Deep Learning Based Detection of Phishing Websites

Week 1 Report

David Toluwani Ehigie

February 21, 2026

1 Introduction

Phishing attacks remain one of the most prevalent cybersecurity threats, exploiting human trust to steal sensitive information such as credentials, financial data, and personal records. Traditional rule-based detection techniques struggle to adapt to rapidly evolving phishing strategies.

This project aims to develop a deep learning based model for detecting phishing websites using URL and website based features.

2 Objectives of Week 1

The primary goal of Week 1 is to establish the foundation and give insight of the project. The objectives include:

- Selecting and understanding the phishing dataset
- Performing data preprocessing and feature preparation
- Building a baseline machine learning model for comparison with future deep learning models.

3 Dataset Description

This project uses the **Web Page Phishing Detection Dataset** from Kaggle.

Key properties of the dataset:

- Over 11,000 website samples
- Binary classification labels:

  - $1 \rightarrow$ Phishing website
  - $0 \rightarrow$ Legitimate website

- Features extracted from URLs and webpage properties, including:
  - URL length
  - Use of HTTPS
  - Presence of special characters
  - Domain-based features
  - Prefix and suffix patterns

These features are widely used in phishing detection research. The dataset can be found in here

4 Data Preprocessing

To ensure high-quality input for the machine learning models, several preprocessing techniques were applied to the dataset.

4.1 Data Cleaning

The dataset was inspected for missing or inconsistent values. Any null or invalid entries were removed to prevent training bias and ensure data integrity.

4.2 Feature Label Separation

The dataset consists of multiple numerical features describing website and URL characteristics. The final column represents the class label. The dataset was divided into:

- Feature matrix X
- Target label vector y

4.3 Feature Scaling

Since the dataset contains features with different ranges, **Standardization** was applied using the StandardScaler technique. This transforms each feature to have:

- Mean = 0
- Standard deviation = 1

Feature scaling improves convergence speed and performance of machine learning and deep learning models.

4.4 Train Test Split

The dataset was divided into training and testing subsets using an 80:20 ratio. The split ensures that the model is evaluated on unseen data to measure its generalization performance.

- 80% Training data
- 20% Testing data
- Random state = 42 for reproducibility

5 Baseline Model

A **Random Forest Classifier** was implemented as the baseline model. Random Forest is an ensemble classification algorithm that combines many decision trees and makes a final prediction by majority voting.

5.1 Why Random Forest?

- Strong performance on tabular data
- Resistant to overfitting
- Provides a benchmark for evaluating deep learning models

6 Initial Results

The baseline model was trained on the processed dataset and evaluated using classification metrics such as accuracy, precision, recall, and F1-score.

These results will serve as a comparison benchmark for the deep learning models developed in subsequent weeks.

7 Conclusion and Next Steps

Week 1 successfully established the project foundation:

- Dataset collected and analyzed.
- Data preprocessing pipeline implemented.
- Baseline machine learning model trained.

Next Week: Development of the first deep learning model using LSTM/CNN for phishing detection.