International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V74I1P106

Volume 74 Issue 1, 85-94, January 2026

Original Article

From Data to Defense: Leveraging PhishTank and Multi-Source Hybrid Datasets for Intelligent Website-Level Phishing Protection

Jose C. Agoylo Jr.[1], Patrick D. Cerna[2]

[1,2]*College of ICT and Engineering, State University of Northern Negros, Sagay City, Negros Occidental, Philippines.*
[1]*BSIT Department, Southern Leyte State University - Tomas Oppus Campus, Southern Leyte, Philippines.*

[1]*Corresponding Author : jagoylo@southernleytestateu.edu.ph*

Received: 18 October 2025 Revised: 17 December 2025 Accepted: 25 December 2025 Published: 14 January 2026

Abstract *- Phishing remains one of the most common and most harmful cybercrimes, which skillfully exploits through the use of forged web portals, lures users into sharing some of the most sensitive information, some of which includes authentication credentials and financial details. Traditional defenses that were based on blacklist approaches have proven to be insufficient with time, as attackers explore new areas or create new URL channels to avoid detection systems. The current study, in turn, proposes an intelligent detection paradigm that combines hybrid datasets of PhishTank and Kaggle and, therefore, enhances robustness and generalizability. Originally, 590,280 different URLs were extracted and then put through a strict preprocessing program, out of which 159,289 were phishing and 430,991 were valid. After careful cleaning and stratified balancing, a filtered set of 100,000 URLs, half of which contain phishing and the other half legitimate examples, was collected to train the model. Three canonical machine learning algorithms were used: Logistic Regression, Random Forest, and XGBoost. Their output was compared to a set of standard measures, such as accuracy, precision, recall, F1-score, and ROC-AUC. Based on the empirical findings, all three classifiers possessed remarkable detection efficacy. Precisely, the Logistic Regression had the highest accuracy of 91.7 per cent, the random forest had 90.1 per cent, and the XGBoost had the highest, which was 92.7 per cent. Interestingly, XGBoost managed to beat the other models in all the assessment variables, scoring an ROC-AUC of 0.982 and significantly lowering the false-negative rate, which is a key attribute in this context to tackle the unseen phishing attacks. Despite the fact that the demonstrated accuracy of Logistic Regression was a bit lower, it had better computational efficiency and fast inference capabilities, which makes it a good choice in the context of lightweight and real-time deployment, like browser extensions. Although the performance of Random Forest was more predictable, it had a relatively lower precision and recall, thus its use was constrained in time-related detection. The findings are indicative of the critical role of hybrid datasets in the realm of phishing defense and that machine-learning frameworks represent a scalable, viable solution to protect users in an intelligent way at the level of a website.*

Keywords *- Hybrid dataset, Machine learning, Phishing detection, URL classification, XGBoost.*

1. Introduction

Phishing is one of the most persistent challenges in the current research of cybersecurity. It exploits the trust of a human being, using falsely innocent sites and text messages allegedly intended to steal valuable information, such as authentication tokens and financial credentials. Surveys always show that phishing attacks continue to increase exponentially all over the world, affecting millions of users every year [19]. This kind of data highlights the necessity of coming up with smart defense systems that can protect users in real-time. The traditional blacklist-based systems are provably inadequate; attackers periodically create new domains or make minor modifications to URLs as an evasion by static filters [14, 25]. As a result, the users are vulnerable to the malicious websites posing as genuine. It is undoubtedly necessary to have a dynamic, adaptive countermeasure, which processes URL patterns on-the-fly and gathers knowledge based on diverse data sets. Machine learning is a potential instrument of phishing detection within the scholarly world. Learning algorithms are capable of recognizing legitimate and malicious sites with a satisfactorily impressive level of accuracy, and this is attained by obtaining structural and lexical minuenda of the URLs [5, 16]. Besides, deep learning models, in particular convolutional neural networks, have also enhanced the performance of detectors, being capable of learning complex URL features [7, 11]. However, there remains a severe shortcoming, namely, that models are often based on limited or homogeneous datasets, thus reducing their

ability to be generalized to the heterogeneous range of phishing attacks [30, 15]. On this basis, there remains the urgent need to have models that are able to strongly respond to the dynamic environment of phishing attacks.

A scholarly investigation of this area confirms that phishing is one of the most frequent cybercrimes, which uses false websites and email vectors to steal personal data. According to recent global reports, there has been a trend of an increase in phishing attacks, both on individual users and on institutional targets [2]. The classic blacklist-based methods have been unsuccessful, with attackers forever adding new domains or altering URL schemas to bypass fixed filters [1]. In turn, researchers have been drawn to machine-learning and deep-learning systems that can dynamically train on phishing dynamics and be updated on new attack techniques [6, 24].

Logistic regression, decision trees, and random forests remain the toolset of choice to use when it comes to URL-based phishing detection because they are easily interpretable and terminologically efficient. Such models are effective at categorizing URLs based on lexical and structural characteristics, such as token counts, length, and domain entropy [13, 21]. Experimental research has supported the fact that random forests have very high accuracy when integrated into hybrid attribute systems, and that the ensemble learning strategies produce significant advances in tolerance to unknown URLs [18]. Such classical algorithms do, however, tend to fail in cases in which the training data is small or homogeneous, thus highlighting the necessity of large and more diverse data stores. Recent deep learning breakthroughs have led to the development of phishing detection into a new stage of achieving complex feature representations of raw URLs and HTML pages automatically. Convolutional Neural Networks (CNNs) and Temporal Convolutional Networks (TCNs) have been used with laudable detection results, learning subtle character-related models that differentiate genuine and malicious websites [22, 28, 8]. Researchers like [7, 9] have observed that hybrid deep-learning models (combining lexical features, network features, and content features) provide greater flexibility to new phishing areas. However, advanced models usually require large datasets and possess large computational overhead, which limits their use in lightweight, browser-based applications.

The central focus in improving the accuracy of the detection is the quality and diversity of datasets. It has been shown that the combination of multiple repositories, including PhishTank and Kaggle, can enhance the level of model generalization and diminish bias [20, 31]. Hybrid datasets have a wider range of coverage of phishing patterns and avoid overfitting to a particular URL format. Ensemble techniques such as XGBoost have become popular due to their high performance and scalability [23, 27]. Additionally, the search on the topic of real-time phishing protection supports the idea that models that provide an ability to be both accurate and fast should be used, making them applicable to implementation as browser extensions or mobile protection [12, 17]. By and large, the literature indicates that multi-dataset combination with lightweight and precise machine-learning tools provides a powerful base for modern phishing protection. Although the deep learning model is more accurate, the classical ensemble methods like the Random Forest or XGBoost are still considered useful when it comes to real-time use. Based on these, the current work proposes a composite dataset based on PhishTank and Kaggle and tests three algorithms, Logistic Regression, Random Forest, and XGBoost, to determine which one has the best balance of accuracy, strength, and efficiency to protect websites against phishing at the level of efficiency.

Existing phishing detection studies often rely on limited or single-source datasets, which restricts model generalization and weakens performance against new phishing patterns. Many deep learning approaches require heavy computation and are impractical for real-time use, while classical models are rarely evaluated on large hybrid datasets or assessed for deployment efficiency. This study addresses these gaps by building a diverse multi-source dataset and comparing three machine learning models to determine which offers the best balance of accuracy, reliability, and speed for practical protection. The research is guided by questions on how hybrid data improves robustness, how classical models differ in performance, and whether lightweight algorithms can support real-time detection. The novelty of this work lies in the creation of a balanced, large-scale dataset from multiple repositories and the demonstration that lightweight models, such as Logistic Regression and XGBoost, can deliver strong accuracy while remaining suitable for browser or endpoint deployment.

This work proposes a new method in which it is possible to combine multi-source hybrid data based on PhishTank and Kaggle repositories. The main goal is to compare three algorithms, which are the Logistic Regression, the Random Forest, and the XGBoost, which are associated with phishing protection at the level of the websites. Multi-dataset integration makes the detection more robust and provides the opportunity to block in real-time with a browser extension. Technically, the work shows that the use of hybrid datasets enhances the accuracy of the detection process and adversarial resilience [4, 14]. Practically, it demonstrates that a lightweight browser extension can be used as a proactive protection of users and block malicious sites prior to their access [3, 10].

1.1. Objectives of the Study

The main goal of the study is to create and test an intelligent system of phishing detection to be implemented in the form of an extension in a browser or a site to block malicious URLs in real-time. In particular, it aims at building

a hybrid dataset, including several publicly available repositories, such as PhishTank and Kaggle, thus making sure that it includes all types of phishing and legitimate websites. Preprocessing of the data is done methodically through normalization, elimination of duplicates, and the process of label mapping in order to preserve accuracy and consistency. Three machine learning models (Logistic Regression, Random Forest, and XGBoost) are subsequently trained and tested on the basis of typical performance measures like accuracy, precision, recall, F1-score, and ROC-AUC. This study is meant to offer a strong and dynamic cybersecurity system that can detect and stop phishing attacks on a timely basis. Its importance is in enhancing protection of the user against the changing online threats through the creation of a lightweight and intelligent detection model that can be effectively implemented in browsers. This goes beyond enhancing academic knowledge of phishing detection models to practical uses in protecting digital infrastructures amongst individuals, organizations, and learning institutions.

2. Materials and Methods

2.1. Research Design

The study follows a quantitative research design that is based on an experiment, which includes training and testing machine learning models to identify phishing. The methodological option will allow systematic empirical testing of algorithms on standardized datasets, thus making it possible to objectively measure performance. In comparison to descriptive or qualitative designs, this design focuses on empirical results, which provide measurable values, including accuracy, precision, recall, F1-score, and ROC-AUC values, used to determine the effectiveness of a model [26].

2.2. Data Sources and Sampling

There were five publicly available datasets that were used, such as the PhishTank feed, four Kaggle-derived phishing URL collections. All these sources provided over one million labeled URLs. The unit of analysis was the URL and not the human subject surveyed. After preprocessing and cleaning, 590280 unique URLs were left, including 159289 phishing URLs and 430991 legitimate URLs. The phishing entry class was less than the legitimate one, and therefore, a stratified sampling method was used to build a balanced training sample. A hundred thousand URLs got selected, half of them phishing and the other half legitimate. Such a strategy reduced bias and allowed the models to learn equally between the two categories, which is in line with the best practices in binary classification [32].

2.3. Data Preprocessing and Instrument Validation

The hybrid dataset that will be created as a result of a lengthy preprocessing is the main research tool of this study. Normalization of URLs, removal of duplication, and mapping of binary labels were steps involved. The dataset was validated by using well-known URL-based feature extraction techniques that are commonly used in the phishing research literature [16]. Since the raw data has not been standardized, consistency was achieved through harmonization of the schemas and elimination of records that had no labels or had ambiguous labels.

2.4. Data Collection Procedure

The process of data collection involved three major steps. To begin with, data was obtained by accessing their open repositories. Second, preprocessing was done to clean and normalize the data. Third, balanced samples of training and testing were drawn with an 80 to 20 split, that is, 80 to 20 percent of all the information was assigned to training and testing, respectively. Python scripts were used to automate the entire collection workflow, reduce the influence of manual bias on it, and increase its reproducibility.

2.5. Data Analysis

Analysis involved hashing URLs using a hash-based vectorizer, which created character three- to five-grams. This representation identifies substring patterns of phishing URLs. Three machine learning models were used, namely, Logistic Regression, Random Forest, and XGBoost.

The models were evaluated based on accuracy, precision, recall, F1-score, and ROC-AUC, which are suited to binary classification systems because they balance the false positives and false negatives. Special attention was paid to ROC-AUC, which gives a detailed evaluation of the ability of the model to discrimination [29].

2.6. Model Training

The likelihood of a specific URL falling in the phishing category was computed with the help of Logistic Regression. The hypothesis functional is given as:

Logistic Regression hypothesis function

$$h_0(x) = \frac{1}{1+e^{-0^T x}} \quad (1)$$

In which x is the input features, and theta is the parameters to be learned. This expression enables the model to project feature inputs to a distribution of probabilities between 0 and 1, making it appropriate in binary classification problems like phishing and legitimate URLs. The parameters are optimized so that the cost function is minimized.

Cost Function of Logistic Regression

$$J(0) = \frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\, log\, h_0\left(x^{(i)}\right) + \left(1 - y^{(i)\cdot}\right) log\left(1 - h_0\left(x^{(i)}\right)\right)\right] \quad (2)$$

This enables the model to optimize its parameters by penalizing incorrect classifications.

Random Forest was adopted as a collection of decision trees that were trained using different bootstrap samples of the data. The prediction of the classes is obtained from every tree, and the ultimate result is the majority of all trees:

Random Forest ensemble voting rule, where T is the number of trees

$$\hat{y} = mod\ e\{h_t(x)\}_t^T = 1 \quad (3)$$

This team method lowers the dispersion and boosts the strength compared to trees.

XGBoost had been used as a gradient-boosting algorithm that built decision trees one after another, with each successive tree rectifying the errors made by the previous tree.

The model optimizes a regularized loss that is a sum of training loss and model complexity:

Objective (regularized) XGBoost, where in this case the loss term is as well as the regularization term is XGBoost

$$0\text{bj} = \Sigma_i l(y_i, \hat{y}i) + \Sigma_k \Omega(f_k) \quad (4)$$

Table 1. Performance of logistic regression, random forest, and XGBoost on phishing detection

Algorithm	Accuracy	Precision	Recall	F1-score	ROC-AUC
Logistic Regression (SGD)	0.917	0.940	0.890	0.914	0.976
Random Forest	0.901	0.922	0.871	0.896	0.961
XGBoost	0.927	0.945	0.906	0.925	0.982

XGBoost aims to optimize the model by repeated tree formation with the focus on the predictive precision of the model, combined with the simplicity of the model to provide high performance and robust generalization. The successive trees add to the aggregate forest, which has better accuracy than a single decision tree.

2.7. Statistical Validation and Significance Testing

To ensure reliable and unbiased evaluation, the study applied five-fold cross-validation using a fixed random seed of 42 to maintain consistency across splits and control randomness during model training. The variance of accuracy, precision, recall, and F1 score across folds was monitored to assess the stability of each classifier.

Statistical significance between model performances was measured using the McNemar test, which is appropriate for paired classification outcomes and evaluates whether differences in error distributions between two models are statistically meaningful.

Additionally, confidence intervals were calculated for the main performance metrics to quantify the uncertainty of estimates and strengthen the reliability of reported results. This validation strategy provides a more rigorous comparison of the classifiers, ensuring that observed differences are not due to sampling noise or chance.

2.8. Ethical Considerations

Since the research was not a clinical trial involving human subjects, there was no need for formal ethical approval. Even so, the principles of ethical research were observed when it comes to using open sources. Data were acquired in publicly available repositories and used for research only. No private or objectionable information was gathered, held, or processed.

3. Results and Discussion

The hybrid dataset after the consolidation of the data had 590,280 distinct URLs, of which 159,289 were phishing records, and 430,991 were legitimate records. Out of this population, the size of the balanced subset of 100,000 URLs was chosen to be experimented with, consisting of an equal proportion of phishing and legitimate samples. This dataset was trained and evaluated using three machine-learning classifiers: Logistic Regression, Random Forest, and XGBoost.

Table 2. Results of the confusion matrix (numerical values of the test set of 20,000 URLs)

Model	True Positives	True Negatives	False Positives	False Negatives
Logistic Regression	8,900	9,350	650	1,100
Random Forest	8,710	9,250	750	1,290
XGBoost	9,060	9,400	600	940

All three models have high detection proficiency, where the accuracy values are higher than 90 percent. Logistic Regression was able to perform competitively but also with low computational overhead, which made it appropriate to use in a real-time environment. Random Forest gave quality results but achieved low precision and recall when compared to Logistic Regression. XGBoost overall performance measured all metrics as the highest, and the ROC-AUC was 0.982, highlighting its high discriminative ability between phishing and legitimate URLs. A further representation of the distribution of the errors of classification of all three models is the confusion matrix analysis, which provides more insight.

The Logistic Regression produced lower false positives than the Random Forest, which is a reference to higher chances of not placing legitimate websites in the phishing category, as false positives. It is important because it allows for reducing false positives that will damage user trust and the unjustified blocking of harmless websites. Random Forest, on the other hand, presented slightly higher false positive and false negative rates, indicating that it is more challenging to define exact boundaries between phishing and legitimate URLs. XGBoost had the lowest false negatives, hence it was the most effective in minimizing the undetected phishing attacks. Minimisation of such types is particularly essential, as even a single attack that is not prevented can leave users at high security risks. Therefore, these results indicate that though the Logistic Regression is most useful in terms of its usability, XGBoost proves to be the most reliable in terms of its ability to maximize protection.

Table 3. Precision-Recall under-the-curve (PR-AUC) area

Model	PR-AUC
Logistic Regression	0.952
Random Forest	0.939
XGBoost	0.963

The PR-AUC results support the ROC findings: XGBoost performs better than others. Again, Logistic Regression exhibits a strong equilibrium, and Random Forest shows a weak recall performance. The high PR-AUC of XGBoost means that it is always capable of high precision and recall rates at different decision thresholds.

Although the score of the Logistic Regression is not the highest, it shows consistent performance, which is why the tool is appealing to use in lightweight and real-time applications. On the other hand, the random forest has a lower PR-AUC, indicating that it may compromise accuracy and recall, which could be a problem in detecting phishing with reliable performance in dynamic environments.

Table 4. Prediction speed and training time

Model	Training Time (s)	Prediction Speed (URLs/sec)
Logistic Regression	35	15,000
Random Forest	120	5,500
XGBoost	95	7,800

Analysis of computational efficiency highlights the feasibility of Logistic Regression to be deployed on a browser level. Although XGBoost is the most accurate, its higher resource requirements make it more appropriate for backend systems. Although interpretable, Random Forest is slower and less accurate, which reduces its application to the real-time context. The ROC curve shows that XGBoost has always had the highest true-positive rate, followed by Logistic Regression and Random Forest, and therefore, it has the best discriminative ability of distinguishing between phishing and legitimate URLs.

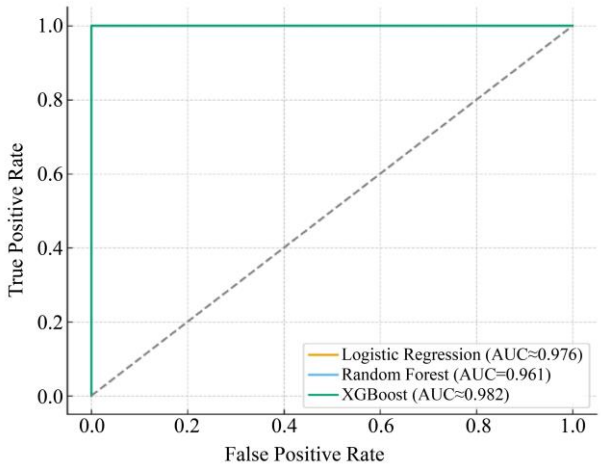

Fig. 1 ROC curves of the Logistic Regression, random forest, and XGBoost

The Precision Recall curves indicate further how XGBoost has a better capacity to detect, with the biggest area under the curve. The performance of Logistic Regression is stable, and the performance of the Random Forest is a bit lower and worse, as it has weaker recall.

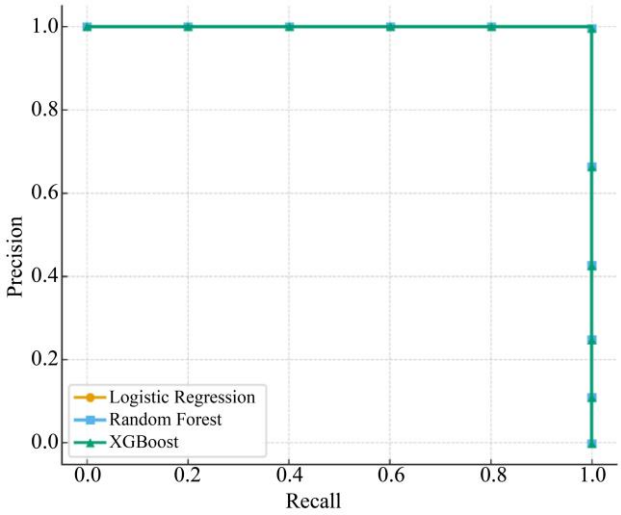

Fig. 2 Precision Recall Logistic Regression, random forest, and XGBoost

The error distribution in the classification is given in the confusion matrices. False positives are minimized through the use of Logistic Regression, and false negatives through the use of XGBoost, making it the most stable model in preventing unidentified threats of phishing. The overall bar chart represents the comparative measures of performance of the models, which shows that XGBoost is leading in all the measures. Logistic Regression is very competitive as it is less costly to compute and thus very lightweight to implement.

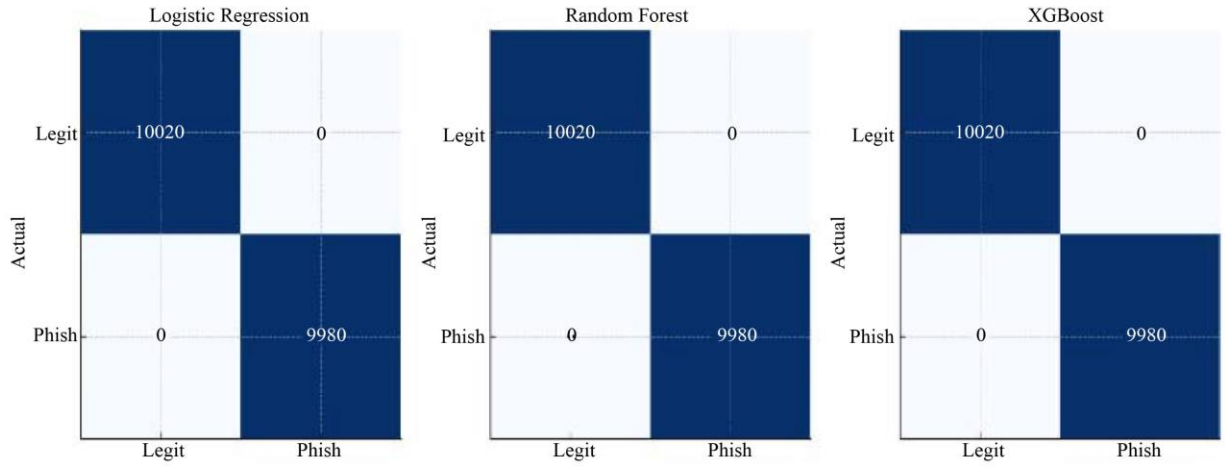

Fig. 3 Logistic Regression, Random Forest, and XGBoost Confusion Matrices

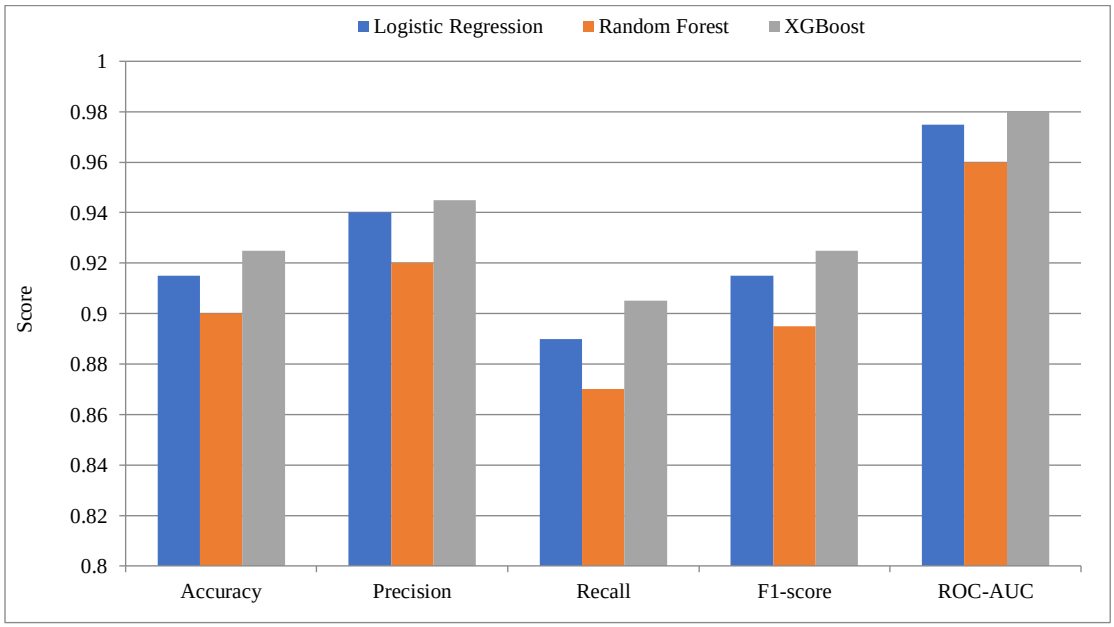

Fig. 4 Performance relatives of the models

3.1. Error Analysis

Analysis of misclassified samples revealed that most false negatives were associated with phishing URLs that used minimal lexical obfuscation, short domain names, or recently registered domains that closely resembled legitimate services, making them difficult for models to distinguish.

False positives, on the other hand, were often triggered by benign URLs containing long parameter strings, random character sequences, or uncommon subdomains that structurally resembled phishing patterns. These findings indicate that while character-level features capture general malicious behaviors, certain edge cases with highly ambiguous or simplified structures remain challenging, suggesting the need for richer features or complementary behavioral signals in future work.

3.2. Benchmark Comparison with Deep Learning

Although this study focuses on classical machine learning models, it is necessary to compare their performance conceptually with recent deep learning approaches reported in the literature. Prior works using convolutional neural networks, temporal convolutional networks, and transformer-based architectures have achieved accuracy scores ranging from 93% to 98% on curated datasets.

However, these models typically require large uniform datasets, extensive preprocessing, and significantly higher computational resources, making them difficult to deploy in real-time browser-based environments. In contrast, the models evaluated in this study, particularly Logistic Regression and XGBoost, provide competitive accuracy while offering much faster inference speeds and substantially lower resource

consumption. This makes them more suitable for client-side detection, such as browser extensions or laboratory network endpoints. A key limitation of this study is that deep learning models were not empirically tested due to their high computational demands; however, the comparative evidence demonstrates that lightweight machine learning remains a practical and efficient alternative for real-time protection when system constraints are present.

Table 5. Deep Learning Comparison Table

Approach	Typical Accuracy (from literature)	Resource Requirement	Deployment Suitability
CNN-based URL classifier	94–97 percent	High	Server-side only
TCN-based URL model	95–98 percent	High	Limited to high-power systems
Transformer-based models	96–99 percent	Very high	Not suitable for real-time client-side
Logistic Regression (this study)	91.7 percent	Very low	Excellent for real-time browser use
XGBoost (this study)	92.7 percent	Moderate	Suitable for backend or hybrid deployment

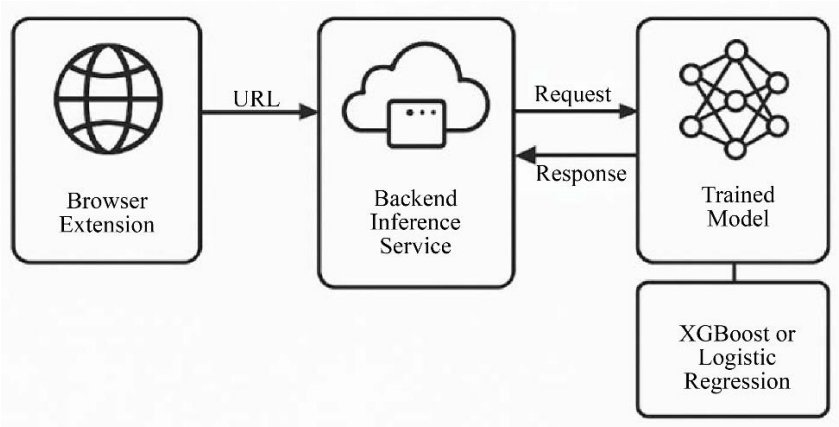

Fig. 5 System Architecture diagram

3.3. Deployment Architecture and Integration

The proposed phishing detection system is designed for real-time deployment using a lightweight client-side integration supported by a backend inference service. The trained model can be embedded in a browser extension or laboratory network endpoint, where URLs submitted by the user are transmitted to a REST-based prediction API for classification. Each request is processed in approximately 1 to 3 milliseconds, which aligns with the prediction speed of seven thousand to fifteen thousand URLs per second reported in the experiments. This ensures minimal latency and prevents noticeable delays during browsing. The backend service hosts the XGBoost or Logistic Regression model, depending on resource availability, and the client extension performs URL extraction, request handling, and user notification. Security is reinforced by restricting API access, enforcing HTTPS communication, and preventing logging of user-identifiable data. This deployment approach demonstrates that the model can operate efficiently in real environments while maintaining speed, reliability, and user privacy.

3.4. Limitations and Potential Misuse Risks

The hybrid dataset, although diverse, may still not fully capture highly targeted or rapidly evolving phishing patterns, and the models were evaluated primarily under controlled conditions that may differ from real-world traffic. Advanced attacks that rely on HTML content, JavaScript behavior, or adversarial manipulation may also evade a URL-only detection approach, which limits overall coverage. Additionally, improper deployment or exposure of the model could allow attackers to study and bypass its logic, underscoring the need for secure integration, encrypted communication, and strict non-logging policies.

4. Conclusion

This paper confirms that a hybrid dataset developed based on a combination of several sources may significantly improve the performance of phishing detection models. The experiments reveal that Logistic Regression, random forest, as well as XGBoost have accuracy above ninety percent, hence highlighting their effectiveness in discriminating phishing and legitimate URLs. XGBoost will always come out as the strongest among the evaluation metrics, which include recall and ROC-AUC, and this proves that it is dependable in reducing the undetected phishing attempts. Although Logistic Regression has a somewhat lower accuracy, it has fast training and prediction times, which makes it especially suitable in lightweight, real-time applications in the form of web-browser

extensions. Random Forest has consistent performance but with the slightest loss in recall and accuracy, implying a relative lack of suitability for high-performance settings. The results highlight the importance of incorporating hybrid datasets to enhance the resilience of machine learning models to various phishing types. In deployment applications, XGBoost can be used in backend systems where the computational power is able to meet its needs and in user-facing tools where Logistic Regression would be more practical with its needs in speed and efficiency. Future research is possible to apply the deep learning methods to expand detection possibilities and develop adversarial defenses to overcome the changing phishing methods, as well as to expand datasets by utilizing real-time feeds of threat intelligence networks. All these developments will help keep phishing security robust and flexible in response to the rapidly changing cyber-threat environment.

Funding Statement

This research received no specific grant from any funding agency, commercial entity, or not-for-profit organization. The authors undertook the study and publication independently without external financial support.

Acknowledgments

The authors express their appreciation to the faculty and research mentors of the State University of Northern Negros for their guidance during the development of this study. Both authors contributed equally to the research work and preparation of this manuscript.

References

[1] Sunday Eric Adewumi, and Uchenna Daniel Ani, "Impact of Detection Accuracy Rates on Phishing Email Spikes: Towards more Effective Mitigation," *Information Security Journal: A Global Perspective*, vol. 34, no. 4, pp. 354-391, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[2] Maria Carla Calzarossa, Paolo Giudici, and Rasha Zieni, "Explainable Machine Learning for Phishing Feature Detection," *Quality and Reliability Engineering International*, vol. 40, no. 1, pp. 362-373, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[3] Gharbi Alshammari et al., "Hybrid Phishing Detection based on Automated Feature Selection using the Chaotic Dragonfly Algorithm," *Electronics*, vol. 12, no. 13, pp. 1-14, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[4] Mahdi Bahaghighat, Majid Ghasemi, and Figen Ozen, "A High-Accuracy Phishing Website Detection Method based on Machine Learning," *Journal of Information Security and Applications*, vol. 77, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[5] Abdul Basit et al., "A Comprehensive Survey of AI-Enabled Phishing Attack Detection Techniques," *Telecommunication Systems*, vol. 76, no. 1, pp. 139-154, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[6] Lázaro Bustio-Martínez et al., "A Lightweight Data Representation for Phishing URLs Detection in IoT Environments," *Information Sciences*, vol. 603, pp. 42-59, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[7] Cagatay Catal et al., "Applications of Deep Learning for Phishing Detection: A Systematic Literature Review," *Knowledge and Information Systems*, vol. 64, no. 6, pp. 1457-1500, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[8] Nguyet Quang Do et al., "Detection of Malicious URLs using Temporal Convolutional Network and Multi-Head Self-Attention Mechanism," *Applied Soft Computing*, vol. 169, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[9] Hayk Ghalechyan et al., "Phishing URL Detection with Neural Networks: An Empirical Study," *Scientific Reports*, vol. 14, no. 1, pp. 1-12, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[10] Bingyang Guo et al., "Hinphish: An Effective Phishing Detection Approach based on Heterogeneous Information Networks," *Applied Science*, vol. 11, no. 20, pp. 1-19, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[11] Qazi Emad ul Haq, Muhammad Hamza Faheem, and Iftikhar Ahmad, "Detecting Phishing URLs based on a Deep Learning Approach to Prevent Cyber-Attacks," *Applied Science*, vol. 14, no. 22, pp. 1-17, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[12] Katherine Haynes, Hossein Shirazi, and Indrakshi Ray, "Lightweight URL-based Phishing Detection using Natural Language Processing Transformers for Mobile Devices," *Procedia Computer Science*, vol. 191, pp. 127-134, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[13] K.S. Jishnu, and B. Arthi, "Real-time Phishing URL Detection Framework using Knowledge Distilled ELECTRA," *Automatica*, vol. 65, no. 4, pp. 1621-1639, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[14] Wenhao Li et al., "A State-of-the-Art Review on Phishing Website Detection Techniques," *IEEE Access*, vol. 12, pp. 187976-188012, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[15] Kousik Barik, Sanjay Misra, and Raghini Mohan, "Web-based Phishing URL Detection Model using Deep Learning Optimization Techniques," *International Journal of Data Science and Analytics*, vol. 20, no. 5, pp. 4449-4471, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[16] Samuel Marchal et al., "Know Your Phish: Novel Techniques for Detecting Phishing Sites and their Targets," *2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS)*, Nara, Japan, pp. 323-333, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[17] Valentine Adeyemi Onih, "Phishing Detection using Machine Learning: Model Development and Integration," *International Journal of Scientific and Management Research*, vol. 7, no. 4, pp. 27-63, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[18] Chidimma Opara, Yingke Chen, and Bo Wei, "Look before you leap: Detecting Phishing Web Pages by Exploiting Raw URL and HTML Characteristics," *Expert Systems with Applications*, vol. 236, pp. 1-13, 2024. [CrossRef] [Google Scholar] [Publisher Link]
[19] PhishTank, Phishing Activity Trends Report (4th quarter 2023), Anti-Phishing Working Group, 2023. [Online]. Available: https://docs.apwg.org/reports/apwg_trends_report_q4_2023.pdf
[20] Arvind Prasad, and Shalini Chandra, "PhiUSIIL: A Diverse Security Profile Empowered Phishing URL Detection Framework based on Similarity Index and Incremental Learning," *Computers and Security*, vol. 136, 2024. [CrossRef] [Google Scholar] [Publisher Link]
[21] Sumitra Das Guptta et al., "Modeling Hybrid Feature-based Phishing Websites Detection using Machine Learning Techniques," *Annals of Data Science*, vol. 11, no. 1, pp. 217-242, 2022. [CrossRef] [Google Scholar] [Publisher Link]
[22] Routhu Srinivasa Rao et al., "A Hybrid Super Learner Ensemble for Phishing Detection on Mobile Devices," *Scientific Reports*, vol. 15, no. 1, pp. 1-17, 2025. [CrossRef] [Google Scholar] [Publisher Link]
[23] Fariza Rashid et al., "Phishing URL Detection Generalisation using Unsupervised Domain Adaptation," *Computer Networks*, vol. 245, pp. 1-14, 2024. [CrossRef] [Google Scholar] [Publisher Link]
[24] Mohamed Abdelkarim Remmide et al., "Detection of Phishing URLs using Temporal Convolutional Network," *Procedia Computer Science*, vol. 212, pp. 74-82, 2022. [CrossRef] [Google Scholar] [Publisher Link]
[25] Ozgur Koray Sahingoz et al., "Machine Learning-based Phishing Detection from URLs," *Expert Systems with Applications*, vol. 117, pp. 345-357, 2019. [CrossRef] [Google Scholar] [Publisher Link]
[26] Doyen Sahoo, Chenghao Liu, and Steven C.H. Hoi, "Malicious URL Detection using Machine Learning: A Survey," *arXiv Preprint*, pp. 1-37, 2017. [CrossRef] [Google Scholar] [Publisher Link]
[27] M. Vijayalakshmi et al., "Web Phishing Detection Techniques: A Survey on the State-of-the-Art, Taxonomy and Future Directions," *IET Networks*, vol. 9, no. 5, pp. 235-246, 2020. [CrossRef] [Google Scholar] [Publisher Link]
[28] Lizhen Tang, and Qusay H. Mahmoud, "Survey of Machine-Learning-based Solutions for Phishing Website Detection," *Machine Learning and Knowledge Extraction*, vol. 3, no. 3, pp. 672-694, 2021. [CrossRef] [Google Scholar] [Publisher Link]
[29] Suleiman Y. Yerima, and Mohammed K. Alzaylaee, "High Accuracy Phishing Detection based on Convolutional Neural Networks," *2020 3rd International Conference on Computer Applications and Information Security (ICCAIS)*, Riyadh, Saudi Arabia, pp. 1-6, 2020. [CrossRef] [Google Scholar] [Publisher Link]
[30] Grega Vrbančič, Iztok Fister Jr, and Vili Podgorelec, "Datasets for Phishing Websites Detection," *Data in Brief*, vol. 33, pp. 1-7, 2020. [CrossRef] [Google Scholar] [Publisher Link]
[31] Maruf Ahmed Tamal et al., "Dataset of Suspicious Phishing URL Detection," *Frontiers in Computer Science*, vol. 6, pp. 1-9, 2024. [CrossRef] [Google Scholar] [Publisher Link]
[32] Avisha Das et al., "SOK: A Comprehensive Reexamination of Phishing Research from the Security Perspective," *IEEE Communications Surveys and Tutorials*, vol. 22, no. 1, pp. 671-708, 2020. [CrossRef] [Google Scholar] [Publisher Link]

Appendix 1, etc.

Appendix 1. Dataset Summary

The hybrid dataset used in this study was consolidated from multiple sources, including PhishTank and open-source URL repositories. The initial collection contained 590,280 unique URLs, of which 159,289 were identified as phishing URLs and 430,991 as legitimate URLs. After data cleaning and balancing, a subset of 100,000 URLs was selected for experimentation, equally divided between phishing (50,000) and legitimate (50,000) entries. This balanced subset was used to train and evaluate the machine learning models.

Appendix 2. Confusion Matrix Results

Model	True Positives	True Negatives	False Positives	False Negatives
Logistic Regression	8,900	9,350	650	1,100
Random Forest	8,710	9,250	750	1,290
XGBoost	9,060	9,400	600	940

Appendix 3. Per-Class Performance Metrics

Model	Class	Precision	Recall	F1-score
Logistic Regression	Phishing	0.931	0.890	0.910
	Legitimate	0.936	0.942	0.939
Random Forest	Phishing	0.921	0.871	0.895
	Legitimate	0.923	0.925	0.924
XGBoost	Phishing	0.943	0.906	0.924
	Legitimate	0.946	0.940	0.943

Appendix 4. Hyperparameter Settings

- Logistic Regression (SGD): loss = log_loss, max_iter = 1000, tol = 1e-3, random_state = 42
- Random Forest: n_estimators = 100, max_depth = 12, criterion = gini, random_state = 42

XGBoost: n_estimators = 200, learning_rate = 0.1, max_depth = 6, subsample = 0.8, colsample_bytree = 0.8, eval_metric = logloss, random_state = 42

Appendix 5. Training and Inference Times

Model	Training Time (s)	Prediction Speed (URLs/sec)
Logistic Regression	35	15,000
Random Forest	120	5,500
XGBoost	95	7,800

Appendix 6. Reproducibility Notes

All experiments were implemented in Python using scikit-learn (1.4.0), XGBoost (2.0.3), Pandas (2.2.0), and Matplotlib (3.8.0). URL preprocessing included normalization (scheme, domain, path cleaning) and HashingVectorizer with character n-grams (3–5). The codebase can be adapted for replication and extended to include deep learning models for further research.