

Abstract

Phishing is a cybercrime that uses deceptive URLs to trick users into revealing sensitive information. This project evaluates machine learning techniques for detecting phishing URLs using only lexical features. A labelled dataset was engineered and tested using domain-based splitting for realistic evaluation. All models achieved high accuracy, with random forest performing best, though interpretability trade-offs were noted. Lexical-based models provide a scalable solution with strong potential for real-time use and further improvement.

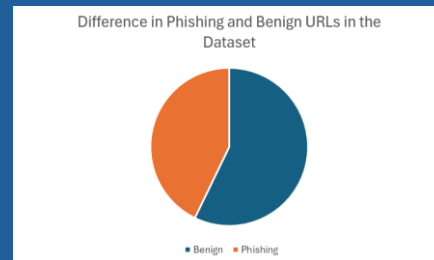
Aims and Objectives

This project aims to evaluate the effectiveness of machine learning algorithms for detecting phishing URLs using lexical features, focusing on a lightweight approach.

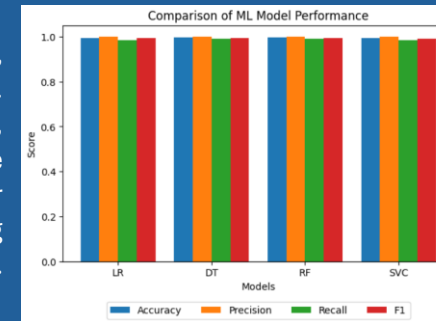
- Conduct a review of existing phishing detection methods, focusing on machine learning and URL-based features.
- Select and preprocess a publicly available dataset to ensure quality and consistency.
- Design and implement feature extraction techniques based on URL characteristics.
- Train and evaluate multiple machine learning models using metrics such as accuracy, precision, and recall.
- Compare model performance to assess trade-offs between effectiveness and interpretability, informing practical applications.

Implementation

The system follows a structured pipeline: dataset collection, preprocessing, feature extraction, model training, and evaluation. A Kaggle dataset was used, and multiple models were compared, with random forest selected for performance. The model was deployed via a Flask API with a user-friendly frontend. Development moved from Jupyter Notebook to VS Code for better structure, scalability, and integration.



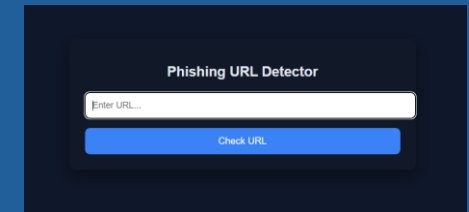
The dataset shows a moderate class imbalance, with 57% legitimate and 43% phishing URLs. While not extreme, imbalance can bias models toward the majority class, making accuracy misleading and reducing effectiveness in detecting phishing instances [34].



All models achieved over 98% performance, confirming the effectiveness of lexical features. Random forest performed best overall, balancing metrics and generalisation [46]. While other models were competitive, slightly lower recall indicates a higher risk of missing phishing URLs.

This project evaluated machine learning models in terms of both performance and interpretability, revealing a clear trade-off. Logistic regression and decision trees offered strong explainability through coefficients and rule-based logic but showed more misclassifications of phishing URLs [47]. SVM provided moderate interpretability but limited transparency in decisions. Random forest achieved the best predictive performance and generalisation, though with reduced interpretability due to its ensemble structure. Despite this, it was selected for deployment due to its superior accuracy and robustness in real-world phishing detection scenarios.

Frontend Deployment



The Random Forest model was deployed using Flask, providing an API that processes user-input URLs through the same feature extraction pipeline and classifies them using the trained random forest model. A simple HTML frontend allows users to submit URLs and view results with clear, colour-coded feedback, ensuring fast, user-friendly interaction.

Conclusion

The findings show that lexical feature-based models can reliably detect phishing URLs, with all classifiers achieving consistently high performance. Random forest offered the best balance across metrics, particularly in recall and generalisation. The results also highlight key trade-offs between interpretability and accuracy, reinforcing the importance of model choice in practical cybersecurity applications.

References

- [34] H. He and E. A. Garcia, "Learning from Imbalanced Data - IEEE Journals & Magazine," *IEEE*, 2009. <https://ieeexplore.ieee.org/document/5128907> (accessed Mar. 21, 2026).
- [46] "Ensemble Methods for Machine Learning," *Google Books*, 2023. https://books.google.co.uk/books?hl=en&lr=&id=hok3EAAQBAJ&oi=fnd&pg=PR11&dq=machine+learning+ensemble+methods&ots=FczXlp219f&sig=VNC17zGX-Xj4bloen_yt7j4Cnag&redir_esc=y#v=onepage&q=machine%20learning%20ensemble%20methods&f=false (accessed Apr. 21, 2026).
- [47] "1.10. Decision Trees — scikit-learn 0.23.2 documentation," *scikit-learn.org*. <https://scikit-learn.org/stable/modules/tree.html#complexity> (accessed Apr. 21, 2026).