Mini Project Report on

Cyber Threat analysis - Phishing emails detection

Submitted in partial fulfillment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE & ENGINEERING

Submitted by:

Adit Sharma **University Roll No: 2018634**

Under the Mentorship of

Siddhant Thapliyal

Assistant Professor

Department of Computer Science and Engineering
Graphic Era (Deemed to be University)
Dehradun, Uttarakhand

July-2024

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the project report entitled **"Cyber Threat analysis - Phishing emails detection"** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineering of the Graphic Era (Deemed to be University), Dehradun shall be carried out by the under the mentorship of **Siddhant Thapliyal, Assistant Professor**, Department of Computer Science and Engineering, Graphic Era (Deemed to be University), Dehradun.

Name : Adit Sharma **University Roll no: 2018634**

Table of Contents

Chapter 1

Introduction

In the following sections, a brief introduction and the problem statement for the work has been included.

1.1 Introduction

In the ever-evolving digital landscape, cyber threats pose a constant challenge. Cyber threat analysis is the cornerstone of a proactive cybersecurity strategy. It's a meticulous process of identifying, assessing, and understanding potential vulnerabilities and threats that could exploit them. Imagine it as a detective story – meticulously gathering clues (vulnerabilities) to predict the criminal's (attacker's) next move. This analysis empowers us to take preventative measures, like patching security holes or implementing stronger access controls, before attackers can exploit them. By anticipating these threats, cyber threat analysis helps organizations become cyber-resilient, safeguarding valuable data and critical infrastructure.

Phishing emails remain a persistent and costly cyber threat, tricking unsuspecting users into revealing sensitive information or downloading malware. Our project tackles this issue by developing a system to automatically detect phishing emails, safeguarding individuals and organizations from these malicious attacks.

1.2 Objectives

Our project has two main objectives in the fight against phishing emails:

1. **Enhance Detection Accuracy:** We aim to develop a robust system that surpasses existing detection methods in accurately identifying phishing emails. This includes minimizing false positives (legitimate emails flagged as phishing) and false negatives (phishing emails slipping through the cracks).
2. **Reduce User Reliance on Manual Detection:** By creating an automated detection system, we aim to empower users and organizations to rely less on manual identification of phishing attempts. This reduces human error and streamlines the process of filtering out malicious emails, ultimately improving overall cybersecurity posture.

1.3 Scope

Our project tackles the persistent threat of phishing emails by developing a system for automated detection. We aim to empower individual and SMB users with a robust tool that utilizes machine learning to identify and flag malicious emails.

While achieving high accuracy is a priority, we acknowledge limitations inherent to any detection system. These include the vastness of email data, the constant evolution of phishing tactics, and the need for user vigilance alongside automated filtering. This initial stage focuses on analyzing email content for phishing indicators, laying the foundation for future enhancements.

Chapter 2

Literature Survey

2.1 Introduction:

Phishing emails remain a prevalent and costly cyber threat, causing significant financial losses and data breaches. To combat this challenge, researchers have explored various methods for phishing email detection. This literature survey delves into existing research on this crucial area of cybersecurity.

2.2 Existing Methods and Models:

- **Rule-Based Filtering:** This approach relies on predefined rules that identify common characteristics of phishing emails, such as suspicious sender addresses, urgency tactics, or malicious URLs. While simpler to implement, rule-based systems may struggle to adapt to evolving phishing tactics.
- **Hybrid Approaches:** Some studies explore combining machine learning and rule-based filtering, leveraging the strengths of both methods. ML can handle complex patterns, while rules can target specific phishing tactics.

2.3 Identified Gaps:

Despite existing research, there's still room for improvement in automated phishing email detection. Some key gaps identified in the literature review include:

- **Evolving Phishing Techniques:** New tactics and email content variations constantly emerge, requiring detection systems to adapt and learn from new data.
- **Data Imbalance:** Training datasets may be imbalanced, with a vast amount of legitimate emails compared to a smaller number of phishing samples. This can hinder model performance.

- **Explainability of ML Models:** While ML models can achieve high accuracy, understanding their decision-making processes remains a challenge. This can limit user trust in automated detection systems.

These gaps highlight the need for continuous research and development in this area.

By understanding the current landscape of phishing email detection methods and their limitations, we can position our project to address these challenges and contribute to a more robust defense against phishing attacks.

Chapter 3

Methodology

3.1 Data Collection

The dataset used for this project was sourced from Kaggle, a platform providing a variety of datasets for machine learning and data science projects. The phishing email dataset includes features such as the content of the email, headers, sender and receiver information, and a label indicating whether an email is phishing or legitimate. The dataset comprises almost 18000 emails, providing a comprehensive base for training and evaluating our model.

3.2 Data Preprocessing

Text Processing: The text content of emails underwent several preprocessing steps:

- Conversion to lowercase to maintain consistency.
- Removal of punctuation, special characters, and common stop words to reduce noise.
- Tokenization to split the text into individual words or tokens.

Feature Extraction: We employed Term Frequency-Inverse Document Frequency (TF-IDF) to convert the text data into numerical form. This technique helps to highlight important words in each email by considering their frequency and distribution.

Balancing: SMOTE was used to resample the values in order to balance the minority class.

Splitting: The processed data was split into training and testing sets using an 80:20 ratio. This ensures that we have a sufficient amount of data for training the model while retaining a separate set for unbiased evaluation.

3.3 Model Development

Model Selection: The Random Forest algorithm was selected due to its robustness and effectiveness in handling a mix of numerical and categorical features. It operates by constructing multiple decision trees during training and outputting the class that is the mode of the classes (classification) of the individual trees.

3.4 Training the Model

Training Process: The Random Forest model was trained using the training dataset. This involved fitting multiple decision trees on various sub-samples of the dataset and averaging their predictions to improve accuracy and control overfitting. The model was trained using the following steps:

1. Bootstrapping the dataset to create multiple sub-samples.
2. Training individual decision trees on these sub-samples.
3. Aggregating the predictions from all trees to make the final prediction.

Validation: Cross-validation was performed to monitor the model's performance and tune the hyperparameters. This process helped in selecting the best set of hyperparameters to ensure the model's robustness.

3.5 Model Evaluation

Metrics: The model's performance was evaluated on accuracy and an accuracy of 97% was achieved.

Testing: The trained Random Forest model was tested using the testing dataset, which was not seen by the model during training. This evaluation provided an unbiased assessment of the model's generalization capability.

Comparison: To gauge the effectiveness of the Random Forest model, its performance was compared with that of other machine learning models such as Decision Trees and Naive Bayes. This comparison highlighted the advantages of using Random Forests for phishing email detection.

Chapter 4

Result and Discussion

4.1 Model Performance

The performance of the Random Forest model was evaluated using the testing dataset and a high accuracy of 97% was achieved.

4.2 Confusion Matrix

The confusion matrix provides a detailed breakdown of the model's performance in terms of true positives, true negatives, false positives, and false negatives. The following table summarizes the confusion matrix:

	Predicted: Phishing	**Predicted: Legitimate**
Actual: Phishing	[True Positives] 2118	[False Negatives] 96
Actual: Legitimate	[False Positives] 37	[True Negatives] 2278

The confusion matrix helps to understand the distribution of correctly and incorrectly classified instances, providing insights into the areas where the model performs well and where improvements may be needed.

3. Comparison with Other Models

The performance of the Random Forest model was compared with other traditional machine learning models such as Logistic Regression and Support Vector Machines (SVMs). The following table summarizes the comparison:

Model	Accuracy
Random Forest	97.06
Decision Trees	92.24
Naive Bayes	95.71

The Random Forest model outperformed both Decision Trees and Naive Bayes in terms of all performance metrics, highlighting its superiority for the task of phishing email detection.

Chapter 5

Conclusion and Future Work

Moving forward, there are several avenues for enhancing the effectiveness and scope of phishing email detection systems:

1. **Improved Dataset**: Enhance dataset diversity with more emails from various sources, languages, and time periods. Real-time data integration and detailed annotations can further enrich the dataset's quality.
2. **Advanced Techniques**: Explore advanced machine learning approaches such as Recurrent Neural Networks (RNNs) and Transformer models (e.g., BERT) to capture complex email content dependencies and improve detection accuracy.
3. **Enhanced Feature Extraction**: Utilize semantic analysis, behavioral features, and email metadata to extract deeper insights from email content and improve phishing detection capabilities.
4. **Model Evaluation and Deployment**: Implement real-time evaluation, adaptive learning strategies, and user feedback integration to continuously refine and optimize the detection models in real-world environments.

These enhancements aim to develop more robust and adaptive phishing detection systems, safeguarding users against evolving cyber threats effectively.