1. Title

Title: URL Threat Classification Using Machine Learning

Author: Pranav Dhumale, Zaid Shikalgar, Atul Shetty

Internship: Cybersecurity Internship 2025

Organized by: Digisuraksha Parhari Foundation

Powered by: Infinisec Technologies Pvt. Ltd.

2. Abstract

This research presents an AI-powered tool designed to classify URLs based on their likelihood to be involved in cyber threats such as phishing, malware, or website defacement. The proposed method uses a Random Forest classifier trained on handcrafted features extracted from URLs. These include characteristics like length, subdomains, usage of HTTPS, and presence of suspicious keywords. Unlike conventional blacklist approaches, our model dynamically learns from patterns in URL structures. It achieves high detection accuracy across all classes and is built with efficiency and real-time application in mind. The tool is modular, privacy-conscious, and useful in cybersecurity operations for early threat detection and prevention.

3. Problem Statement & Objective

Malicious URLs are among the most common vectors for cyberattacks today, especially in phishing campaigns and malware distribution. Manual blacklisting is reactive and lacks scalability. The objective of this project is to build a proactive tool that can automatically detect and classify malicious URLs in real time using machine learning. This will enhance preventive cybersecurity efforts and reduce risk exposure for individuals and organizations.

4. Literature Review

Prior works rely heavily on blacklists or regular expressions, which are static and fail to detect novel threats. Ma et al. (2009) and Sahingoz et al. (2019) introduced machine learning methods for phishing detection using URL features. Deep learning models like CNNs and RNNs have been applied, but they are resource-intensive. Our research builds on these ideas using Random Forests for interpretability and performance, aligned with real-world deployment constraints.

5. Research Methodology

Dataset: CSV file of URLs labeled as benign, phishing, defacement, or malware.

Preprocessing: Removed nulls and mapped string labels to integers.

Feature Extraction: 16 handcrafted features such as URL length, presence of IPs, digits, query parameters, suspicious keywords, etc.

Model: Random Forest Classifier with class_weight='balanced'.

Evaluation: Train/test split with performance measured using classification reports.

6. Tool Implementation

extract_features.py extracts URL structure features.

train_model.py processes the dataset, trains the model, evaluates performance, and saves it as model.pkl.

check_url.py loads the trained model and performs classification of user-input URLs.

Code is modular and executable via command line with minimal dependencies.

7. Results & Observations

The tool achieved:

High precision and recall on phishing and malware URLs.

Best performance on phishing URLs due to identifiable patterns.

Low false positives on benign URLs due to balanced training.

Top features by importance included:

Use of HTTPS

Number of subdomains

Presence of suspicious keywords like “login” or “verify”

8. Ethical Impact & Market Relevance

This tool has a positive ethical impact by helping users avoid scams and system compromise. It can be integrated into email filters, browsers, or SOC (Security Operations Center) workflows. It respects privacy (no page fetching) and relies only on URL strings, avoiding legal risks of scraping or content analysis.

9. Future Scope

Future enhancements include:

Integration of WHOIS data (e.g., domain age, registration info)

Browser extension or API deployment

Incorporation of content-based and visual features

Use of deep learning or transformer models for richer representations

10. References (Minimum 10 Genuine Sources)

1. Ma, J. et al. (2009). Beyond Blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs: Used this foundational research to justify machine learning for URL classification and move beyond blacklists.

2. Sahingoz, O. K. et al. (2019). Machine Learning Based Phishing Detection from URLs: Referenced for feature-based URL analysis using supervised models like Random Forests and SVMs.

3. Le, H. M. et al. (2011). PhishDef: URL Names Say It All: Inspired the idea that URL strings alone can reveal malicious intent — justifying my decision to not scrape full web pages.

4. Zhang, J. et al. (2014). URL-Based Web Phishing Detection: Contributed to my understanding of lexical and host-based features that help classify phishing URLs.

5. Kaggle – Malicious URL Dataset: The source of the dataset or helped in benchmarking.

6. PhishTank: Used to validate or augment your dataset with real phishing URLs.

7. OpenDNS Security Reports: Helped me understand real-world threat trends and URL-based attack strategies.

8. Google Safe Browsing API: Could be cited as a real-world tool my system complements or benchmarks against.

9. Scikit-learn Documentation: Used while implementing the RandomForestClassifier and other ML tools in the code.

10. Python urllib and re modules: Power the feature extraction logic in the extract_features.py script (e.g., parsing URLs, detecting patterns).