IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

PHISHING MAIL DETECTION USING MACHINE LEARNING

[1]B Leela Venkata Sai Ram, [2] A Mary Sowjanya
[1]M.Tech, [2]Associate Professor
[1,2]Department of CS&SE,
[1,2] Andhra University, Visakhapatnam, India

Abstract: Phishing attacks continue to be a significant cyber-security concern, targeting individuals and organizations worldwide. As phishing techniques become increasingly sophisticated, conventional rule-based methods struggle to keep pace with the evolving threat landscape. To address this challenge, Machine Learning (ML) has emerged as a promising approach for detecting phishing emails effectively. In this work, we propose a phishing email detection system that leverages Machine Learning algorithms to automatically analyze and identify malicious emails from legitimate ones. This research involves building a comprehensive dataset consisting of labeled examples of both phishing and legitimate emails and extracting relevant features from email content and metadata. We use supervised learning techniques such as Support Vector Machines (SVM), Decision tree and Random Forest to train and fine-tune our detection models.

***Key Words* - Phishing, Email detection, Support Vector Machines (SVM), Decision tree, Random Forest, Machine Learning.**

I.INTRODUCTION

Phishing attacks have become increasingly prevalent in the digital landscape, posing significant threats to individuals and organizations alike. Phishing emails are deceptive messages designed to trick recipients into revealing sensitive information, such as login credentials, financial data, or personal details. As these attacks become more sophisticated and harder to distinguish from legitimate communications, traditional rule-based approaches struggle to keep up with the evolving tactics of cybercriminals. To combat the ever-changing nature of phishing attacks, machine learning (ML) has emerged as a powerful tool in email security. ML algorithms can analyze vast amounts of data, identify patterns, and learn from historical examples to detect subtle characteristics of phishing emails. By automating the detection process, ML-based phishing email detection systems can efficiently analyze incoming emails and flag potentially malicious ones in real-time, significantly enhancing the overall cyber security posture. Phishing attacks have reached unprecedented levels especially with emerging technologies such as mobile and social media [1]. For instance, from 2017 to 2020, phishing attacks have increased from 72 to 86% among businesses in the United Kingdom in which a large proportion of the attacks are originated from social media.

The goal of this project is to develop an effective phishing email detection system using machine learning techniques. By leveraging a diverse dataset of legitimate and phishing emails, we aim to train and fine-tune ML models to accurately distinguish between benign and malicious messages. The underlying ML algorithms may include supervised learning techniques like Support Vector Machines (SVM), Random Forest and Decision tree.

II.LITERATURE REVIEW

Due to its effect on users' security, the detection of phishing emails has recently drawn a lot of attention. As a result, numerous methods have been developed to identify phishing emails, ranging from communication-oriented methods like authentication protocols, blacklisting, and white-listing, to method based on content. Although they have not yet been shown to be sufficiently effective when applied to many domains, the blacklisting and white-listing procedures are not widely used. The content-based phishing filters, meanwhile, are extensively utilized and have a high level of effectiveness. Research has concentrated on creating machine learning and data mining approaches based on the header and body of emails, as well as content-based mechanisms, as a result of this.

An analysis of the effectiveness of the available phishing detection systems was done in 2007. According to this survey, over 20% of phishing websites were missed by even the top phishing detection toolbars [2]. Another study from 2009 found that the majority of anti-phishing solutions did not begin blocking phishing sites until several hours or days after the phishing emails were delivered to lure users [3]. As a result, I draw the conclusion that the detection techniques now in use do not totally (100%) identify these phishing emails and websites [4]. Toolan created a new C5.0 algorithm to filter data into categories for phishing and non-phishing [5]. 8,000 emails were included in the sampled data, with half of them being phishing scams and the other half being genuine communications. In terms of higher recall efficiency, this method fared better than any other individual classifier or group of classifiers [6].

A detection tool was established by Abu-Nimeh and colleagues [7] to defend mobile platforms from assaults. To increase their predictive accuracy and remove the overhead of variable selection, the client-server distributed server uses Additive Regression Trees alongside the server with the help of automatic variable selection.

Using newly created features from these emails, Gansterer [8] suggested a filtering system that divides received emails into three categories: valid (solicited e-mail), spam, and phishing emails. To categorize received messages, the system includes a variety of classifiers. Between the three groups, a classification accuracy of 97% was attained, which is thought to be superior to the ternary classification problem being solved by a series of two binary classifiers.

III.METHODOLOGY

3.1 Data Collection and Preprocessing:

The success of any machine learning model relies on high-quality data. The data is from kaggle with some .mbox files with separate two files one is phishing mails [9] and legal mails [10]. Data preprocessing techniques, including email parsing, feature extraction, handling missing values, and data normalization.

3.2 Feature Extraction:

Extracting relevant features from emails is crucial for model training. This section will explore common features used in phishing email detection, such as sender information, URL analysis, email content analysis, and header information. Feature engineering plays a pivotal role in differentiating legitimate emails from phishing emails.

3.3 Classification Algorithms:

For this study the classification algorithms is used as follow:

3.3.1 Support Vector Machines:

Support Vector Machines are a powerful and versatile supervised learning algorithm used for both classification and regression tasks. SVM is particularly effective for binary classification, where the goal is to separate data points into two classes based on their features. However, SVM can also be extended to handle multi-class classification tasks.

3.3.2 Random Forest:

Random Forest is an ensemble learning method used for both classification and regression tasks. It constructs multiple decision trees during training and makes predictions by averaging or voting the results of individual trees.

3.3.3 Decision Tree

A decision tree is a popular machine learning algorithm used for both classification and regression tasks. It works by recursively splitting the dataset into subsets based on the values of input features, with the goal of creating a tree-like structure of decisions that leads to a final prediction or outcome. Each internal node of the tree represents a decision based on a particular feature, and each leaf node represents a predicted class or value.

SVM, Random Forest and Decision tree are valuable machine learning algorithms, each with its strengths and weaknesses. The choice between them depends on the specific characteristics of the dataset and the problem at hand. SVM is powerful for binary classification and can handle high-dimensional data, while Random Forest is well-suited for large datasets and provides good accuracy through ensemble methods and Decision tree is intuitive, interpretable, and can capture complex non-linear relationships in the data. The choice between these algorithms should be based on factors such as the nature of the data, the complexity of the problem, the interpretability required, and the trade-off between model performance and computational resources.

3.4 Model Training and Evaluation:

In this section, the steps taken for model training and testing for phishing detection project. The utilized Support Vector Machines (SVM), Random Forest algorithms and Decision Tree to build and evaluate our models. The dataset used for training and testing consists of labeled examples of phishing and legitimate emails. The preprocessed dataset into a training set (80% of the data) and a testing set (20% of the data).

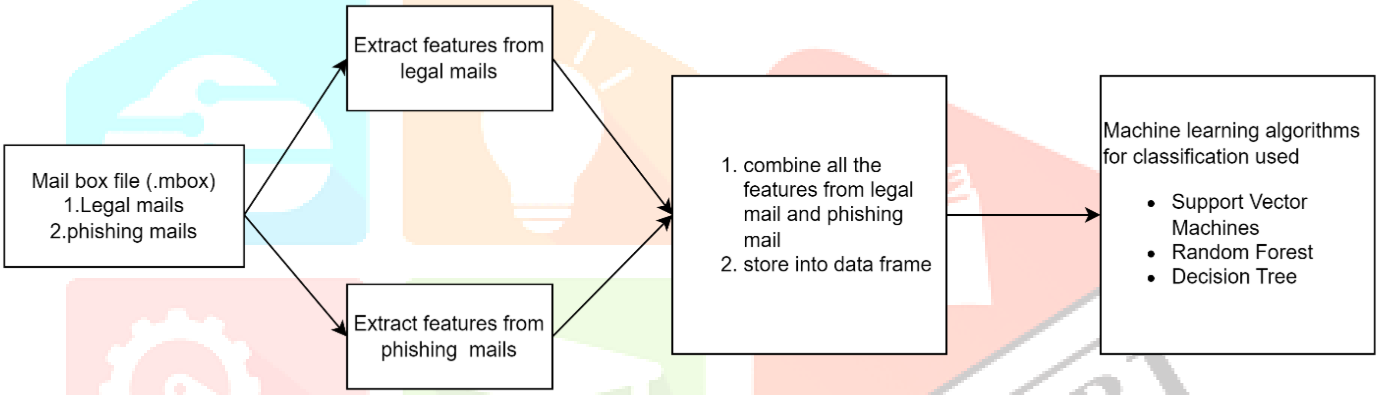
Mail box file (.mbox)
1.Legal mails
2.phishing mails
Extract features from legal mails
Extract features from phishing mails
1. combine all the features from legal mail and phishing mail
2. store into data frame
Machine learning algorithms for classification used
• Support Vector Machines
• Random Forest
• Decision Tree

Figure 1: Process flow for Phishing Email detection

3.4.1 Model Training:

Support Vector Machines (SVM):

- We used the scikit-learn library in Python to train an SVM classifier on the training set using the radial basis function (RBF) kernel.
- During training, we tuned the hyper parameters, such as the regularization parameter (C) and the kernel coefficient (gamma), using cross-validation to optimize the model's performance.
- The model was then trained on the entire training set using the optimal hyper parameters.

Random Forest:

- The scikit-learn library in Python to train a Random Forest classifier on the training set.
- Experimented with different numbers of decision trees (n_estimators) and maximum depth of trees (max_depth) to find the best configuration.
- The final Random Forest model was trained using the optimal hyper parameters.

Decision Tree:

- The scikit-learn library in Python to train a Decision Tree classifier on the training set.
- Explored various hyper parameters, including tree depth (max_depth) and the minimum number of samples required to split a node (min_samples_split), to discover the most suitable configuration.

- Employed the identified optimal hyper parameter values to train the final Decision Tree model.

3.4.2 Model Testing and Evaluation:

After training both models, we evaluated their performance on the testing set to assess their ability to detect phishing Emails accurately. The evaluation metrics used were precision, recall, F1-score, and accuracy.

	Accurac y	**Precisio n**	**Recal l**	**F1-score**
SVM	94.51	0.8	0.79	0.79
Random Forest	97.64	0.94	0.97	0.96
Decision Tree	96.27	0.94	0.96	0.95

Table1: Evaluation Metric for algorithms

IV.RESULTS

Based on the evaluation results, we observed that the Random Forest algorithm gave good performance and competition the SVM and Decision Tree algorithms in terms of accuracy and overall performance. It achieved a higher accuracy of 98%, whereas the Decision Tree and SVM achieved accuracies of 96% and 94% respectively. The Random Forest model also exhibited better precision, recall, and F1-score values compared to the other two algorithms. (Figure 2).

However, it is crucial to consider various factors such as computational efficiency, interpretability, and scalability when selecting a model for real-world applications. Depending on the specific requirements and limitations of the project, one algorithm may be more suitable than the others.

In the context of our phishing detection project, the effectiveness of machine learning algorithms in identifying potentially malicious emails was evident. The Random Forest model, in particular, demonstrated superior performance in accurately distinguishing between phishing and legitimate emails. Nevertheless, it's important to emphasize that continuous monitoring and updates to the models are imperative to stay ahead of the ever-evolving landscape of phishing attacks.

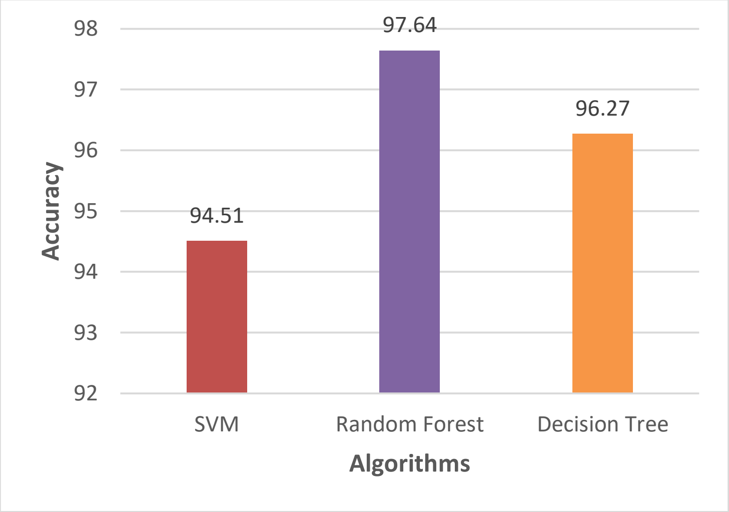

Figure 2: Accuracy plotting for all three algorithms

V. CONCLUSION

In this work, we developed phishing mails detection using Machine Learning techniques. Our main goal was to construct and assess models with the ability to differentiate between legitimate emails and phishing emails. This endeavor aimed to fortify internet security and shield users from potential cyber hazards. To achieve this, we employed three robust algorithms: Support Vector Machines (SVM), Random Forest, and Decision Tree. The initial phase encompassed data preprocessing, involving meticulous cleansing of the dataset, extraction of pertinent features from .mbox files, and appropriate scaling of these features. These preliminary steps were critical to ensure data quality and facilitate effective learning by the models. Subsequently, we opted for SVM, Random Forest, and Decision Tree algorithms due to their proven aptitude in binary classification tasks, as well as their capacity to handle intricate feature interactions. This result suggests that the Random Forest algorithm adeptly balanced precision and recall, rendering it a robust choice for efficient phishing detection. However, it's important to note that the optimal model selection hinges on specific application requirements. Future endeavors could explore additional features and innovative feature engineering techniques to amplify the models' discriminative capabilities.

REFERENCES

[1] Marforio, C., Masti, R. J., Soriente, C., Kostiainen, K., and Capkun, S. (2015). “Personalized security indicators to detect application phishing attacks in mobile platforms”. Available at: http://arxiv.org/abs/1502.06824.

[2] Ponnurangam Kumaraguru, Yong Rhee, Alessandro Acquisti, Lorrie Faith Cranor, Jason Hong, Elizabeth Nunge, "Protecting people from phishing: the design and evaluation of an embedded training email system", Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, April 2007 Pages 905–914, https://doi.org/10.1145/1240624.1240760

[3] Parmar, B. (2012). Protecting against spear-phishing. Computer Fraud & Security, 8-11. Ramanathan, V., & Wechsler, H. (2012). phishGILLNET—phishing detection methodology using probabilistic latent semantic analysis, AdaBoost, and co-training. EURASIP Journal on Information Security, 1-22.

[4] Cao, Y., Han, W., & Le, Y. (2008). Anti-phishing based on automated individual white-list. In Proceedings of the 4th ACM workshop on Digital identity management (pp. 51-60).

[5] Kumaraguru, P., Sheng, S., Acquisti, A., Cranor, L. F., & Hong, J. (2010). Teaching Johnny not to fall for phish. ACM Transactions on Internet Technology (TOIT), 10(2), 7.

[6] Gansterer, W. N., & Pölz, D. (2009). E-mail classification for phishing defense.In Advances in Information Retrieval (pp. 449-460). Springer Berlin Heidelberg.

[7] Toolan, F., & Carthy, J. (2009, September). Phishing detection using classifier ensembles. In eCrime Researchers Summit, 2009. eCRIME'09.(pp. 1-9). IEEE.

[8] Abu-Nimeh, S., Nappa, D., Wang, X., & Nair, S. (2007, October). A comparison of machine learning techniques for phishing detection. InProceedings of the anti- phishing working groups 2nd annual eCrime researchers summit (pp. 60-69). ACM.

[9] https://www.kaggle.com/code/riyapatel1697/phishing-email-detection-ai-ml/input?select=emails-phishing-nazario.mbox

[10] https://www.kaggle.com/code/riyapatel1697/phishing-email-detection-ai-ml/input?select=emails-enron-legal-mails.mbox