

An XGBoost Approach for Phishing URL Detection and Cross-Dataset Generalization Analysis

Rutuja Dhanwate¹, Prof. P. V. Gaikwad², Dr. H. B. Jadhav³

¹ Student, ²Guide, ³Head of Department (HOD)

Department of Computer Engineering

Adsul Technical Campus, Chas, Ahilyanagar, Maharashtra, India

Abstract: *Phishing URLs can direct users to fraudulent websites designed to capture credentials, financial details, and other sensitive information. Machine-learning models can detect suspicious links by analysing URL characteristics. However, strong performance on one dataset does not necessarily transfer to a different URL population. This paper presents an analysis of an XGBoost-based approach for phishing URL detection, focusing on cross-dataset performance, feature stability, model efficiency, and browser-oriented deployment. The reported study evaluates a custom dataset containing 75,738 URLs, the Gram Beddings dataset, and a feature-compatible subset of PhiUSIIL containing 47,103 URLs. The reported accuracy is 90.72% on the custom test set, 70.65% on Gram Beddings, and 72.16% on PhiUSIIL, with an average accuracy of 77.84% across the three evaluation datasets. These results illustrate how differences between datasets can affect model performance and why independent evaluation is necessary. The analysis also highlights variations in feature importance and the trade-off between classification performance and computational efficiency. The findings support the importance of cross-dataset validation when developing practical phishing detection systems.*

Keywords: Phishing URL Detection, XGBoost, Machine Learning, Cybersecurity, Feature Engineering, Cross-Dataset Validation, Dataset Shift, Browser Security.

I. INTRODUCTION

Phishing is a social-engineering attack in which attackers attempt to deceive users into revealing confidential information or performing unsafe actions. These attacks may be distributed through emails, text messages, social media platforms, online advertisements, or compromised websites. When users trust a fraudulent link, their account credentials, banking information, and other sensitive data may be exposed.

One approach to identifying phishing attacks is to analyse the URL before the destination website is opened. A URL contains information about its length, character patterns, domain structure, security indicators, and other properties. These characteristics can help distinguish suspicious links from legitimate ones without requiring the complete web page to be downloaded and analysed.

Machine-learning techniques can learn patterns from URL features and classify addresses as legitimate or potentially malicious. However, a major challenge is ensuring that a model performs reliably outside the dataset used during development. A model may achieve high accuracy on a test set drawn from the same source as its training data but perform less effectively when evaluated on another dataset. Differences in data collection methods, time periods, URL characteristics, and labelling practices can contribute to this problem, commonly referred to as dataset shift.

The source study investigates an XGBoost-based phishing URL detection method and evaluates its performance across multiple datasets. Rather than considering accuracy alone, it examines feature stability, model efficiency, and generalization to independent data sources.



This paper reviews the methodology and reported results of that study. Its primary focus is to explain the existing detection approaches, the proposed system, the experimental setup, and the practical implications of cross-dataset performance.

II. EXISTING SYSTEMS AND RELATED WORK

A. Blocklist-Based and Rule-Based Systems

Blocklist-based systems identify suspicious URLs by comparing them with previously reported malicious addresses or domains. These systems can be effective when a phishing URL has already been identified and added to the database. However, newly created phishing URLs may not be recognised immediately.

Rule-based systems examine predefined characteristics such as URL length, unusual characters, suspicious domain structures, and security indicators. These rules are generally easy to understand and implement. Their limitation is that attackers can modify URL patterns to avoid triggering established rules. Consequently, blocklists and manually defined rules can provide useful protection, but they may not be sufficient as the only detection mechanism.

B. Traditional Machine-Learning Systems

Traditional machine-learning systems use extracted URL features to classify addresses as legitimate or malicious. Algorithms such as logistic regression, decision trees, random forests, and gradient-boosted trees can learn relationships between URL characteristics and their corresponding labels.

These systems can recognize feature combinations that are challenging to characterize with individual rules. However, the quality of the training data, the features chosen, and the degree of similarity between them are crucial for their effectiveness. Previous studies have examined machine-learning algorithms, including XGBoost, for phishing website detection. Such work provides a foundation for comparing different classification approaches and evaluating their suitability for practical security applications.

C. Deep-Learning-Based Systems

Deep-learning methods can learn complex patterns from URL strings and other website-related information. Neural networks may use character sequences, token representations, or embeddings to identify suspicious patterns.

GramBeddings is an example of research involving n-gram embeddings for URL-based phishing identification. Patterns that would be difficult to display with manually created features can be detected by deep learning techniques. However, a number of factors, including the caliber of the data and the construction of the models, affect their success. However, their computational requirements, model size, and inference time depend on the architecture and implementation.

For browser-oriented applications, these factors must be considered alongside classification performance.

D. Hybrid Detection Systems

Hybrid systems combine multiple sources of information, such as URL characteristics, domain information, HTML content, screenshots, and website behaviour. Combining these signals may improve detection because a URL alone cannot describe every aspect of a website. However, collecting and processing additional information can increase computational costs and implementation complexity.

E. Cross-Dataset Validation

Cross-dataset validation evaluates a model on data collected from a source different from the one used for training. This approach provides a more realistic indication of how a model may perform when exposed to unfamiliar URL patterns.

The reviewed XGBoost study follows this principle by evaluating the model using a custom dataset, GramBeddings, and PhiUSIIL. The reported results show that performance decreases on the independent datasets, demonstrating why evaluation on a single dataset may provide an incomplete picture of a model's practical reliability.



III. PROBLEM STATEMENT AND OBJECTIVES

A. Problem Statement

Phishing URL detection models may achieve high classification accuracy during development but experience reduced performance when applied to URLs collected from different sources. Differences in URL distributions and feature availability can affect the model's predictions. Some characteristics that are highly informative in one dataset may have considerably less predictive value in another.

Therefore, the problem addressed by the source study is the evaluation of an XGBoost-based phishing URL detection approach under different dataset conditions, with particular attention to cross-dataset generalization and feature stability.

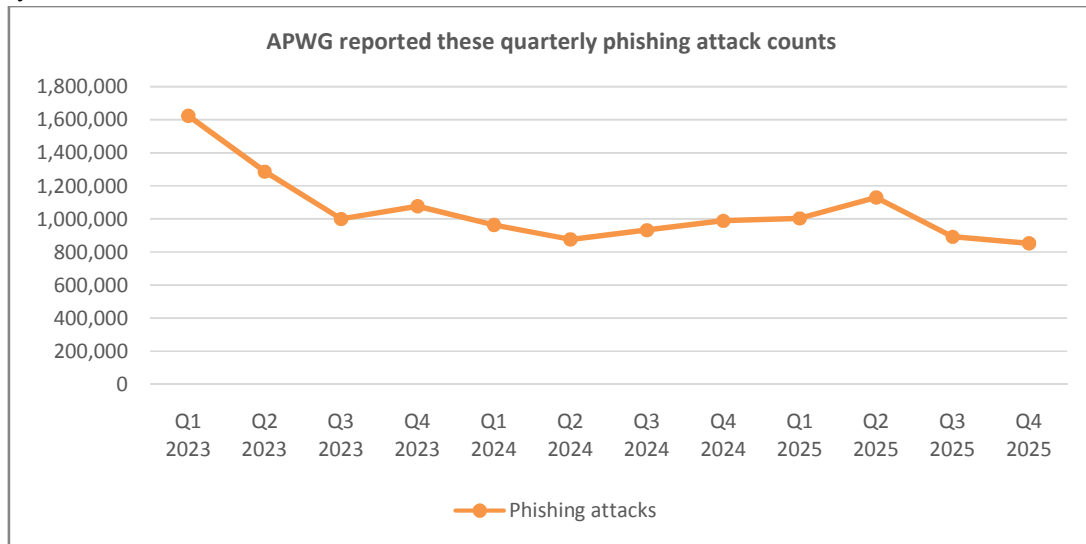


Fig. 1. Global phishing attacks reported by APWG.

B. Objectives

The main objectives are:

1. To examine an XGBoost-based approach for phishing URL detection.
2. To describe the URL feature extraction and selection process.
3. To summarize the reported model optimization and evaluation methodology.
4. To compare classification performance across the custom, GramBeddings, and PhiUSIIL datasets.
5. To examine differences in feature importance across datasets.
6. To discuss computational performance and browser-oriented deployment.

To identify limitations and possible directions for future research.

These objectives concern the analysis of the published study rather than the development of a newly trained model.

IV. PROPOSED SYSTEM AND METHODOLOGY

A. System Overview

The reviewed system accepts a URL as input and converts it into a numerical feature vector. The selected features are passed to an XGBoost classifier, which estimates the probability that the URL is associated with phishing.

For browser-oriented use, the reference design combines the classifier with an allowlist of known legitimate domains and a blocklist of known malicious domains. URLs not resolved by these lists can be evaluated by the machine-learning model.

The reported design uses three decision categories:

- Safe: Predicted probability below 0.40.



- Suspicious: Predicted probability between 0.40 and 0.80.
- Potentially harmful: Predicted probability above 0.80.

B. Dataset Description

The source study uses three datasets for model development and evaluation.

1. Custom Dataset

The custom dataset contains 75,738 URLs collected from multiple phishing and legitimate URL sources. It is used for model development and held-out testing.

2. GramBeddings Dataset

GramBeddings contains 639,723 URLs and serves as a large benchmark for feature analysis and cross-dataset evaluation.

3. PhiUSIIL Dataset

A feature-compatible subset of 47,103 URLs from PhiUSIIL is used for independent evaluation.

The datasets contain different URL populations and feature distributions. This variation allows the study to examine whether patterns learned from one data source remain useful when the model is evaluated on another.

C. Feature Extraction

The source methodology can extract more than 90 URL characteristics. These characteristics include URL length, character counts, special-character patterns, domain information, security-related indicators, reputation information, and redirect-related signals.

The extracted features are converted into numerical values that can be processed by the XGBoost classifier.

However, not every dataset provides the same features. Therefore, the source study identifies common attributes and evaluates their importance and stability before selecting the final feature set.

D. Feature Selection

The source study reports the selection of eight features from a larger set of available URL characteristics.

The reported selection criteria include:

- Mean feature importance greater than 0.02.
- Minimum feature importance greater than 0.005.
- Coefficient of variation below 0.8.
- Feature availability across at least 75% of the evaluated datasets.

These criteria aim to identify features that provide useful predictive information while remaining reasonably stable and available across different data sources. This approach is important because a feature that performs well on one dataset may not contribute equally to predictions on another dataset.

E. XGBoost Classification

XGBoost is a gradient-boosting algorithm that builds decision trees sequentially. Each new tree attempts to improve the predictions made by the existing ensemble.

The algorithm is suitable for structured tabular data, including numerical features extracted from URLs. It can model nonlinear relationships between features and provides feature-importance information that can help analyse the model's behaviour.

In the reviewed system, the selected URL features are supplied to XGBoost, which produces a prediction used to classify the URL.



F. Handling Class Imbalance

Phishing datasets may contain different numbers of legitimate and malicious URLs. This class imbalance can affect model training and prediction performance.

The source study addresses this issue through XGBoost's `scale_pos_weight` parameter. The reported optimized value is approximately 1.95.

This parameter adjusts the relative contribution of classes during training without generating synthetic URL strings.

G. Hyperparameter Optimization

The source study uses Bayesian optimization through BayesSearchCV to identify suitable XGBoost hyperparameters.

The reported optimization process includes 100 evaluations and five-fold stratified cross-validation on the custom dataset. After selecting the best configuration, the model is retrained and evaluated on the held-out custom test set and the independent datasets.

H. Evaluation Metrics

The study considers several classification metrics.

Accuracy: Measures the proportion of all predictions that are correct.

Precision: Measures the proportion of URLs predicted as phishing that are actually phishing.

Recall: Measures the proportion of actual phishing URLs correctly detected.

Specificity: Measures the proportion of legitimate URLs correctly classified as legitimate.

F1 Score: Combines precision and recall using their harmonic mean.



Fig. 2. High-level workflow for the reviewed phishing URL detection system.

V. RESULTS AND ANALYSIS

A. Performance on the Custom Dataset

The source study reports an accuracy of 90.72% on the held-out custom test set.

The reported precision is 88.0%, recall is 94.6%, specificity is 86.8%, and F1 score is approximately 91.16%.

These results indicate that the model identifies a high proportion of phishing URLs in the custom test set. However, the reported performance on this dataset alone does not establish that the same level of accuracy will be maintained on other URL populations.

TABLE

I

Performance Of Xgboost Across Different Datasets

Dataset	Accuracy (%)	
Custom Dataset	90.12	91.16
GramBeddings	70.65	73.25
PhiUSIIL	72.16	69.37
Average	77.84	77.93

B. Cross-Dataset Performance

The model's performance decreases when it is evaluated on independent datasets.

The reported accuracy is 70.65% on GramBeddings and 72.16% on PhiUSIIL. The average accuracy across the three evaluation datasets is reported as 77.84%.

The reported F1 scores are 91.16% for the custom test set, 73.25% for GramBeddings, and 69.37% for PhiUSIIL. The average F1 score is reported as 77.93%.



C. Interpretation of Results

The difference between the custom test-set results and the independent dataset results demonstrates the importance of cross-dataset evaluation.

The custom test-set accuracy is 90.72%, whereas the reported average across the three datasets is 77.84%. This indicates that the model's performance depends on the URL population used for evaluation.

Possible contributing factors include differences in URL structure, feature distributions, collection methods, and the characteristics represented in each dataset.

VI. MODEL COMPARISON AND COMPUTATIONAL PERFORMANCE

XGBoost was compared with a neural network in terms of predictive performance and computational requirements.

XGBoost achieved higher accuracy and F1-score while requiring substantially less training time. However, the neural network had lower inference latency and a smaller model size. Therefore, model selection should consider both predictive performance and deployment requirements.

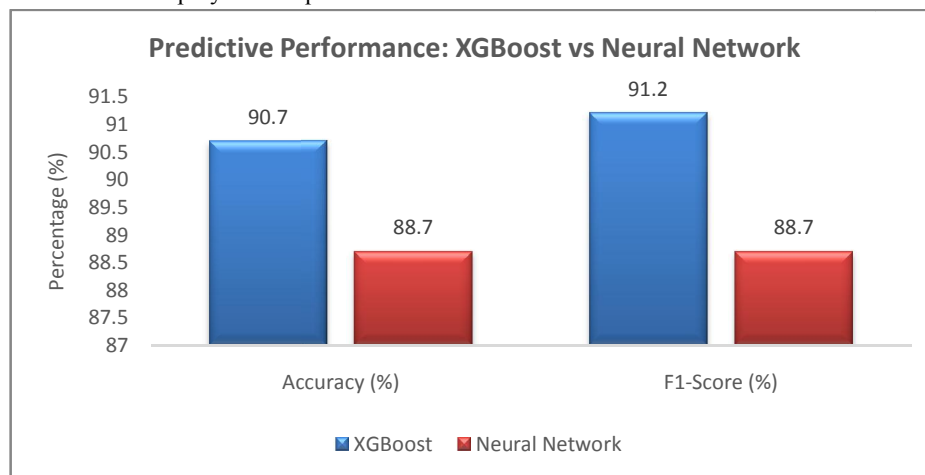


Fig. 3. Predictive Performance comparison Between XGBoost and Neural Network

VII. FEATURE STABILITY ANALYSIS

Feature importance varied across datasets. HTTPS-related features showed importance values of 0.125 for GramBeddings and 0.513 for PhiUSIL, indicating that their contribution depends on dataset characteristics.

URL length had a mean feature importance of 0.0319 and a coefficient of variation of 0.7843, suggesting considerable variation across evaluations. These findings highlight the importance of evaluating feature stability rather than relying solely on feature rankings from a single dataset.

VIII. BROWSER-BASED DEPLOYMENT

The proposed X-PHIDE system uses a browser-oriented architecture to assess potentially malicious URLs. The system integrates a FastAPI backend with Docker-based deployment to support model execution and service management.

The classifier assigns risk scores to URLs, categorizing scores below 0.40 as safe, scores above 0.80 as harmful, and intermediate scores as suspicious. This three-level approach provides a practical way to communicate uncertainty rather than forcing every URL into a strictly safe or harmful category.

IX. LIMITATIONS AND FUTURE WORK

The first limitation is that phishing behavior changes over time. A model trained on older URLs may gradually lose accuracy. Future work should include fresh labeled data and regular model updates.



URL-only detection also cannot observe everything happening on a web page. Future systems can combine URL features with DNS, WHOIS, HTML, page behavior, or other runtime signals. These additions may improve detection but will also increase processing cost. Browser deployment is another challenge because model size and memory use matter. Model pruning, smaller ensembles, quantization, ONNX or Tree lite export, and knowledge distillation can be explored to make deployment lighter. Future research should also test the model on temporally separated datasets and adversarial created URLs. The main goal should be reliable performance when conditions change, not only a high score on one dataset

X. CONCLUSION

This paper reviewed an XGBoost-based approach to phishing URL detection with cross-dataset validation. The methodology combines URL feature extraction, feature selection, Bayesian hyperparameter optimization, class-imbalance handling, and independent dataset evaluation.

The source study reports 90.72% accuracy on the custom test set and 77.84% average accuracy across the three evaluation datasets. The lower accuracy on Gram Beddings and PhiUSIIL demonstrates that strong performance on one dataset does not guarantee equivalent performance on another.

The feature analysis provides an additional finding: HTTPS has substantially different reported importance across datasets, while URL length is described as comparatively more stable. This supports the importance of evaluating multiple features rather than relying on a single indicator.

The model comparison also highlights the trade-off between classification performance, training time, inference time, and model size.

Overall, XGBoost is a practical candidate for classification using structured URL features. However, practical deployment requires ongoing evaluation, updated data, adversarial testing, and careful consideration of computational requirements. Cross-dataset validation should therefore be treated as an important component of phishing detection research.

REFERENCES

1. M. Misiek and T. Hyla, "XGBoost-Based URL Phishing Detection Method with Cross-Dataset Validation", *IEEE Access*, vol. 14, pp. 43200–43212, 2026, DOI: 10.1109/ACCESS.2026.3672690.
2. F. Rashid, B. Doyle, S. C. Han, and S. Seneviratne, "Phishing URL Detection Generalisation Using Unsupervised Domain Adaptation," *Computer Networks*, vol. 245, Art. no. 110398, 2024, doi: 10.1016/j.comnet.2024.110398.
3. Prasad and S. Chandra, "PhiUSIIL Phishing URL (Website)," *UCI Machine Learning Repository*, 2024, doi: 10.1016/j.cose.2023.103545.
4. S. Bozkir, F. C. Dalgic, and M. Aydos, "Gram Beddings: A New Neural Network for URL Based Identification of Phishing Web Pages Through N-Gram Embeddings," *Computers & Security*, vol. 124, Art. no. 102964, 2023, doi: 10.1016/j.cose.2022.102964.
5. S. K. H. Ahammad et al., "Phishing URL detection using machine learning methods," *Advances in Engineering Software*, vol. 173, Art. no. 103288, 2022, doi: 10.1016/j.advengsoft.2022.103288.
6. D. M. Divakaran and A. Oest, "Phishing Detection Leveraging Machine Learning and Deep Learning: A Review," *IEEE Security & Privacy*, vol. 20, no. 5, pp. 86–95, 2022, doi: 10.1109/MSEC.2022.3175225.
7. Hannousse and S. Yahiouche, "Towards Benchmark Datasets for Machine Learning Based Website Phishing Detection: An Experimental Study," *Engineering Applications of Artificial Intelligence*, vol. 104, Art. no. 104347, 2021, doi: 10.1016/j.engappai.2021.104347.



8. "URL Phishing Detection Using Machine Learning Techniques Based on URLs Lexical Analysis," in Proceedings of the 12th International Conference on Information and Communication Systems (ICICS), 2021, doi: 10.1109/ICICS52457.2021.9464539.
9. Aleroud and L. Zhou, "Phishing Environments, Techniques, and Countermeasures: A Survey," Computers & Security, vol. 68, pp. 160–196, 2017, Doi: 10.1016/j.cose.2017.04.006.

