# A Phishing Detection System Leveraging DistilBERT For Phishing Mail Classification

Thando Lesego Dlamini

May 2025

# Contents

# 1. Research Overview

## Phishing Detection Using AI

AI Phishing Detection refers to the application of machine learning and natural language processing algorithms to identify and respond to phishing attacks in their early stages. These systems analyse multiple email characteristics to recognize social engineering attempts before they reach users, preventing possible data breaches, protecting sensitive data, and reducing compliance risk [1].

## How AI can be used to detect phishing attacks

Artificial Intelligence (AI) and machine learning (ML) models can be trained to analyse the content (text) of an email or the URLs that it points to. By leveraging machine learning algorithms, AI can be used to identify suspicious patterns in emails and websites that traditional methods miss. Using AI allows us to analyse multiple data points simultaneously including: email content, sender reputation, URL structures and user behavior, to detect sophisticated phishing attempts. In contrast to conventional phishing detection methods, AI models continuously learn from new threat patterns, adapting to evolving tactics without requiring manual updates. Additionally, AI can detect previously unknown attack variations by identifying subtle anomalies that signature-based approaches would miss [2–4].

# Requirements Specification

This section outlines the functional requirements for the AI-Powered Phishing Email Detection System.

## Functional Requirements

1. **Email Input Processing**
    - The system must accept email content through file upload.
    - The system must support common .eml email formats.
    - The system must extract and parse email headers, body content, and embedded URLs.
2. **AI-Based Detection**
    - The system must implement a machine learning model to classify emails as phishing or legitimate.
    - The system must achieve a minimum detection accuracy of 80% on the test dataset.
    - The system must analyse multiple data points including text content, sender information, URL patterns, and metadata.
    - The system must generate a confidence score (0-100%) for each classification.
3. **Explainable AI Features**
    - The system must provide explanations for its classification decisions.
    - The system must highlight suspicious elements within emails (phrases, URLs, sender details).
4. **User Interface**
    - The system must be user friendly and intuitive.
    - The system must display classification results.
    - The system must display the confidence score.

# UML Design Diagrams

This section presents the formal UML design diagrams for the AI-Powered Phishing Email Detection System. These diagrams illustrate the interactions, and workflows of the system.

## System Architecture

### Class Diagram

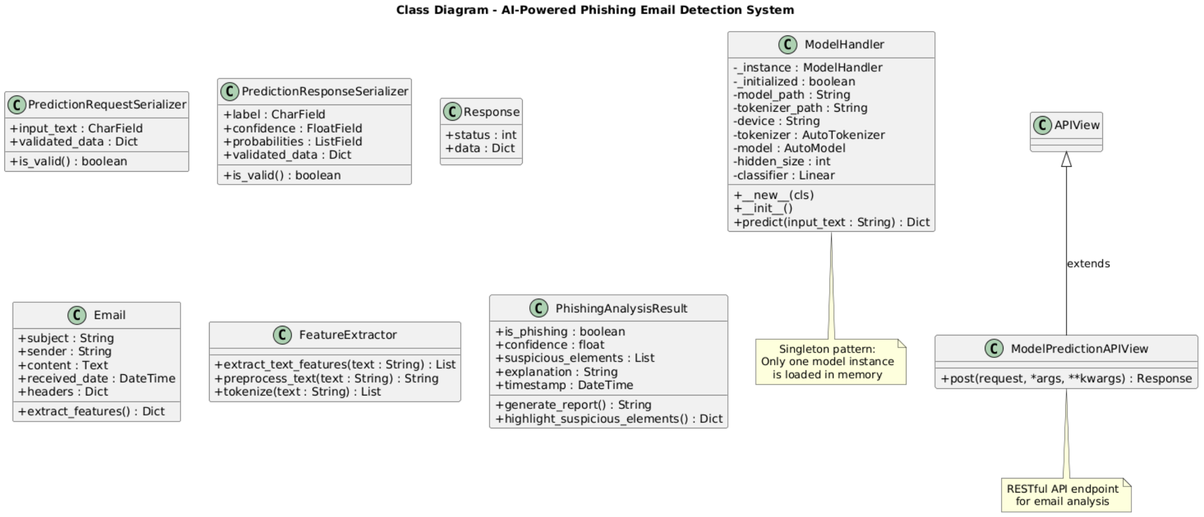

Figure 1: Class Diagram of the Phishing Detection System

The class diagram (Figure 1) shows the key classes and their relationships within the system:

- **Email** class represents the email data submitted for analysis, containing properties like subject, sender, content, received date, and headers. Provides methods to extract features from the email content and metadata.
- **FeatureExtractor** class handles preprocessing of email text and extracting relevant features for analysis. Provides methods for text preprocessing, tokenization, and feature extraction from the input text.
- **ModelHandler** class manages the machine learning model used for phishing detection. Implements a singleton pattern ensuring only one model instance is loaded in memory. Handles model initialization and prediction operations.
- **PredictionRequestSerializer** class handles the serialization of input text and validated data for the prediction request, with validation capabilities to ensure data integrity.
- **PredictionResponseSerializer** class manages the serialization of prediction results, including labels, confidence scores, probabilities, and validated data with validation capabilities.
- **Response** class is a simple data structure that contains the status code and response data for API communications.
- **PhishingAnalysisResult** class stores the results of phishing detection analysis, including boolean phishing status, confidence score, list of suspicious elements, explanations, and timestamp. Provides methods to generate reports and highlight suspicious elements.
- **APIView** class is a base class for API endpoints in the system.
- **ModelPredictionAPIView** class extends the APIView class to provide a RESTful API endpoint specifically for email analysis, handling post requests with arguments to process email data.

## System Workflows

### Sequence Diagram

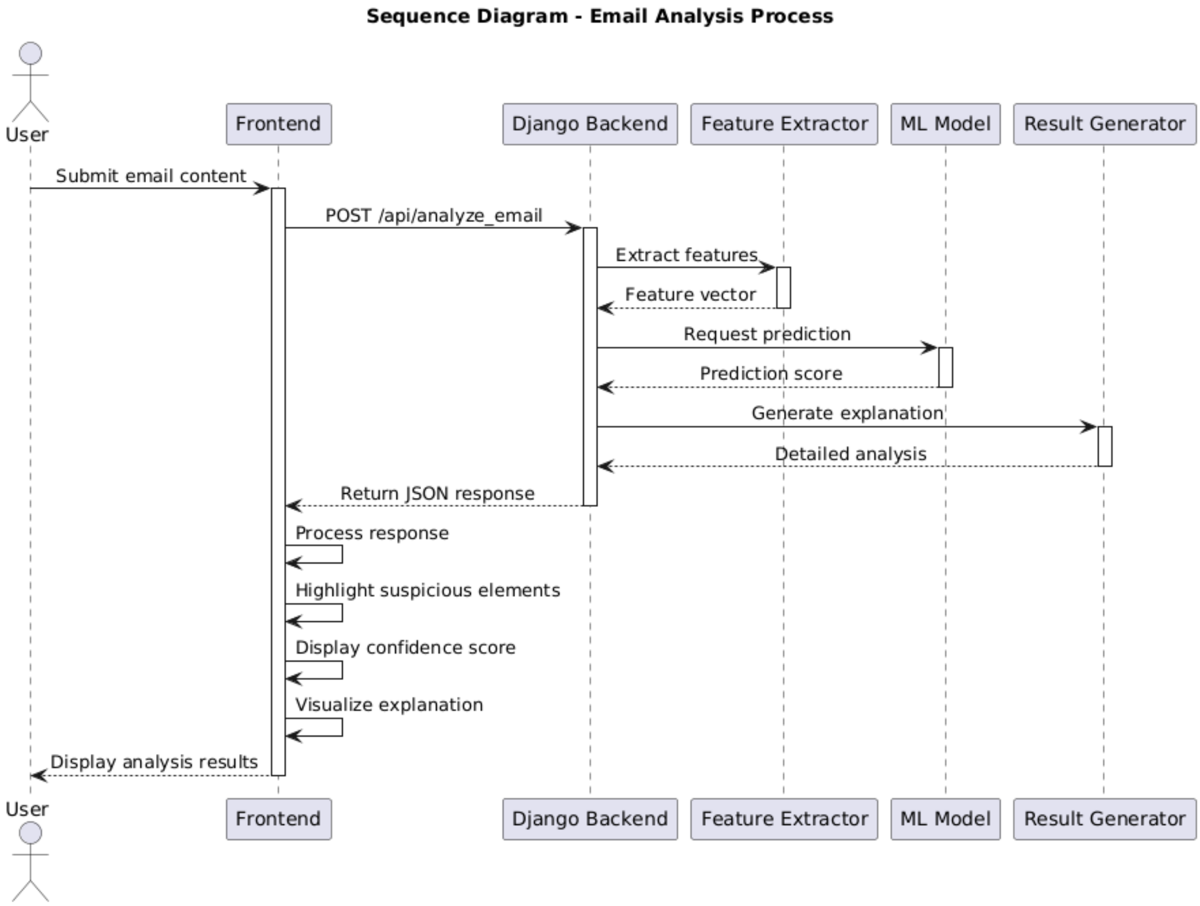

Figure 2: Sequence Diagram for Email Analysis Process

The sequence diagram (Figure 2) illustrates the interactions between system components during the email analysis process:

(a) The user submits an email through the frontend.

(b) The frontend sends the email content to the Django backend.

(c) The backend extracts features from the email.

(d) The ML model analyses the features and makes a prediction.

(e) The system generates explanations for the prediction.

(f) The results are returned to the frontend and displayed to the user.

### Activity Diagram

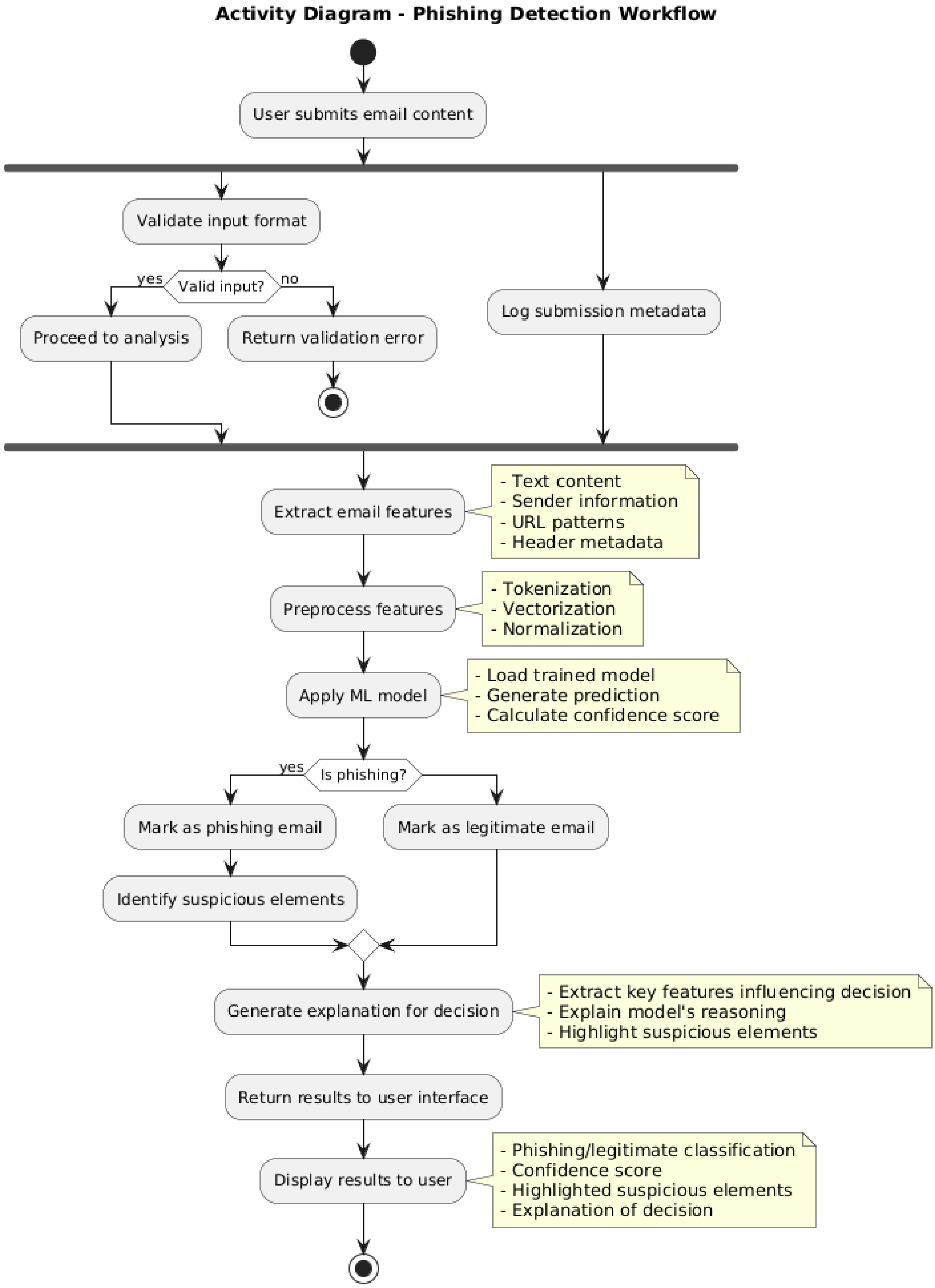

Figure 3: Activity Diagram for Phishing Detection Workflow

The activity diagram (Figure 3) shows the complete workflow of the phishing detection process:

(a) The email submission and initial validation

(b) The feature extraction and preprocessing

(c) The model prediction and confidence scoring

(d) The explanation generation for AI decisions

(e) The result presentation and feedback collection

# Model Selection Process

For this project, **DistilBERT** was selected as the foundation for our phishing email detection system.

## Evaluation Metrics

The DistilBERT model was fine-tuned on a dataset containing labeled phishing and legitimate email samples and evaluated performance using several key metrics:

- **Accuracy** – The proportion of correctly classified emails across both categories.
- **Precision** and **Recall** – Precision quantifies the reliability of positive phishing classifications, while recall measures the model's ability to identify all actual phishing emails.
- **F1-Score** – A harmonic mean of precision and recall, providing a single balanced performance metric.
- **Computational Efficiency** – Evaluation of model size and resource requirements to ensure practical deployment feasibility.

## Model Section Justification

**DistilBERT** was selected on several the following advantages:

- Exceptional performance in natural language understanding tasks, particularly for analysing email context and intent.
- Exceptional accuracy and F1-score metrics when differentiating between malicious and legitimate email content.
- Availability of pre-trained weights on extensive text corpora, minimizing the need for training from the ground up.

# 2. Testing Report

## 2.1 Model Evaluation Results

The model was trained and evaluated over 4 epochs. Below are the detailed performance metrics:

Table 1: Training and Validation Metrics by Epoch

| Epoch | Training Loss | Validation Loss | Accuracy | Precision | Recall | FPR |
|---|---|---|---|---|---|---|
| 1 | 0.2447 | 0.2435 | 0.9101 | 0.9496 | 0.9036 | 0.0793 |
| 2 | 0.1999 | 0.2736 | 0.9145 | 0.9377 | 0.9241 | 0.1015 |
| 3 | 0.1963 | 0.2469 | 0.9215 | 0.9499 | 0.9227 | 0.0805 |
| 4 | 0.1550 | 0.3038 | 0.9259 | 0.9342 | 0.9478 | 0.1102 |

## 2.2 Final Evaluation Metrics

The model achieved the following performance metrics after the final epoch:

- **Accuracy:** 0.9101 (91.01%)
- **Precision:** 0.9496 (94.96%)
- **Recall:** 0.9036 (90.36%)
- **False Positive Rate (FPR):** 0.0793 (7.93%)
- **Evaluation Loss:** 0.2435
- **Processing Speed:** 61.62 samples/second

## 2.3 Performance Analysis

The model shows consistent improvement across training epochs, with accuracy increasing from 91.01% to 92.59%. Key observations:

- The model maintains high precision (93.42%-94.99%) throughout training, indicating strong capability to correctly identify phishing emails when it predicts positive.
- Recall improves from 90.36% to 94.78%, showing better detection of actual phishing emails.
- The false positive is relatively at low (7.93%-11.02%), suggesting good specificity in distinguishing legitimate emails.
- The evaluation runtime was 73.81 seconds for 1137 samples, demonstrating efficient processing capability.

## 2.4 Limitations

- The false positive rate increased over epochs, suggesting potential overfitting.
- The current evaluation metrics are based on the validation set.

# References

[1] A. Al-Subaiey, M. Al-Thani, N. A. Alam, K. F. Antora, A. Khandakar, and S. A. U. Zaman, "Novel interpretable and robust web-based ai platform for phishing email detection," *ArXiv e-prints*, vol. abs/2405.11619, May 2024.

[2] Check Point Software, "Phishing detection techniques," 2025.

[3] Proofpoint, "Phishing: Threat reference," 2025.

[4] Flare.io, "Phishing detection," 2025.

[5] L. Marshall, "Theory of data processing (tdp)." `https://www.cs.up.ac.za/cs/lmarshall/TDP/TDP.html`, n.d. Accessed: 2025-05-16.

[6] C. Whittaker, B. Ryner, and M. Nazif, "Phishing detection: A literature survey," *IEEE Communications Surveys & Tutorials*, vol. 12, no. 2, pp. 209–222, 2010.

[7] W. Liu, Z. Wang, X. Liu, N. Zeng, and D. Bell, "Deep learning for phishing detection: Taxonomy, current trends and future directions," *IEEE Access*, vol. 9, pp. 23571–23594, 2021.

[8] D. Apoorva, S. Sahana, and G. Shobha, "Phishing url detection using machine learning," in *2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE)*, pp. 318–323, IEEE, 2020.

[9] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning.* MIT press, 2016.

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[11] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[13] W. Han, Y. Cao, E. Bertino, and J. Yong, "A survey on phishing email detection techniques," *IEEE Access*, vol. 8, pp. 219543–219563, 2020.

[14] M. J. M. Chowdhury, M. S. Ferdous, K. Biswas, N. Chowdhury, A. S. Kayes, M. Alazab, and P. Watters, "Phishing attack detection using machine learning and deep learning models," *IEEE Access*, vol. 9, pp. 161003–161017, 2021.