**A SUPERVISED LEARNING APPROACH TO PHISHING DETECTION: IMPLEMENTATION AND EVALUATION**

**Abstract**

Phishing attacks pose a significant threat to cybersecurity, with increasingly sophisticated tactics being employed to deceive users. This paper presents a supervised learning approach to detect phishing emails using a Random Forest Classifier. We describe the implementation of a Phishing Detection Dashboard application and evaluate its performance on real-world email data. Our findings demonstrate the effectiveness of this approach, with significant potential for enhancing email security and mitigating security vulnerabilities.

### Keywords

Phishing Detection, Supervised Learning, Random Forest, Machine Learning, Cybersecurity

### 1. Introduction

Phishing attacks have become one of the most prevalent forms of cybercrime, targeting individuals and organizations by masquerading as trustworthy entities in electronic communications (Gupta et al., 2017). These attacks aim to steal sensitive information such as usernames, passwords, and credit card details, social security details by tricking recipients into revealing personal information. The primary objective of this research is to develop an automated system capable of detecting phishing emails to reduce the risk of security breaches. This study explores the use of supervised learning, particularly Random Forest Classifier, to classify emails as phishing or non-phishing, and presents an application to visualize and manage these detections.

### 2. Background

Phishing detection has traditionally relied on heuristic-based approaches, which, while useful, can often be circumvented by more sophisticated phishing techniques. Recent advancements in machine learning offer promising alternatives by leveraging patterns within the data to identify phishing attempts (Bergholz et al., 2010). Machine learning models can learn from labeled data and identify complex patterns that are not easily captured by rule-based systems. This study aims to harness these advancements by employing a Random Forest Classifier due to its robustness and interpretability. The choice of a Random Forest model is motivated by its ability to handle high-dimensional data and its resilience to over fitting.

### 3. Methodology

#### 3.1 Data Collection

Emails were fetched using the IMAP protocol from a secure server. The dataset includes both phishing and non-phishing emails, manually labeled for supervised learning purposes. The use of real-world email data ensures that the model is trained and evaluated on practical and diverse samples, capturing a wide range of phishing techniques.

#### 3.2 Preprocessing

Emails were preprocessed to extract relevant textual features. This included cleaning the email content, removing HTML tags, and normalizing text. The preprocessing step is crucial to ensure that the textual data is in a suitable format for feature extraction and model training. The preprocessing pipeline includes tokenization, stopword removal, and stemming to reduce the text to its base form.

#### 3.3 Feature Extraction

We utilized the TfidfVectorizer from scikit-learn to convert the email text into numerical feature vectors, capturing the importance of each word relative to the entire dataset. TF-IDF (Term Frequency-Inverse Document Frequency) is a statistical measure used to evaluate the importance of a word in a document relative to a collection of documents. This method helps in highlighting the significant words that are indicative of phishing content.

#### 3.4 Model Selection

A Random Forest Classifier was chosen for its balance between performance and simplicity. The model was trained on a subset of the dataset and evaluated on the remaining data. Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes for classification tasks. Its ability to handle large datasets with higher dimensionality makes it suitable for phishing detection.

### 4. Implementation

#### 4.1 System Architecture

The system comprises a Flask backend for email fetching and processing, and a React frontend for visualization. The backend handles data preprocessing, model training, and prediction, while the frontend provides an interactive dashboard for users to view and manage detected phishing emails.

### References

* Bergholz, A., De Beer, J., Glahn, S., Moens, M., Paaß, G. and Strobel, S., 2010. New filtering approaches for phishing email. Journal of Computer Security, 18(1), pp.7-35.
* Gupta, B.B., Arachchilage, N.A.G. and Psannis, K.E., 2017. Defending against phishing attacks: taxonomy of methods, current issues and future directions. Telecommunication Systems, 67, pp.247-267.