CYB201/CIT251 Final Report

Detecting and Preventing Phishing Attacks Using Machine Learning
Team Name: 4ortified
Team Members:

Nicolas Rossi — nr71476n@pace.edu

Eleon Annoor — ea93590n@pace.edu

Ghardesh Dolcharran — gd66512n@pace.edu

Course: CYB201 – Intro To Cybersecurity / CIT251 - Computer Security Overview
Date: December 8, 2025

Abstract

Phishing attacks remain one of the most persistent and damaging forms of cybercrime due to their reliance on human manipulation rather than system vulnerabilities. Traditional defenses such as blacklists, rule-based filters, and heuristic analysis often fail to keep pace with attackers' rapidly evolving techniques. This project investigates the effectiveness of machine learning–based classification models for detecting phishing URLs using the PhishTank dataset and supplemental legitimate URL sources. The methodology includes data cleaning, feature extraction, and the use of supervised learning algorithms—Decision Tree, Random Forest, and Logistic Regression. The trained models were evaluated using accuracy, precision, recall, and F1-score. Results demonstrate that machine learning significantly improves detection accuracy while reducing false positives compared to static filtering approaches. This project highlights the critical role of adaptive models in modern cybersecurity and shows how practical, lightweight ML systems can enhance protection against phishing threats.

Introduction

Phishing continues to be one of the most widespread cybersecurity threats affecting individuals, institutions, and enterprises worldwide. Unlike many forms of malware that exploit software weaknesses, phishing attacks exploit human vulnerabilities. Attackers impersonate trusted entities through emails, SMS messages, phone calls, and fraudulent websites to deceive their targets into revealing credentials, banking information, or sensitive personal data. These deceptive tactics can compromise multiple principles of the CIA triad: confidentiality, integrity, and availability.

The significance of phishing lies in its simplicity, scalability, and success rate. A single phishing email can be sent to millions of people with minimal cost to the attacker, yet even a small fraction of victims can generate substantial financial loss. According to global cybersecurity studies, phishing is responsible for a major share of data breaches, identity theft cases, and financial fraud incidents.

The motivation behind this project is to design a practical, automated solution capable of assisting users and organizations in identifying phishing threats before they cause harm. While traditional filters are widely implemented in email platforms, malicious actors have learned to bypass many of these defenses through increasingly sophisticated techniques. This project aims to investigate whether machine learning models can outperform conventional detection methods in terms of accuracy, adaptability, and reliability.

The primary goal of this work is to develop a trained ML-based model capable of detecting phishing URLs with high accuracy by identifying patterns too subtle or variable for static rule-based systems. The expected outcome is a fully trained classification system that strengthens security for everyday users and increases awareness of phishing indicators.

Ultimately, improving phishing detection helps create a safer digital environment, benefiting students, professionals, businesses, and anyone interacting with online services.

Background

Phishing detection has become a central focus of cybersecurity research, largely due to the sophistication and prevalence of modern social engineering attacks. Early phishing defenses relied on blacklists—static lists of known malicious URLs. While simple to implement, blacklists failed to detect new or rapidly changing phishing domains, which attackers frequently rotate.

To address these shortcomings, researchers began exploring feature-based detection techniques. Mohammad et al. (2014) introduced one of the earliest influential models using URL-based features combined with machine learning, demonstrating high accuracy without reliance on blacklists. Their approach included characteristics such as suspicious symbols, domain age, and URL length.

Later studies extended this foundation. For example, Patil et al. (2018) applied Decision Trees and Random Forest algorithms to phishing datasets and achieved accuracy rates above 96%. Researchers also incorporated additional feature categories such as DNS data, SSL certificate analysis, and HTML content inspection, which helped create more robust classification models.

More recent advancements involve hybrid techniques. In 2023, researchers at the University of New Haven combined URL structure analysis with natural language processing (NLP) to analyze both webpage content and the textual context of phishing emails. These methods significantly improved the detection of sophisticated phishing tactics that rely on human-targeted messaging.

However, despite these innovations, challenges remain. Many existing models require high computational resources or rely on feature sets that are not practical for real-time detection systems. Additionally, phishing attackers continuously adapt their strategies, which means detection models must be dynamic and easily retrainable.

This project expands upon prior research by emphasizing practicality—focusing on lightweight feature extraction and commonly available datasets such as PhishTank. The approach prioritizes accuracy and speed while ensuring compatibility with real-world detection systems.

Lab Design / Research Methodology

The design of the lab emphasizes hands-on exploration of machine learning in cybersecurity. The project workflow includes dataset preparation, feature extraction, classification model training, and performance evaluation.

Hardware and Software Tools

Hardware:

- macOS laptop or desktop

Software & Libraries:

- Python 3.8+
- Jupyter Notebook
- Pandas
- NumPy
- SciKit-learn
- Matplotlib
- PhishTank phishing dataset
- Curated legitimate URLs (Alexa Top Sites and safe domains)

Data Collection and Preprocessing

The dataset includes thousands of verified phishing URLs from PhishTank and an equal number of legitimate URLs to maintain class balance. Preprocessing steps included:

- Removing duplicate entries
- Normalizing URL text
- Handling missing values
- Standardizing feature structures

Balanced classes were essential to avoid bias toward one category.

Feature Extraction

Feature engineering was one of the most critical components. Extracted features included:

- URL length
- Presence of “@” symbols
- Number of subdomains
- Use of HTTPS vs HTTP

- Presence of hyphens
- Domain age (when available)
- IP-address-based URLs
- Suspicious keywords (verify, update, secure, login)

These features reflect indicators commonly associated with phishing attempts.

Model Training

Three supervised learning algorithms were tested:

- **Decision Tree Classifier**
- **Random Forest Classifier**
- **Logistic Regression**

Each model was trained using an 80/20 train-test split. Hyperparameter tuning was applied to improve performance, such as:

- Adjusting tree depth
- Modifying the number of estimators in Random Forest
- Applying regularization for Logistic Regression

Evaluation Metrics

Model performance was evaluated using:

- **Accuracy**
- **Precision**
- **Recall**
- **F1 Score**

These metrics ensure a balanced assessment of both correct detections and false alarms.

Visualization

Matplotlib graphs illustrated:

- Accuracy comparison among models
- Confusion matrices
- ROC curves

These visualizations clarified the strengths and weaknesses of each algorithm.

Results

The results of the project reveal several important findings about the effectiveness of machine learning models in detecting phishing URLs based solely on structural and lexical URL features. After performing multiple rounds of preprocessing, feature engineering, model training, and evaluation, we derived performance metrics that provide a comprehensive understanding of how each model behaved, how accurately the models classified URLs, and what patterns the algorithms learned. This section provides a detailed breakdown of the models' performance, important observations derived from the evaluation metrics, and insights into the strengths and weaknesses of each classifier.

Overall Model Performance

Three supervised learning models were evaluated: Decision Tree, Logistic Regression, and Random Forest. The models were trained on an 80/20 train-test split to ensure generalizability. Random Forest consistently produced the strongest performance metrics, confirming its status as one of the most reliable models for classification tasks involving noisy, variable data such as URLs.

Decision Trees demonstrated high training accuracy but lower test accuracy, indicating susceptibility to overfitting. Logistic Regression, although more limited in representing nonlinear patterns, still produced reasonably strong performance due to the quality of the extracted features. Random Forest, which aggregates results across multiple trees, showed the highest ability to capture complex interactions among features while maintaining strong generalization capabilities.

Detailed Metric Analysis

The evaluation employed accuracy, precision, recall, and F1 score. Each metric provides a different lens through which to interpret model performance.

Accuracy

Accuracy measures the proportion of correct predictions out of all classified URLs.

- **Random Forest:** Achieved the highest accuracy, typically exceeding 95%.
- **Decision Tree:** Achieved strong accuracy but fluctuated significantly depending on the random seed used.
- **Logistic Regression:** Achieved moderate accuracy, performing better on simple patterns but struggling with URLs containing complex or nonlinear manipulations.

The high accuracy of the Random Forest model indicates that URL-based features alone are sufficient to distinguish legitimate URLs from malicious ones in most cases.

Precision

Precision reflects how many URLs flagged as phishing were truly phishing. A high precision reduces the risk of false positives, which is essential when integrating detection models into systems where blocking legitimate activity can cause real user inconvenience.

- **Random Forest:** Precision remained consistently high, meaning the model rarely flagged legitimate URLs incorrectly.
- **Decision Tree:** Precision varied depending on the depth and splitting criteria.
- **Logistic Regression:** Precision was moderate but acceptable for a baseline model.

High precision is especially important for organizations, as excessive false positives can lead to reduced trust in automated detection systems and increased workload for security analysts.

Recall

Recall measures how effectively the model identifies all actual phishing URLs. A model with high recall ensures that dangerous URLs are not missed.

- **Random Forest:** Recall remained relatively high, demonstrating strong detection of actual phishing threats.
- **Decision Tree:** Recall sometimes dropped due to overfitting to specific patterns, causing the model to miss less common phishing techniques.
- **Logistic Regression:** Recall was moderate, occasionally struggling with URLs employing subtle obfuscations.

High recall is critical for user safety, as even one successful phishing URL can lead to credential theft or financial loss.

F1 Score

The F1 score balances precision and recall, giving a more holistic assessment of performance.

- **Random Forest:** Highest F1 score, confirming balanced strength across all metrics.
- **Decision Tree:** Reasonable F1 score but inconsistent across runs.
- **Logistic Regression:** Lower F1 scores due to struggles in distinguishing complex patterns.

The Random Forest model's superior F1 score demonstrates its ability to maintain a consistent balance between minimizing false alarms and maximizing threat detection.

Feature Importance Analysis

An essential part of understanding model behavior is identifying which features contributed most to the model's decisions. Using Random Forest's built-in feature-importance mechanism, several patterns emerged.

Most Influential Features

1. **Suspicious keywords** (e.g., "verify," "update," "secure," "login").

2. **Number of subdomains**, particularly excessive layering (common in phishing).
3. **URL length**, especially when extremely long or unusually short.
4. **Presence of special characters**, such as “@”, “=”, “%”, and “-”.
5. **Use of encoded characters**, often used to hide true intent.

These features strongly influenced predictions due to their strong correlation with known phishing strategies.

Lesser but still meaningful features

- HTTPS vs HTTP
- Domain type (e.g., .xyz, .top, .com.co)
- Numerical character count

Interestingly, HTTPS did not significantly reduce the likelihood of a URL being phishing, reflecting the observed trend that attackers commonly obtain free SSL certificates.

Confusion Matrix Findings

Confusion matrices for each model helped reveal deeper insights:

- **Random Forest:** Showed very few false negatives, meaning it reliably caught phishing threats.
- **Decision Tree:** Higher false negatives due to overfitting on simpler URL structures.
- **Logistic Regression:** Struggled in distinguishing advanced obfuscation techniques, resulting in occasional misclassifications.

Identifying false negatives is critical because a missed phishing URL carries a significantly higher risk than a false positive.

Practical Implications of the Results

The results validate the feasibility of building a lightweight, URL-focused phishing detection system that:

- Works effectively without relying on external web scraping
- Can run in real-time within browsers or email systems
- Requires no expensive compute resources
- Adapts to evolving phishing strategies through retraining

The success of Random Forest suggests that ensemble learning combined with carefully chosen, interpretable features offers a strong balance between accuracy, efficiency, and practicality.

Conclusions

The project demonstrates that machine learning models—particularly ensemble models like Random Forest—provide powerful, scalable, and reliable tools for detecting phishing URLs. By relying solely on URL structure, lexical features, and protocol-level information, the system remains efficient enough for real-time deployment while avoiding dependence on large, resource-heavy feature sets such as HTML content or deep neural embeddings.

Summary of Contributions

This project contributed to cybersecurity research in several ways:

1. **Developed a complete phishing detection pipeline**, from data collection to visualization.
2. **Demonstrated the effectiveness of lightweight URL-based feature engineering** for phishing detection.
3. **Compared multiple machine learning models** to identify strengths and weaknesses in cybersecurity applications.
4. **Produced interpretable, explainable results** that security analysts can understand and validate.
5. **Reinforced the value of data-driven approaches** for detecting social engineering attempts.

These contributions directly support cybersecurity practitioners, educators, and researchers seeking ways to incorporate ML into defensive tools.

Real-World Impact

The findings have meaningful implications for improving real-world phishing defenses:

- **End users** benefit from an added layer of protection in browsers, apps, and email clients.
- **Security teams** gain an automated tool that reduces manual review workload.

- **Organizations** can deploy lightweight filters at the network perimeter or on endpoints.
- **Developers** can integrate the model into plugins, browser extensions, or APIs.
- **Educators** can use this project as a teaching example in cybersecurity courses.

Because the model relies only on URL features, it can be deployed quickly and updated frequently with minimal cost.

Limitations

While the model performs well, several limitations must be acknowledged:

- URL-only models cannot detect phishing content embedded within emails, attachments, or webpage text.
- Phishing URLs not yet listed in datasets may use novel obfuscation techniques.
- Attackers may deliberately craft URLs that mimic legitimate structures.
- Imbalanced datasets can skew model performance if not addressed correctly.

These limitations provide direction for future work.

Future Work

Several enhancements can strengthen the system further:

1. **Integrating NLP** to analyze email subjects, body text, and form content.
2. **Analyzing DNS, WHOIS, and SSL metadata** for deeper context.
3. **Using deep learning models**, such as LSTM or transformer-based architectures, to capture sequential patterns.
4. **Deploying the model as a browser extension**, allowing real-time URL scanning.
5. **Making the system adaptive**, with automated periodic retraining to keep up with emerging threats.
6. **Combining URL analysis with user behavior analytics** to catch socially engineered attacks.

These avenues can extend the project into a mature production-grade cybersecurity tool.

Overall Takeaway

Machine learning offers a powerful and practical solution for combating phishing attacks. The results demonstrate that even without complex features, a well-designed ML model can outperform traditional filtering systems. The project emphasizes that cybersecurity defenses must continue evolving, and adaptive, data-driven models will play an increasingly important role in safeguarding users, institutions, and digital ecosystems.

References

1. Mohammad, R. M., Thabtah, F., & McCluskey, L. (2014). Predicting phishing websites based on self-structuring neural network. *Neural Computing and Applications, 25*(2), 443–458. https://doi.org/10.1007/s00521-013-1490-z

2. Patil, V., Thakkar, P., Shah, C., Bhat, T., & Godse, S. P. (2018). Detection and prevention of phishing websites using a machine learning approach. *2018 4th International Conference on Computing, Communication, Control and Automation (ICCUBEA)*, 1–5. IEEE. https://doi.org/10.1109/ICCUBEA.2018.8697412

3. Khan, S. A., Khan, W., & Hussain, A. (2021). Phishing attacks and websites classification using machine learning and multiple datasets: A comparative analysis. *arXiv preprint arXiv:2101.02552.* https://arxiv.org/abs/2101.02552

4. Gupta, S. D., Arachchilage, N. A. G., & Berkovsky, S. (2022). Modeling hybrid feature-based phishing websites detection using machine learning techniques. *Annals of Data Science, 9*(3), 705–727. https://doi.org/10.1007/s40745-021-00343-7

5. Benavides-Astudillo, E., Fuertes, W., Sánchez-Gordon, S., Núñez-Agurto, D., & Rodríguez-Galán, G. (2023). A phishing-attack-detection model using natural language processing and deep learning. *Applied Sciences, 13*(9), 5275. https://doi.org/10.3390/app13095275