Final Project: A Comprehensive Analysis of Phishing Detection

Members

Chenyi Weng, Email: wengchen@usc.edu, USC ID: 3769237784
Zixi Wang, Email: zwang049@usc.edu, USC ID: 2854187591

Short Description

Abstract: Phishing websites pose significant cybersecurity threats by mimicking legitimate sites to steal sensitive user information. This project leverages machine learning techniques to classify URLs as either phishing or legitimate. By analyzing structural, contextual, and semantic features extracted from data sources such as OpenPhish and Tranco, the study evaluates multiple models, including K-Nearest Neighbors, Logistic Regression, Random Forest, Support Vector Machine, Naïve Bayes, and XGBoost. The analysis highlights common phishing characteristics and achieves high accuracy, contributing to improved detection mechanisms and cybersecurity.

Introduction: Phishing is a significant cybersecurity threat that exploits human vulnerabilities. This project aims to understand the distinguishing characteristics of phishing websites by analyzing labeled data. The primary objective is to develop a classification model to accurately predict phishing websites based on extracted features. This study answers two core questions:

1. What are the common characteristics of phishing websites?
2. How can data analysis and machine learning effectively identify phishing websites?

By integrating data from diverse sources, including HTML content, WHOIS metadata, and textual patterns, the project constructs robust machine-learning models to predict phishing sites. These efforts aim to enhance phishing detection capabilities, mitigate cybersecurity risks, and provide actionable insights into phishing trends.

Data Collection (Data Sources and Number of Data Samples)

1. OpenPhish Feed:
   - Source: OpenPhish’s publicly available feed.
   - Data: 500 phishing URLs.
   - Link: https://openphish.com/feed.txt
2. Tranco List:
   - Source: Tranco Top 1 Million websites list.
   - Data: 500 legitimate URLs were selected, and an additional 1,880 legitimate sublinks were generated by crawling the top 500 websites. These sublinks were included to enhance dataset diversity and match phishing URL length patterns.
   - Link: https://tranco-list.eu/top-1m.csv.zip
3. VirusTotal API:
   - Data: URL safety and threat analysis, including last analysis stats, votes, malicious counts, suspicious counts, harmless counts, and risk scores.
4. Whois API:
   - Retrieved domain registration information, such as creation date, expiration date, and registrar details, using the Python whois library.
5. Total Data Samples:
   - 500 phishing URLs from OpenPhish.

- ○ 500 legitimate URLs from Tranco.
- ○ 1,880 legitimate sublinks from Tranco.
- ○ Total: 1,000 URLs for feature extraction and model training/testing.
- ○ Additional: 2,380 URLs (500 phishing + 1,880 sublinks) processed for NLP analysis.

Data Cleaning, Analysis & Visualization

Data Cleaning: Extract key features from URLs using the extract_url_features function, including URL length, domain length, subdomains, special characters, and phishing keywords. Additional features such as external link count, script count, and phishing keyword frequency are extracted from HTML content. Advanced features such as malicious count, suspicious count, and risk score are generated using the VirusTotal API. WHOIS data provides information about domain age, expiration status, and registration details. Use the is_valid_url function to filter out invalid URLs, while removing duplicate values and features with low variance or high correlation to improve data quality. Use the coefficient of variation (CV) method to select 13 most relevant classification features and normalize numerical features. In addition, NLP techniques convert URL text into stem tokens and TF-IDF vectors to lay the foundation for text analysis.

Data Analysis: The CV method is chosen for feature selection because it can identify features with high variability relative to their mean, ensuring that they effectively contribute to classification without being dominated by noise or outliers. This helps reduce overfitting and improves the generalization ability of the model. The dataset was split into 80% training and 20% testing using the train_test_split function, and a fixed random seed was used to improve reproducibility. Four machine learning models were trained and evaluated - KNN, logistic regression, random forest, and SVM. KNN balanced overfitting and underfitting through neighbor adjustment, logistic regression provided a baseline for linear separability, random forest handled nonlinear data while providing insights into feature importance, and SVM used the RBF kernel to effectively classify nonlinear patterns. The selected features and the robust evaluation process ensured high accuracy and AUC scores for all models, laying a solid foundation for phishing URL detection.

Data Analysis and Visualization (Some examples)

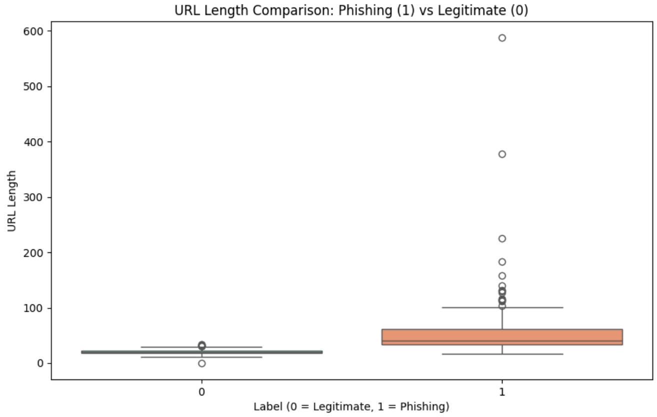

Figure 1. Boxplot of URL Length by Label

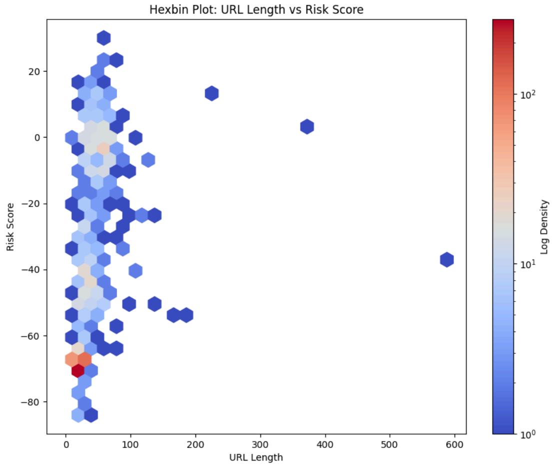

Figure 2. Hexbin Plot: URL Length vs. Risk Score

Figure 1: Box Plot of URL Length by Label - This figure illustrates the distribution of URL lengths for phishing (label = 1) and legitimate (label = 0) websites. The longer URLs are more indicative of phishing attempts, showcasing a clear disparity between the two labels.

Figure 2: Hexbin Plot of URL Length vs. Risk Score - This hexbin plot visually demonstrates the relationship between URL length and risk scores, indicating a positive correlation where longer URLs generally correspond to higher risk scores. The density of points is shown with color intensity, providing insights into common URL characteristics for phishing websites.

- Feature Importance: Ranked features using Random Forest and cumulative importance metrics.

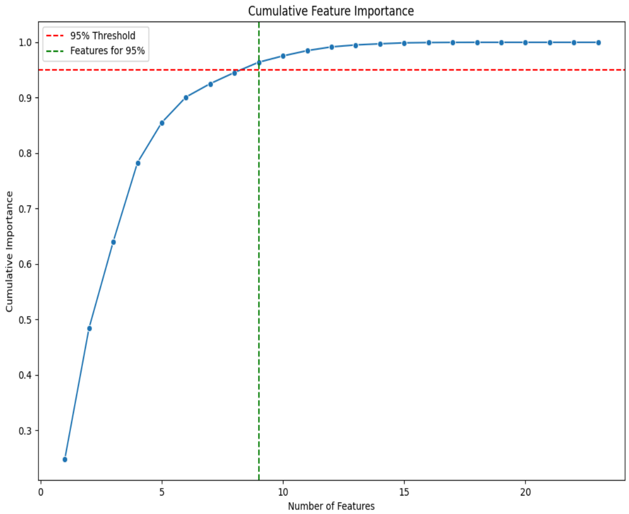

Figure 3. Cumulative Feature Importance Analysis

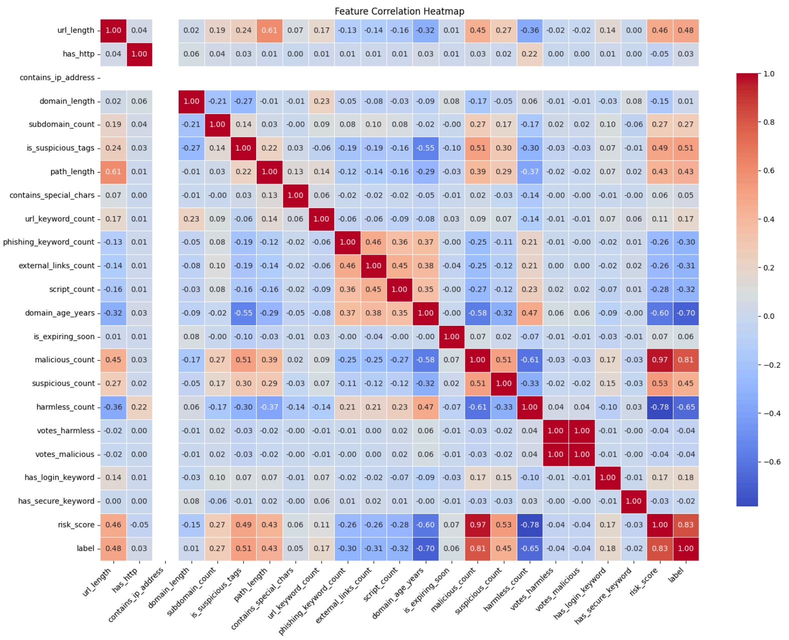

Figure 4. Feature Correlation Heatmap

Figure 3: Cumulative Importance of Features - Figure 3 highlights the cumulative importance of features in the Random Forest model, with risk_score and malicious_count emerging as the most impactful. These two features alone accounted for a significant portion of the model's predictive power, capturing key indicators of phishing activity such as risk assessment and antivirus detections. The emphasis on these features validated the effectiveness of the coefficient of variation (CV) in selecting relevant attributes, contributing to the model's perfect classification performance with an AUC of 1.000.

Figure 4: Feature Correlation Heatmap - This heatmap visualizes relationships among key features, such as URL length, risk scores, and subdomain count. Strong correlations highlight interdependencies crucial for phishing detection.

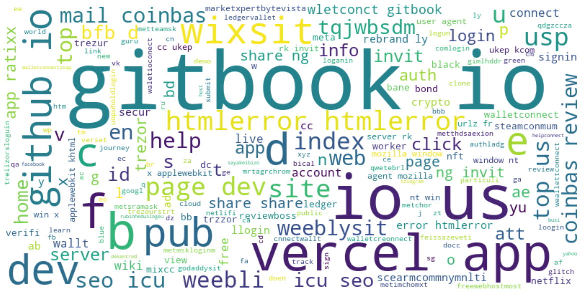

Figure 5. WordCloud for Normal URLs

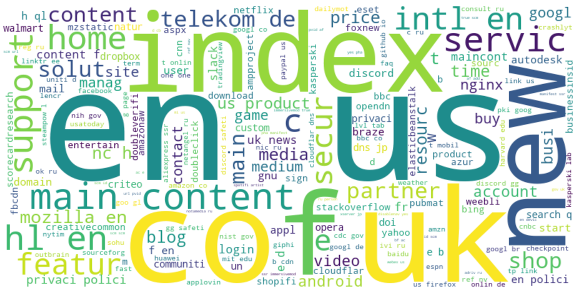

Figure 6. WordCloud for Phishing URLs

Figure 5 & figure 6: The word clouds revealed distinct patterns between phishing and legitimate URLs. Phishing URLs (label 1) frequently included keywords like "secure," "support," and "login," reflecting urgency and manipulation. Legitimate URLs (label 0) featured technical terms such as "github" and "app," indicating functionality. These differences underscore the potential of textual features for identifying phishing attempts.

Machine Learning Models: The dataset was split into 80% training and 20% testing using the train_test_split function with fixed random seeds for reproducibility. Six models—KNN, Logistic Regression, Random Forest, SVM, Naïve Bayes, and XGBoost—were evaluated:

- **KNN**: Tuned the number of neighbors (k=1) to balance overfitting and underfitting. Achieved a test accuracy of **98.32%** and an AUC of **0.982**.

- **Logistic Regression**: Provided a benchmark for linear feature separability. Achieved a test accuracy of **99.33%** and an AUC of **1.000**.
- **Random Forest**: Outperformed others with robustness and effective handling of nonlinear relationships. Achieved a perfect test accuracy of **100%** and an AUC of **1.000**.

NLP Analysis: Word clouds depicted phishing-related terms such as "login" and "secure."

- **SVM**: Used RBF kernels to capture complex patterns with competitive accuracy. Achieved a test accuracy of **97.98%** and an AUC of **0.980**.
- **Naïve Bayes**: Efficient for NLP features but limited by the uniformity of legitimate sublink text patterns. Achieved a test accuracy of **93.70%** and an AUC of **0.94**.
- **XGBoost**: Effective for structured features but less robust for NLP analysis due to dataset limitations. Achieved a test accuracy of **91.87%** and an AUC of **0.92**.

Random Forest achieved the best overall performance (Delivered perfect classification (100% accuracy), while NLP-based models highlighted areas for improvement

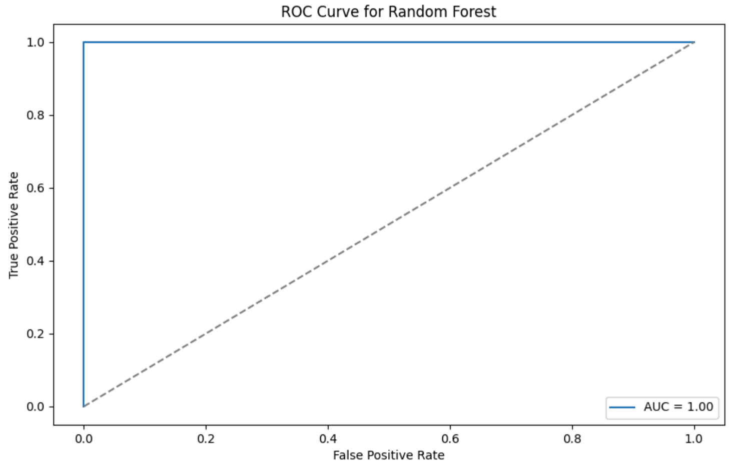

Figure 7. ROC Curve for Random Forest Model

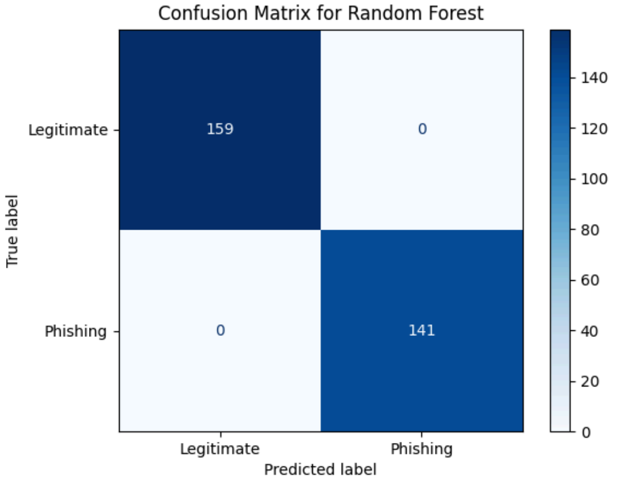

Figure 8. Confusion Matrix for Random Forest Model

Figure 7: ROC Curve for Random Forest Model - The ROC curve demonstrates the Random Forest model's classification performance, achieving an AUC-ROC value of 1.00, which indicates perfect separability between phishing and legitimate websites.

Figure 8: Confusion Matrix for Random Forest Model - This confusion matrix displays the classification results of the Random Forest model. It shows perfect classification of phishing and legitimate websites, with no false positives or negatives. Perfect classification of phishing and legitimate websites.

Results: Key Findings

- Risk Scores: Most significant feature in distinguishing phishing from legitimate websites.
- URL Length: Longer URLs were more indicative of phishing attempts.
- Malicious Count: Strongly correlated with phishing labels.

Conclusion: This project demonstrated that phishing URLs can be effectively distinguished from legitimate URLs by combining URL-based structural features, metadata, and text analysis. Visualization revealed that phishing URLs exhibited different characteristics, such as frequent use of keywords like "security" and "login," suspicious subdomains, short domain age, and high malware detection rates. Box plots showed that phishing URLs tended to be much longer than legitimate URLs, highlighting a key structural difference, while feature correlation heatmaps confirmed interdependence between features, with risk score and malware count emerging as the most predictive factors. Word clouds further highlighted the differences in keyword usage, showing that phishing URLs frequently contain emergency and promotional terms such as “security,” “support,” and “login.” URL-based features, including malware_count, phishing_keyword_count, and risk_score, were identified as the most effective predictors, with strong class separability.

Random forest achieved the best performance among all models, with an AUC of 1.000 and a test accuracy of 100%, attributed to its ensemble nature, which effectively handles nonlinear relationships and feature interactions while resisting overfitting. Logistic regression also performed well, leveraging the linear separability of features, while KNN and SVM showed competitive results but were slightly less effective due to their reliance on simpler algorithms and sensitivity to parameter tuning. Text analysis using NLP provided additional insights by capturing urgency and promotional keywords commonly used in phishing URLs. However, models like Naive Bayes and XGBoost underperformed compared to URL-based models, with AUCs of 0.94 and 0.92, due to limited diversity and uniformity in legitimate sublinks, many of which consisted of static pages or technical documents.

This project highlights the importance of combining structural, metadata, and text features for phishing URL detection. The superior performance of random forests underscores the value of ensemble models for handling complex datasets, while the lower performance of NLP-based models indicates opportunities for improvement, such as increasing dataset diversity and adopting advanced NLP models like BERT. These findings provide a strong foundation for developing practical tools to mitigate phishing threats and improve detection accuracy through richer data and more sophisticated text analysis techniques.

Changes from Original Proposal: The core goals and approach of the project remained consistent with the original plan, but adjustments were made to address challenges and improve outcomes. Initially, PhishTank was considered as a source for phishing URLs, but due to its registration restrictions, we switched to OpenPhish, which provided 500 phishing URLs. To address the limited data volume, we generated 1,880 legitimate URLs by crawling sub-links of Tranco's top 500 websites, significantly expanding the dataset and improving the analysis of URL patterns. Feature selection was refined using the coefficient of variation (CV) method to identify the 13 most relevant features, which helped enhance model performance. Missing WHOIS data was handled with error management techniques, and stricter API rate limits were managed by batching queries and adding delays. These changes improved the project’s depth while staying aligned with its original objectives.

Mention of Future Work: With additional time and resources, the project could be improved by expanding the dataset to include more phishing URLs from sources like PhishTank (if access becomes available) and collecting more diverse and content-rich legitimate website links to improve data representativeness. Further exploration of feature engineering, such as analyzing temporal patterns of URL activity or HTML structures like JavaScript obfuscation, could identify more phishing characteristics. Using advanced NLP models such as RNN or BERT could further improve classification accuracy by capturing complex patterns. Finally, developing real-time detection tools, such as browser extensions or API-based services, could make the project more practical for real-world applications.

References

1. OpenPhish: https://openphish.com
2. Tranco: https://tranco-list.eu
3. VirusTotal API: https://www.virustotal.com
4. Scikit-learn documentation for Random Forest Classifier
5. Matplotlib and Seaborn documentation for visualization techniques