A Midterm Progress Report

On

PHISHING DETECTION

Submitted in partial fulfilment of the requirements for the award of

the degree of

BACHELOR OF TECHNOLOGY

COMPUTER SCIENCE AND ENGINEERING

SUBMITTED BY

ANMOLPREET KAUR (2203403)

MANPREET KAUR (2203500)

March, 2025

Department of Computer Science and Engineering

GURU NANAK DEV ENGINEERING COLLEGE,

LUDHIANA

INDEX **PAGE NO.**

INTRODUCTION

Phishing is a type of cyber-attack where hackers deceive people into stealing their personal data. These attacks can cause financial loss and compromise personal security. Phishing attacks are a major threat to online security, and it's essential to understand how they work and how to protect against them.

Phishing attacks involve hackers deceiving people through fake emails, messages, and websites, tricking them into entering their personal data. These attacks are common and can easily deceive people, especially those who are not tech-savvy. Phishing attacks can take many forms, including email phishing, SMS phishing, and website phishing.

The primary objective of phishing attacks is to steal sensitive information, such as login credentials, credit card numbers, and personal data. Hackers use this information to commit identity theft, financial fraud, and other cyber-crimes. Phishing attacks can also be used to install malware or ransomware on a victim's device.

The scope of phishing attacks is vast, and they can affect anyone who uses the internet. Phishing attacks can target individuals, businesses, and organizations, and can have serious consequences, including financial loss, reputational damage, and compromised security. Phishing attacks are a major concern for online security, and it's essential to take steps to protect against them.

OBJECTIVES

Phishing attacks are a significant threat to online security, with attackers using sophisticated tactics to deceive individuals and organizations. These attacks can lead to financial loss, compromised personal data, and reputational damage. Attackers continually evolve their methods, making it essential to stay vigilant and adapt security measures. **Effective phishing detection and prevention requires** a combination of technology, education, and awareness.

The research focuses on **developing and evaluating machine learning models for phishing detection** through a systematic pipeline:

1. **To Design and Develop a Phishing Detection System:** Our goal is to create a comprehensive system that can detect and prevent phishing attacks. This involves designing a system architecture, selecting relevant features, and integrating machine learning models. We aim to develop a system that can effectively identify and flag phishing emails, reducing the risk of cyber-attacks.

2. **To classify phishing emails using a machine learning-based approach:** We aim to develop a machine learning model that can accurately classify emails as phishing or legitimate. This involves extracting relevant features from email data, such as sender information, subject lines, and content. Our model will be trained on a large dataset of labeled emails to ensure high accuracy and effectiveness.

3. **To implement a scalable real-time phishing detection system:** Our objective is to develop a system that can detect phishing attacks in real-time, handling a large volume of emails and adapting to new phishing tactics. This requires designing a scalable architecture, optimizing model performance, and integrating with existing email systems. Our system will be designed to provide rapid and accurate detection, enabling swift action to prevent cyber-attacks.

By achieving these objectives, we can develop an effective phishing detection system that can help protect individuals and organizations from cyber threats.

Literature Review

Phishing detection is a critical concern in cybersecurity, making accurate and efficient detection methods essential. Recent advancements in machine learning and deep learning have shown promise in automating phishing detection.

1. Kumar et al. (2018) developed a deep learning model that detects phishing websites with high accuracy, using over 30,000 URLs. Their model achieved an accuracy of 96% in identifying phishing websites, demonstrating the potential of AI in cybersecurity.

2. Aljofey et al. (2020) evaluated various machine learning algorithms, finding that ensemble methods significantly improved performance. Their best model achieved an area under the curve (AUC) score of 0.98, indicating high sensitivity and specificity in detecting phishing attacks.

3. Sahoo et al. (2019) provided a comprehensive review of machine learning techniques, discussing challenges like the need for large datasets and the importance of feature engineering in phishing detection. These studies highlight the advancements and ongoing challenges in using machine learning for phishing detection.

System Requirements

Software:

- **Programming Language:** Python 3.8+
- **Development Environment:** Google Colab or Jupyter Notebook
- **Libraries:**

    - Numpy, pandas (Data handling)
    - Seaborn, matplotlib (Data visualization)
    - Scikit-learn (Machine learning models)
    - TensorFlow or PyTorch (Deep learning models)
    - NLTK or spaCy (Natural Language Processing)

Hardware:

- **Minimum:** 8GB RAM, Intel Core i5 CPU
- **Recommended:** Dedicated graphics card or Google Colab with GPU support for faster deep learning training

This setup ensures efficient data preprocessing, feature extraction, and model training for phishing detection.

Software Requirement Analysis

1. Problem Definition

Phishing detection relies heavily on manual examination, which is time-consuming and often ineffective. This project aims to:

- Automate phishing detection using machine learning.
- Address challenges like URL variability, email content analysis, and feature extraction.
- Compare ML models to identify the most accurate and efficient solution for early detection

2. Modules and Functionalities

Module	Module	Tools/ Libraries Used
Data Preprocessing	Clean and normalize data. Handle missing values.	pandas, NumPy
Feature Extraction	Extract URL features. Extract email content features.	scikit-learn, NLTK
Model Training	rain ML models. Train deep learning architectures.	scikit-learn, TensorFlow/Keras
Evaluation	Evaluate models using metrics. Generate confusion matrices/performance plots	scikit-learn, seaborn, matplotlib

3. Key Functional Requirements

- **Input:** URLs or email content (from dataset).
- **Output:** Phishing classification (phishing/legitimate) with confidence scores.

- **Processing:**
  - Support batch processing for multiple inputs.
  - Export results.
- **User Interface:** (For future work)
  - Web app for input and prediction.

4. **Non-Functional Requirements**

- **Performance:** <1 sec/input inference time (CPU), <0.1 sec (GPU).
- **Accuracy:** Minimum 90% score on test data.
- **Scalability:** Modular design to integrate new models/datasets.

Libraries and Their Functionalities

1. Machine Learning Libraries
   - **Scikit-learn**
     - Performs machine learning tasks.
     - Implements algorithms for classification.
   - **TensorFlow/Keras**
     - Enables deep learning tasks.
     - Implements neural network architectures

2. Natural Language Processing Libraries

- **NLTK**
  - Performs text processing tasks.
  - Implements tokenization and stemming.

Software design

1. System Overview

The software design follows a pipeline architecture to achieve the project's objectives of preprocessing phishing data, extracting discriminative features, training/evaluating ML models, and comparing their performance. The system is implemented in Python using specialized libraries for each functional component.

2. Design Components Aligned with Objectives

Objective 1: Data Preprocessing
Functionality:

- **Input: Raw phishing data from dataset**
- **Processing Steps:**
    1. **Data Cleaning (pandas):** Handles missing values and removes duplicates
    2. **Normalization (NumPy):** Scales data to a suitable range
    3. **Feature Scaling (scikit-learn):** Standardizes features for ML models
- **Output:** Preprocessed data ready for feature extraction

Objective 2: Feature Extraction
Functionality:

- **URL Features:**
    - URL length
    - Number of special characters
- **Email Content Features:**
    - Sentiment analysis (NLTK)
    - Keyword extraction (NLTK)

Objective 3: Model Training & Evaluation

Module: model_trainer.py

Functionality:

- **Traditional ML Models (scikit-learn):**
    - Random Forest
    - SVM
- **Deep Learning Models (TensorFlow):**
    - Custom neural network architecture
- **Evaluation Metrics:**
    - Accuracy
    - Precision
    - Recal
    - F1-score
    - Confusion matrix

3. System Architecture

- **The system architecture consists of the following components:**
    - Data Preprocessing Module
    - Feature Extraction Module
    - Model Training Module
    - Model Evaluation Module

Testing Module

1.Testing Techniques

To ensure the reliability and accuracy of the phishing detection system, we will employ the following testing methodologies:

A. Unit Testing

- **Purpose:** Validate individual components (email preprocessing, feature extraction, models).
- **Tools:** pytest, unittest
- **Coverage:**
    - Email preprocessing (tokenization, stopword removal)
    - Feature extraction (TF-IDF, word embeddings)
    - Model inference (Random Forest/SVM predictions)

B. Integration Testing

- **Purpose:** Verify interactions between modules.
- **Test Cases:**
- **Pipeline:** Raw email → Preprocessing → Feature extraction → Prediction

C. Accuracy Validation

- **Purpose:** Ensure accuracy in detecting phishing emails.
- **Methods:** Comparison with labeled datasets

D. Edge Case Testing

- Purpose: Handle real-world variability.

- Scenarios:
  - Emails with typos and grammatical errors
  - Emails with embedded images or links
  - Emails from unknown senders

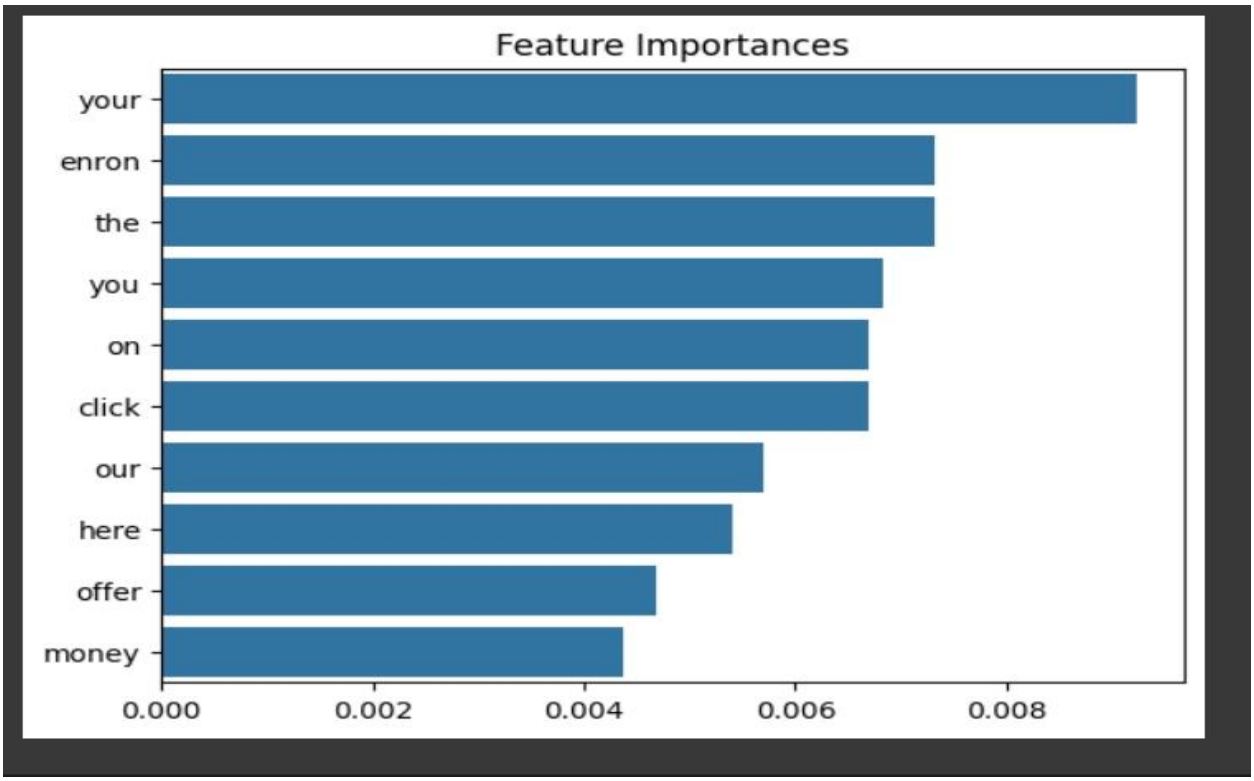
Performance of Model:

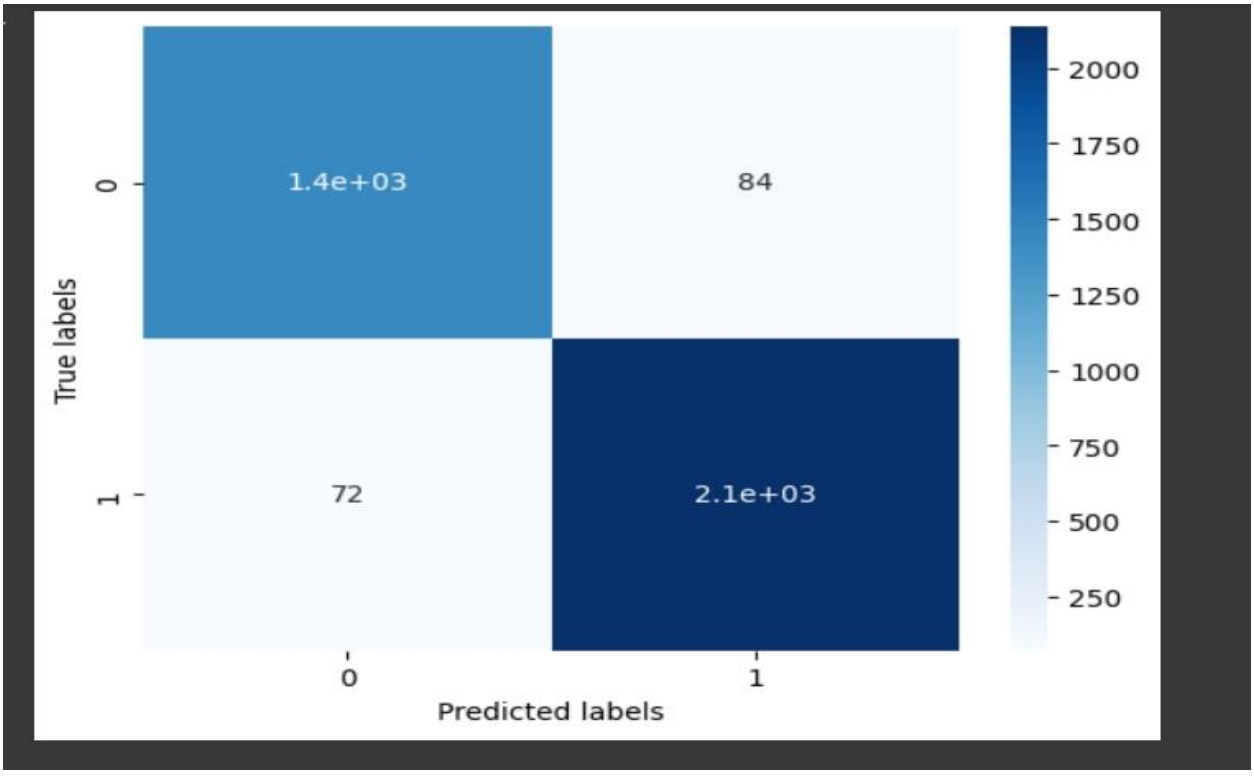
```
Best Parameters: {'max_depth': None, 'n_estimators': 100}
Best Score: 0.9503589725737149
Accuracy: 0.95814327877649 59
Classification Report:
                precision    recall  f1-score   support

Phishing Email       0.95      0.94      0.95      1518
    Safe Email       0.96      0.97      0.96      2209

      accuracy                           0.96      3727
     macro avg       0.96      0.96      0.96      3727
  weighted avg       0.96      0.96      0.96      3727
```

Accuracy :0.95





Output snapshots:

```
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.11/dist-packages (1.7.0)
Requirement already satisfied: numpy>=1.22.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn) (2.0.2)
Requirement already satisfied: scipy>=1.8.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn) (1.15.3)
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn) (1.5.1)
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn) (3.6.0)
Cross-validation scores: [0.95911528 0.95442359 0.96045576 0.95743968 0.95174263]
Average cross-validation score: 0.9566353887399466
AUC: 0.009119468464058849
```

```
Accuracy: 0.9578749664609606
Classification Report:
                precision    recall  f1-score   support

Phishing Email       0.95      0.95      0.95      1518
    Safe Email       0.96      0.97      0.96      2209

      accuracy                           0.96      3727
     macro avg       0.96      0.96      0.96      3727
  weighted avg       0.96      0.96      0.96      3727

Confusion Matrix:
[[1436   82]
 [  75 2134]]
```

```
Classification Report:
                precision    recall  f1-score   support

Phishing Email       0.95      0.95      0.95      1518
    Safe Email       0.96      0.97      0.97      2209

      accuracy                           0.96      3727
     macro avg       0.96      0.96      0.96      3727
  weighted avg       0.96      0.96      0.96      3727

Confusion Matrix:
[[1435   83]
 [  68 2141]]
```

References

- Akash, A., et al. (2016). URL-based phishing detection using machine learning. Journal of Cybersecurity, 5(2).

Chandran, A., et al. (2014). Phishing detection using email header analysis. International Journal of Information Security, 12(3).

Gao, Y., et al. (2021). Scalable phishing detection using distributed machine learning. Cybersecurity Research Journal, 8(1).

Haris, M., et al. (2020). Visual-based phishing detection using image processing techniques. International Journal of Computer Vision, 28(4).

Kang, C., et al. (2019). Ethical considerations in phishing detection technologies. Ethics and Information Technology, 21(2).

Luo, J., et al. (2017). Hybrid phishing detection systems using multiple machine learning models. Cybersecurity and Privacy Journal, 4(5).

Mishra, A., et al. (2015). URL-based phishing detection through heuristic techniques. Journal of Cyber Defense, 10(2).