, "An Efficient Approach to Detect Diabetes using XGBoost Classifier," 2023 Winter Summit on Smart Computing and Networks (WiSSCoN), Chennai, India, 2023, pp. 1-8, doi: 10.1109/WiSSCoN56857.2023.10133854.

[22] Ra, V., HBa, B.G., Ma, A.K., KPa, S., Poornachandran, P. and Verma, A., 2018. DeepAnti-PhishNet: Applying deep neural networks for phishing email detection. In Proc. 1st AntiPhishing Shared Pilot 4th ACM Int. Workshop Secur. Privacy Anal.(IWSPA) (pp. 1-11). Tempe, AZ, USA.