

Detecting Phishing Websites Using Machine Learning

K. Manichander

Roll No.: 160123737043

Cyber Security Assignment-2

October 5, 2025

1. Project Overview

Phishing is a common cyber-attack where fake websites imitate trusted sites to steal user data. Traditional detection methods often depend on blacklists of known malicious websites. However, new phishing sites appear daily, and blacklist-based approaches fail to detect unknown (zero-day) attacks.

This project implements a machine learning approach to detect phishing sites based on the structure and properties of the website URL. By analyzing key URL features such as length, use of special symbols, and presence of HTTPS, the model predicts whether a site is phishing or legitimate.

2. Research Gap

Conventional blacklist-based techniques can only recognize phishing sites that have already been reported. They are ineffective against new or modified phishing URLs. Machine learning models overcome this limitation by learning patterns in URL structures and identifying suspicious characteristics even in unseen websites.

3. Proposed Improvement

A **Random Forest Classifier** was implemented to detect phishing websites using lightweight and easily extractable URL-based features. The selected features are:

- URL length
- Presence of “@” symbol in the link
- Number of dots in the URL
- Use of HTTPS protocol
- Number of slashes and hyphens in the URL
- Length of the domain name

These features are computationally efficient and make the model suitable for real-time detection systems.

4. Methodology

The implementation followed these steps:

1. Collected a dataset of phishing and legitimate URLs.

2. Extracted URL-based features automatically using Python's `urlparse` library.
3. Encoded class labels (**bad** as 1 for phishing and **good** as 0 for legitimate).
4. Divided the data into training (75%) and testing (25%) sets.
5. Trained a Random Forest model using the `scikit-learn` library.
6. Evaluated the model using accuracy, confusion matrix, and classification report.
7. Tested the model with new sample URLs to verify real-world usability.

5. Technologies & Tools Used

- Python 3 with `pandas`, `scikit-learn`, and `joblib`
- Jupyter Notebook / VS Code for development
- CSV file for dataset storage
- Git & GitHub for project version control
- Screenshot images for documentation

6. Testing & Results

The trained Random Forest model achieved approximately **80% accuracy** on the test data. It effectively detected most phishing URLs and correctly identified legitimate ones.

The evaluation metrics were as follows:

- **Accuracy:** 80.1%
- **Confusion Matrix:** $[[7611, 1459], [2723, 9227]]$
- **Precision and Recall:** Around 0.8 for both classes

When tested with new sample URLs such as PayPal and Paytm login pages, the model successfully identified phishing attempts. An example output from the console is shown below.

```

Training Random Forest model...

Model Evaluation:
Accuracy: 80.1 %

Confusion Matrix:
[[7611 1459]
 [2723 9227]]


Classification Report:
              precision    recall  f1-score   support

     0.0         0.74      0.84      0.78       9070
     1.0         0.86      0.77      0.82      11950

 accuracy          0.80          0.80          0.80       21020
  macro avg         0.80          0.81          0.80       21020
 weighted avg         0.81          0.80          0.80       21020

Testing with new URLs:
https://www.paypal.com/account/login → Legitimate
http://freeoffer-update-account-security.com/login → Phishing
https://cbith.ac.in/ → Legitimate
http://verify-user-login-paytm-account.com → Legitimate

Model saved as phishing_detector_model.pkl

 Phishing Detection Complete!
This model can now be reused to classify new URLs.

```

7. Folder Structure

The project folder contains the following files:

- `phishing_detector.py` – Python program for detection
- `e20a870d-0350-4ce7-9a04-f849df16a6bd.csv` – URL dataset
- `phishing_detector_model.pkl` – saved trained model
- `README.md` – project documentation
- `screenshot.png` – example output screenshot

8. GitHub Repository

The complete code and dataset are available at: <https://github.com/Manichander123/phishing-detector>

9. Learning Outcomes

- Gained understanding of phishing detection using machine learning.
- Learned how to extract and use URL-based features for classification.
- Understood the working of the Random Forest algorithm for binary prediction.
- Improved coding and data-handling skills in Python.
- Learned to organize, document, and share work through GitHub.