### Rajarshi Shahu College of Engineering, Pune-411033 Department of Electronics & Telecommunication

**JSPM’s**

## A

**PROJECT REPORT**

On

# “Problem statement 1: AI/ML for Networking under Category: Network Security”

Submitted By:

**Harshada Raut  
Shreya Dandale**

Under the guidance of

**Industry Mentor - Abhishek Nandy  
and  
Product Engineer - Anil Kumar**

* GitHub: <https://github.com/shreya2794/phishing-website-detector>

# JSPM’s

## Rajarshi Shahu College of Engineering, Pune-33

### ACKNOWLEDGEMENT

It gives us immense pleasure to present this report on the project entitled “Phishing Website Detection using Machine Learning,” undertaken as part of the Intel Unnati Industrial Training 2025, under the problem statement “AI/ML for Networking” in the category of Network Security.

This work is completed during the academic year 2024–2025 at JSPM’s Rajarshi Shahu College of Engineering, Tathawade, Pune-33, Department of Computer Science and Business Systems.

We are profoundly grateful to Dr. S. P. Bhosle, Principal of RSCOE, for providing us with the necessary infrastructure and resources to carry out this project.

Our sincere thanks go to Dr. S. C. Wagaj, Head of the Department of Electronics & Telecommunication Engineering, for his constant encouragement, valuable advice, and academic support.

We would like to express our heartfelt gratitude to our industry mentors, whose professional guidance played a pivotal role in this project:

Under the guidance of  
Industry Mentor – Mr. Abhishek Nandy  
Product Engineer – Mr. Anil Kumar

Their insightful inputs, mentorship, and real-world perspectives on AI/ML applications in network security greatly enriched the quality and direction of our work.

We also extend our appreciation to all teaching and non-teaching staff of the department for their continuous support throughout the training and project phase.

This experience has been a valuable part of our academic journey, and we are sincerely thankful for the opportunity to work on a real-world problem under expert guidance.

**Students Name**

1. Student Name: Harshada Raut (RBT22ET129)
2. Student Name: Shreya Dandale (RBT22ET099)

**Index**

|  |
| --- |
| Content |
| Chapter 1. Abstract |
| Chapter 2. Introduction |
| Chapter 3. Block Diagram |
| Chapter 4. Types |
| Chapter 5. Software Aspects |
| Chapter 6. Advantages |
| Chapter 7. Disadvantages |
| Chapter 8. Applications |
| Chapter 9. Case Study:Summary |
| Chapter 10. Conclusion |
| Chapter 11. References |

**Abstract**

Phishing attacks are among the most widespread cybersecurity threats in today’s digital landscape. These attacks typically involve deceptive websites that imitate legitimate ones to trick users into revealing sensitive information such as passwords, banking details, or credit card numbers. As attackers increasingly craft sophisticated and obfuscated URLs, traditional blacklist-based detection methods often fail to provide timely protection.

This project introduces a machine learning-based phishing detection system that uses only URL-based features to classify websites as either legitimate or malicious. Unlike content-based or API-dependent solutions, this approach relies on pre-engineered numerical features extracted from URLs, making the model lightweight, fast, and suitable for offline deployment. A labeled dataset of phishing and benign URLs was used to train and evaluate a Random Forest Classifier, known for its robustness and high accuracy.

The trained model achieved a classification accuracy of approximately **96.7%**, demonstrating strong performance in distinguishing phishing websites. An interactive dashboard built with Streamlit enables users to upload CSV files for bulk classification, providing an efficient and user-friendly way to test datasets.

Overall, this project offers a scalable and practical solution for phishing detection using machine learning and is well-suited for integration into security tools, research environments, or educational platforms.

**Introduction**

In today’s digital age, the rapid expansion of internet usage has been accompanied by a significant rise in cyber threats—among which phishing attacks remain some of the most common and dangerous. Phishing is a type of cyberattack in which malicious actors impersonate trusted entities to trick users into revealing sensitive information such as usernames, passwords, or financial data. These attacks often begin with deceptive URLs that closely resemble those of legitimate websites, making them difficult to detect, even by traditional security systems.

Conventional phishing detection methods—such as blacklists and rule-based filtering—are reactive in nature and frequently struggle to keep pace with the evolving tactics used by attackers. Techniques like deep packet inspection (DPI) are also limited due to challenges with encrypted traffic, privacy regulations, and computational complexity. As a result, there is a growing demand for more proactive and intelligent threat detection strategies.

Machine Learning (ML) offers a promising alternative by enabling models to recognize patterns in data that may indicate malicious behavior. Specifically, analyzing structural features of URLs allows ML algorithms to detect phishing threats without needing to inspect website content or rely on external APIs. This approach ensures faster, more scalable, and privacy-preserving detection.

This project—developed as part of the **Intel Unnati Industrial Training 2025**—aims to create a phishing website detection system that leverages only **URL-based features**. Using a **Random Forest classifier** trained on a labeled dataset of phishing and legitimate URLs, the model achieves high classification accuracy while remaining lightweight and deployable. The project also includes an interactive **Streamlit dashboard** for user interaction, supporting **bulk classification via CSV file uploads**.

This report details the tools, methodology, implementation, and results of the project, along with possible directions for future enhancements.

**Block Diagram**

+----------------------------+

| User Input Interface |

| (CSV Upload of URLs) |

+-------------+--------------+

|

v

+----------------------------+

| URL Preprocessing |

| (Parsing, Normalization) |

+-------------+--------------+

|

v

+----------------------------+

| Feature Extraction |

| (Length, Special Characters|

| Tokens, Entropy, Symbols) |

+-------------+--------------+

|

v

+----------------------------+

| Machine Learning Model |

| (Random Forest Classifier) |

+-------------+--------------+

|

v

+----------------------------+

| Prediction Output |

| 🟢 Legitimate / 🔴 Phishing |

+----------------------------+

(Optional Offline Component)

|

v

+-----------------------------+

| Model Training Module |

| (Using Labeled URL Dataset) |

+-----------------------------+

**Types of Attacks Detected**

This phishing detection system is built to classify **URLs as either legitimate or phishing** using a machine learning model trained solely on **pre-engineered URL-based features**. It does not analyze webpage content or behavior but instead focuses on lexical and structural patterns in the URLs themselves.\

While the system can flag URLs that contain patterns **commonly associated with malicious activity**, it does **not explicitly classify attack types** such as SQL injection (SQLi) or Cross-Site Scripting (XSS).

**✅ Capabilities:**

* Detects **phishing URLs** based on suspicious characteristics such as:
  + Unusual URL length
  + Excessive use of special characters (e.g., @, //, -)
  + Presence of misleading tokens or subdomains
  + Use of IP addresses in place of domain names
* Flags **anomalous patterns** that may resemble:
  + SQL-like syntax (e.g., ' OR 1=1 --)
  + Script-related tokens (e.g., <script>, alert())

This lightweight approach enables fast, scalable URL classification that is suitable for offline or embedded systems without requiring network access or deep content analysis.

**Software Aspects**

The **Phishing Website Detection System** integrates a well-structured software stack encompassing multiple programming tools, libraries, and modular components. The design focuses on simplicity, scalability, cross-platform compatibility, and ease of deployment, making it suitable for both demonstration and integration into larger cybersecurity frameworks.

**1. Programming Language**

* **Python 3.11** is used throughout the project due to its readability, extensive community support, and rich ecosystem of libraries for machine learning, data handling, and web application development.

**2. Key Python Libraries**

| **Library** | **Purpose** |
| --- | --- |
| pandas | Data manipulation and CSV file handling |
| numpy | Numerical computations and array operations |
| scikit-learn | Machine learning model training and evaluation (Random Forest Classifier) |
| joblib | Efficient serialization of the trained ML model |
| re | Regular expressions for suspicious pattern matching in URLs |
| urllib.parse | Structured URL parsing and token extraction |
| streamlit | Building an interactive and lightweight web application interface |

**3. Development Tools & Environment**

* **IDE**: Visual Studio Code and Jupyter Notebook for development and experimentation
* **Version Control**: Git and GitHub for source code management and collaboration
* **Operating System Compatibility**: Fully cross-platform (Windows, Linux, macOS)

**4. System Architecture**

The project follows a modular structure for maintainability and reusability:

| **Module Name** | **Description** |
| --- | --- |
| feature\_extraction.py | Extracts engineered lexical and structural features from URLs |
| train.py | Trains the ML model using labeled phishing/benign URL data |
| app.py | Streamlit-based frontend for URL classification |
| dummy\_data.py *(opt)* | Generates dummy test CSVs for demonstration or testing |

* The trained model (phishing\_model.pkl) is stored in the /model directory and dynamically loaded by the app during runtime for predictions.

**5. System Interface & Deployment**

The system interface is designed with **Streamlit**, offering a clean and responsive frontend:

**Frontend Features:**

* **Single URL Prediction** – Real-time input and immediate classification output
* **Batch Prediction (CSV Upload)** – Upload pre-processed feature sets and get a table of predictions
* **Visual Feedback** – Results displayed using clear status indicators (🟢 Benign / 🔴 Malicious)

**Output Handling:**

* Real-time predictions are shown on the dashboard.
* For CSV uploads, the results are displayed in a DataFrame format, which can be exported using Streamlit’s download option.

**Deployment Options:**

* Easily deployable on:
  + **Streamlit Community Cloud**
  + **HuggingFace Spaces**
  + **Local machines** for offline demonstrations or testing
* No server-side backend, browser extensions, or scraping modules required, keeping the app **lightweight and fast**.

**Advantages**

* Fast and lightweight, as it uses only URL-based features without inspecting page content.
* Achieves high accuracy (~96%) in phishing URL detection using a Random Forest classifier.
* Provides a simple, user-friendly interface built with Streamlit.
* Supports both real-time single URL analysis and bulk classification via CSV upload.
* Does not require decryption of HTTPS traffic, preserving privacy and speed.
* Compatible with multiple platforms, including Windows, Linux, and macOS.
* Modular and maintainable codebase, allowing easy updates and extensions.

**Disadvantages**

* Limited to analyzing structural features of URLs; it does not consider the actual content of web pages.
* Cannot detect phishing based on user behavior or server-side responses.
* May fail to identify sophisticated or obfuscated phishing URLs not present in the training dataset.
* Overall performance and generalization depend heavily on the quality and coverage of the dataset.

**Applications**

* Real-time phishing detection systems for enterprise or consumer use.
* Browser-based extensions for warning users about potentially malicious URLs.
* Integration with network firewalls, intrusion detection/prevention systems.
* Filtering malicious links in emails, SMS, or messaging platforms.
* Educational tools for teaching phishing awareness and cybersecurity best practices.

**Case Study: Summary**

This project demonstrates a practical application of machine learning for combating phishing attacks through URL-based classification. A labeled dataset comprising both malicious (including phishing, SQL injection, and cross-site scripting) and benign URLs was used to train a Random Forest classifier. Feature extraction was performed directly from the URL structure, focusing on elements such as keyword presence, token patterns, entropy, and special character usage.

A lightweight and user-friendly web interface was built using Streamlit, enabling both real-time single URL analysis and batch classification through CSV file uploads. The trained model achieved a high classification accuracy of approximately 96.7%, successfully identifying various forms of threats while maintaining low computational overhead.

This case study underscores the potential of AI/ML techniques to augment traditional cybersecurity measures by providing fast, scalable, and privacy-preserving threat detection. It effectively bridges the gap between theoretical machine learning concepts and real-world cybersecurity challenges.

**Conclusion**

This project demonstrates an effective and lightweight approach to phishing website detection using machine learning techniques. By leveraging only URL-based features—such as length, entropy, token patterns, and the presence of suspicious keywords—the system avoids reliance on deep packet inspection or content analysis, resulting in a fast, scalable, and privacy-respecting solution.

The trained Random Forest classifier achieved an accuracy exceeding 96%, validating the model's effectiveness in distinguishing between legitimate and malicious URLs. The deployment of a Streamlit-based web interface further enhances accessibility, supporting both real-time individual URL analysis and bulk classification through CSV uploads.

Overall, the project successfully translates core machine learning concepts into a practical cybersecurity tool. It provides a foundation for future enhancements, including deep learning integration, real-time traffic analysis, and cloud-based deployment. As such, it offers strong potential for adaptation in real-world security infrastructures, educational environments, and personal cybersecurity applications.

**References**

* Intel Unnati Industrial Training 2025 – Problem Statement: AI/ML for Networking (Network Security)
* Scikit-learn documentation – <https://scikit-learn.org/>
* Streamlit documentation – <https://docs.streamlit.io/>
* Python urllib.parse module – <https://docs.python.org/3/library/urllib.parse.html>
* Python re module (Regular Expressions) – <https://docs.python.org/3/library/re.html>
* Joblib documentation – <https://joblib.readthedocs.io/>
* Public phishing datasets – Curated from open sources including Kaggle and academic repositories