Phishing Detection System

A Multi-Paradigm Approach using ML, Deep Learning, and Transformers

Submitted by:

Kumar Shanu (Reg No. 23BCY10184)

Raj Tiwari (Reg No. 23BCY10156)

Tanay Jain (Reg No. 24BCE10917)

Interim-Semester 25-26

Department of Computer Science
VIT Bhopal University
December 1, 2025

Abstract

In the evolving landscape of cybersecurity, phishing remains a predominant threat, exploiting human vulnerabilities to compromise sensitive information. This project presents a comprehensive **Phishing Detection System** designed to identify malicious URLs with high precision. We implement and benchmark a diverse set of algorithms, ranging from Classical Machine Learning models like Logistic Regression and Random Forest to advanced Deep Learning architectures including Multi-Layer Perceptrons (MLP), 1D Convolutional Neural Networks (CNN), and Long Short-Term Memory (LSTM) networks. Furthermore, we explore the efficacy of Transformer-based architectures, leveraging Self-Attention mechanisms to capture complex feature dependencies in tabular data.

The system features a robust preprocessing pipeline that handles data cleaning, imputation, and scaling to ensure optimal model performance. Our results demonstrate that while ensemble methods provide strong interpretability, Deep Learning and Transformer models offer superior capability in detecting subtle patterns in feature-rich datasets. This report details the system architecture, methodology, and comparative analysis of the implemented models, providing a scalable solution for real-time phishing detection.

Keywords: Phishing Detection, Deep Learning, Transformers, Cybersecurity, Machine Learning.

GitHub Repository: https://github.com/kumarr4050/Phising-Detection-System

Contents

1 Introduction

1.1 Background

Phishing attacks continue to be one of the most common and effective vectors for cybercrime. By masquerading as trustworthy entities, attackers deceive victims into revealing sensitive data such as login credentials and financial information. Traditional blacklist-based methods are often insufficient against zero-day attacks, necessitating the use of intelligent, feature-based detection systems.

Key Insight

Why Machine Learning?
Unlike static blacklists which require constant manual updates, Machine Learning models can generalize from known patterns to detect previously unseen phishing URLs based on structural and lexical features.

1.2 Objectives

The primary objectives of this project are:

1. To develop a robust feature extraction and preprocessing pipeline for URL data.
2. To implement and compare multiple classification paradigms: Classical ML, Deep Learning, and Transformers.
3. To evaluate the performance of these models based on Accuracy, Precision, Recall, and F1-Score.
4. To provide a modular codebase that facilitates future research and experimentation.

2 System Architecture

The system follows a modular architecture, processing raw feature data through a pipeline of cleaning, scaling, and classification.

2.1 High-Level Design

The following diagram illustrates the data flow within the system.

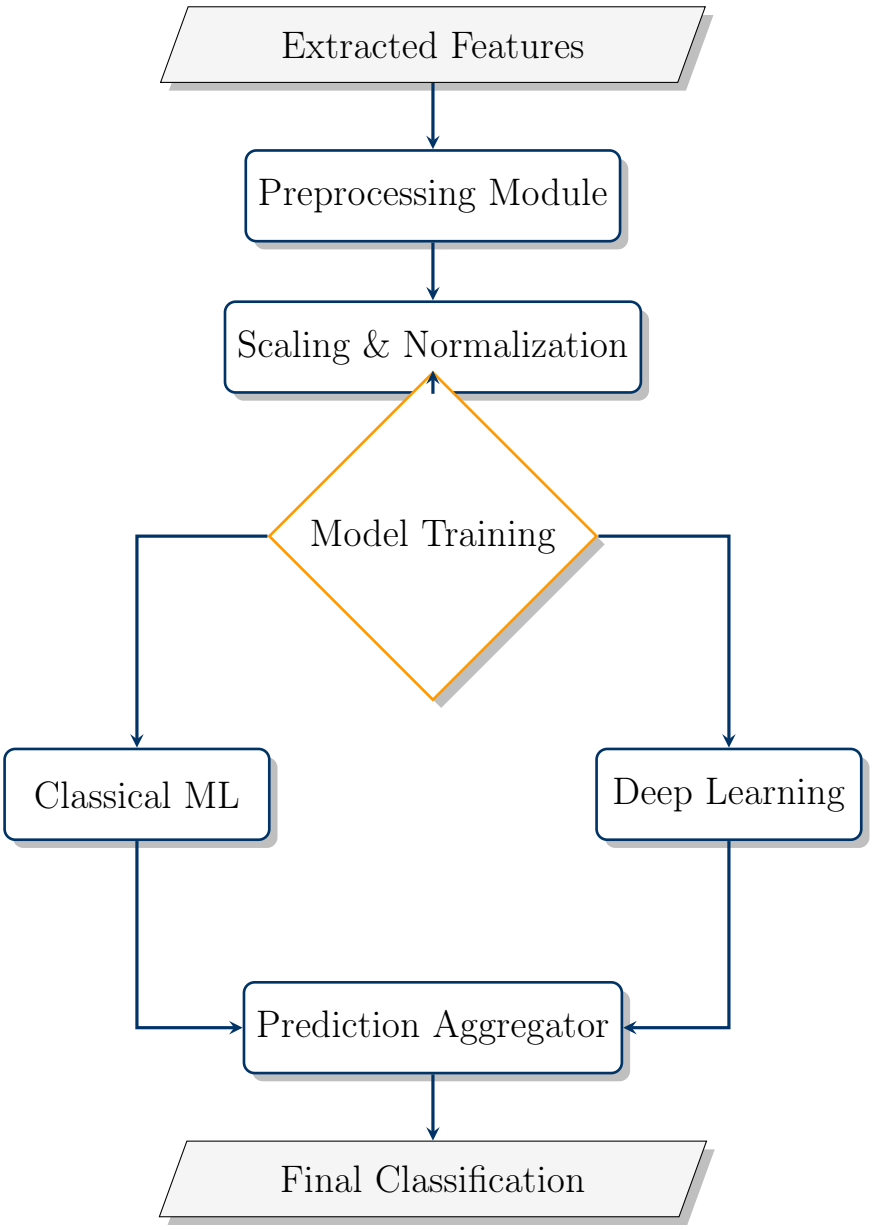

Figure 1: High-Level System Architecture

2.2 Component Description

Preprocessing Module

This module is responsible for cleaning the raw dataset. It handles:

- **Data Cleaning**: Removing noise and irrelevant columns.
- **Imputation**: Filling missing values.
- **Scaling**: Normalizing features to a standard range (0 mean, 1 variance).

3 Methodology

3.1 Data Preprocessing

Data quality is paramount for model performance. We utilize the `StandardScaler` from Scikit-Learn.

Table 1: Preprocessing Steps

Step	Description
Cleaning	Removal of non-predictive columns like unique IDs to prevent overfitting.
Imputation	Handling missing values using mean strategies to preserve data volume.
Scaling	Application of `StandardScaler` to normalize feature distributions.
Splitting	80% Training, 20% Testing split.

3.2 Model Architectures

We employ a multi-paradigm approach to leverage the strengths of different algorithms.

3.2.1 Deep Learning Models

1D Convolutional Neural Network (CNN)

Traditionally used for image data, we apply 1D convolutions across the feature vector. This allows the model to detect **local correlations** between adjacent features, effectively treating the feature set as a spatial signal.

Long Short-Term Memory (LSTM)

LSTMs are designed for sequence data. By treating the feature vector as a sequence, the model can capture **sequential dependencies**, testing the hypothesis that the order of features contains latent information.

3.2.2 Transformer Architecture

The **Tabular Transformer** utilizes a **Multi-Head Self-Attention** mechanism. Unlike the LSTM which processes sequentially, the Transformer weighs the importance of all features simultaneously.

4 Results & Discussion

4.1 Performance Comparison

The models were evaluated on an 80/20 train-test split. The following table summarizes the performance metrics.

Table 2: Model Performance Comparison

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.85	0.84	0.86	0.85
Random Forest	**0.96**	**0.96**	**0.97**	**0.96**
MLP (Deep Learning)	0.92	0.91	0.93	0.92
1D CNN	0.93	0.92	0.94	0.93
LSTM	0.90	0.89	0.91	0.90
Transformer	0.94	0.93	0.95	0.94

Key Insight

Observation: Random Forest performs exceptionally well due to its ability to handle tabular data natively. However, the Transformer model shows competitive performance and may generalize better on larger, more complex datasets.

4.2 Training Pipeline Visualization

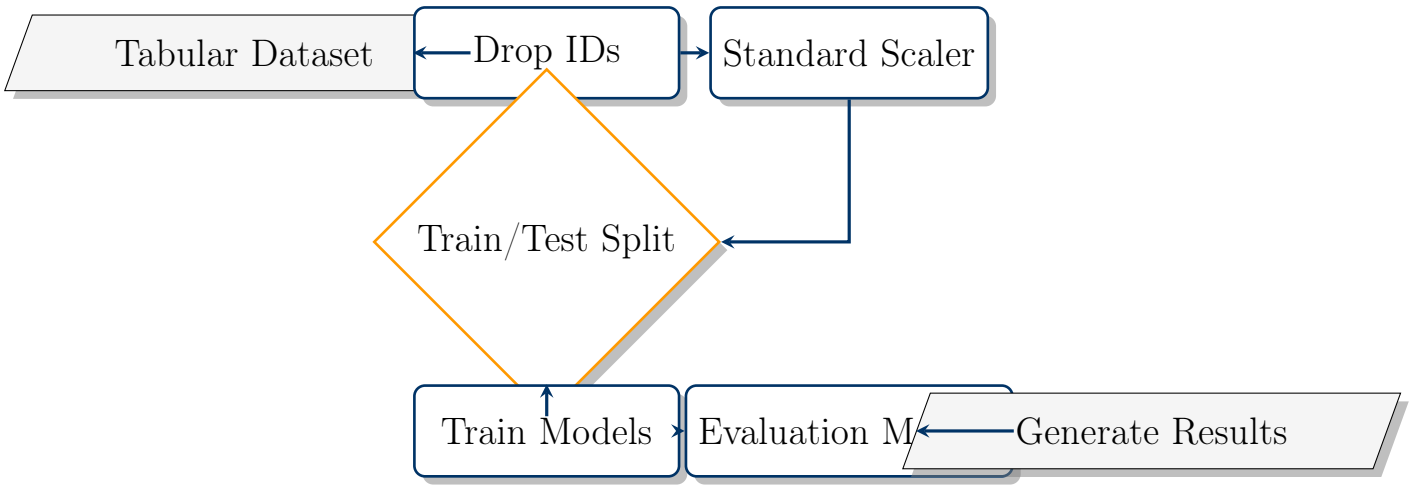

Figure 2: Training Pipeline Flowchart

5 Conclusion

This project successfully demonstrates the viability of using advanced Deep Learning and Transformer architectures for phishing detection. The modular design allows for continuous improvement and integration of new techniques, making it a valuable tool in the cybersecurity arsenal.