Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 4, Juni 2026
e-ISSN 2685-998X
DOI 10.30865/json.v7i4.9777

Client-Side Hybrid Machine Learning for Real-Time Phishing URL Detection in Chrome Extension

Evta Indra*, Wahyu Soekanta Ginting, Katrin Wijay, Nuragustyani Br Bangun
Information Systems, Universitas Prima Indonesia, Medan, Indonesia
Email: [1,*]evtaindra@unprimdn.ac.id, [2]wahyusoekantaginting@gmail.com, [3]katrinwijaya1004@gmail.com, [4]kimagstynbgn31@gmail.com
Correspondence Author Email: evtaindra@unprimdn.ac.id*
Submitted: **07/05/2026**; Accepted: **06/06/2026**; Published: **30/06/2026**

Abstract− Phishing attacks continue to pose a major cybersecurity challenge, with traditional server-side and deep learning detection methods often struggling with network latency, privacy risks, and high computational overhead. This research focuses on creating a lightweight phishing URL detection system that operates in real time on the client side, implemented as a Google Chrome extension. The system employs a hybrid machine learning architecture combining Logistic Regression (LR) and Random Forest (RF) to evaluate 22 purely lexical URL features extracted locally. Evaluated using 5-fold cross-validation, the base models demonstrated high stability with mean performance scores of $\mu = 0.9960$ for LR and $\mu = 0.9968$ for RF. Crucially, confusion matrix analysis revealed that the hybrid approach perfectly resolved the trade-off between user accessibility and security. It successfully captured 20,037 phishing links (True Positives) and permitted 26,957 legitimate URLs (True Negatives). It minimized false alarms to just 13 False Positives, matching LR's efficiency, while also reducing missed detections to 152 False Negatives. Furthermore, a live implementation test on 100 URL samples achieved 92% accuracy rate in providing real-time visual risk alerts. By performing all feature extraction and analysis locally without transmitting data externally, this system overcomes algorithmic trade-offs and network latency, delivering a dependable and user-friendly defense without compromising browsing performance.

Keywords: Chrome Extension; Cross Validation; Hybrid Machine Learning; Phishing URL; Real-Time Security

1. INTRODUCTION

The rapid advancement of digital technology has brought significant changes in many areas of life, making the internet the primary platform for accessing public and financial services. However, behind this convenience comes a cybersecurity threat, especially phishing, which is increasing rapidly. The urgency of this threat is further exacerbated by the fact that attackers are now actively exploiting Artificial Intelligence (AI) and Large Language Models (LLM) to design and propagate phishing websites automatically and on a massive scale [1], [2]

Conventional phishing detection methods, which have traditionally relied on blacklists and centralized (server-side) heuristic analysis, are increasingly proving to be ineffective. [3] These approaches have major limitations in dealing with highly dynamic zero-day attacks, frequently suffer from network latency, and pose severe privacy violation risks by requiring users' browsing history to be transmitted to third-party servers.[4]

Over time, various machine learning (ML) and deep learning (DL) approaches have been implemented for automated phishing detection [5] Recent studies have shown impressive accuracy using Deep Learning models such as Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), and Artificial Neural Networks (ANN) [6], [7]. However, Deep Learning models demand heavy computational resources, prolonged iteration times (epochs), and significant memory capacity constraints. [8], [9], [10]. Furthermore, excessive semantic feature extraction methods, such as natural language processing (NLP) or full HTML code parsing, have proven to drain inference time significantly. [11], [12]. These approaches make them unsuitable and excessively heavy for real-time execution directly on the user's browser (client-side).

For environments with limited resources, traditional machine learning algorithms provide much better computational efficiency. Logistic Regression (LR) has consistently proven effective as a frontline classifier due to its fast fitting speed and low false-positive rates in detecting cyber threats. [13], [14]. Meanwhile, Random Forest (RF) is commonly acknowledged in numerous comparative studies as the classification algorithm with the highest accuracy and stability when dealing with complex datasets. [15], [16], [17].

While traditional machine learning techniques have been widely studied, the shift from conventional models to practical client-side security tools still faces significant challenges [5]. Previous implementations of client-side systems (extensions or plug-ins) have often only been validated on very small and unrepresentative dataset scales. [18]. In addition, the literature indicates that reliance on a single model (such as Random Forest alone) still leads to detection failures (high false-negative rates) due to feature overlap or collinearity [19], [20] A recent finding demonstrates that a Hybrid approach combining Logistic Regression and Random Forest yields network attack classification capabilities that are far superior to those of its constituent standalone models. [21]. However, this powerful architecture has not yet been tailored specifically into a lightweight, real-time phishing URL detection system.

Addressing this research gap, this study introduces an innovative phishing detection system deployed entirely on the client side through a Google Chrome extension. This system uses a hybrid machine learning approach, combining the inference efficiency of Logistic Regression as an initial detection model with the

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 4, Juni 2026
e-ISSN 2685-998X
DOI 10.30865/json.v7i4.9777

precision of Random Forest as an advanced verification layer to suppress false-negative rates. To ensure real-time extension efficiency, this research avoids heavy page content extraction and specifically selects only 22 essential features derived purely from the Uniform Resource Locator (URL) structure. The model is comprehensively validated using the latest URL dataset collection from the PhiUSIIL dataset [22]

To ensure that the hybrid model generalizes well without overfitting to the training data and remains resilient to evolving cyber threats, it is thoroughly tested with k-fold cross-validation. This research aims to deliver a practical, real-time phishing alert system that is lightweight, highly precise, and protective of user privacy.

2. RESEARCH METHODOLOGY

This study is an applied research project employing quantitative experimental methods. It focuses on designing, testing, and evaluating a machine learning-based phishing detection system that is embedded directly within a web browser environment. The experimental methodology is used to measure the model's accuracy in classifying URLs and to evaluate the system's effectiveness under real-world conditions.

2.1 Data Source and Dataset

The main dataset used for this research was obtained from the University of California Irvine (UCI) public repository, specifically the PhiUSIIL Phishing URL Dataset [22]. Before analysis, the dataset underwent a thorough cleaning process, which involved standardizing URL formats, removing null entries, eliminating duplicate URLs, and discarding invalid data to maintain consistency throughout the dataset. After cleaning, the class distribution showed an imbalance, with 134,850 legitimate URLs (57.2%) and 100,945 phishing URLs (42.8%), as illustrated in Figure 1.

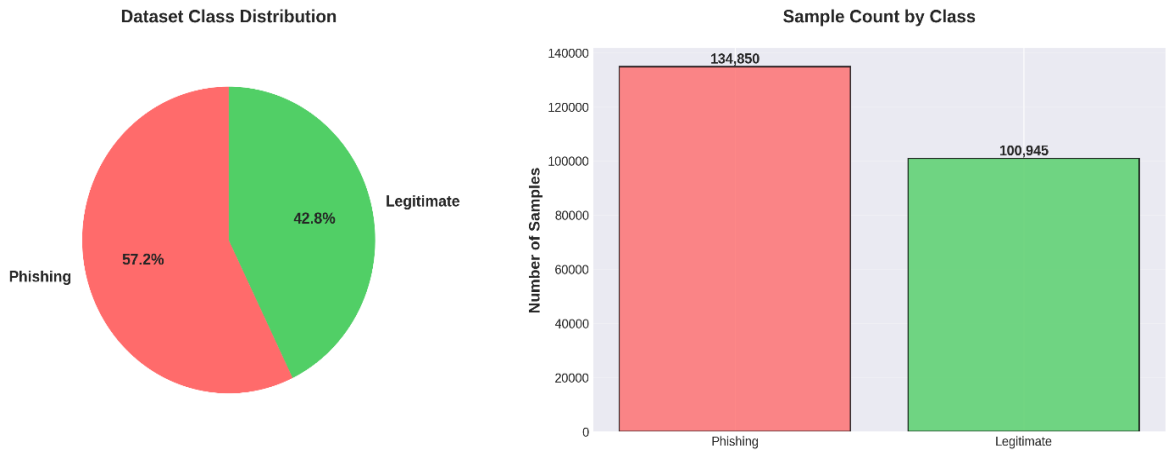

Figure 1. Class Distribution

2.2 Feature Extraction

During the feature extraction phase, this study intentionally avoided extracting webpage content (e.g., HTML structures and JavaScript execution). Content extraction is highly memory-intensive and significantly increases inference time, making it unsuitable for real-time client-side implementation [12] Instead, this system relies on structural features extracted purely from the Uniform Resource Locator (URL). During this extraction process, attributes such as IsHTTPS, CharContinuationRate, NoOfOtherSpecialCharsInURL, and NoOfDigitsInURL were identified as the most significantly contributing features.

Based on the initial 56 features available in the original dataset, the system specifically filtered and reduced the data dimensions to only 22 URL-based lexical features. This URL-only approach is supported by recent literature, demonstrating that a combination of lexical features, special characters, and URL structural probabilities is sufficient to achieve superior classification accuracy in phishing link detection[14], [22]. The selection of the 22 features in this study focused on the anomaly indicators most frequently encountered in fraudulent links, ranging from URL length characteristics and character composition patterns to the legitimacy probability of the Top-Level Domain (TLD). The details of the 22 utilized features are presented in Table 1.

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 4, Juni 2026
e-ISSN 2685-998X
DOI 10.30865/json.v7i4.9777

Table 1. Details of 22 Extracted URL Features

No	Feature Name	Category	Data Type	Description
1	URLLength	Length-based	Numeric	Calculates the total character length of the entire URL
2	DomainLength	Length-based	Numeric	Calculates the length of characters in the domain name.
3	IsDomainIP	Domain-based	Boolean	Detects whether the URL uses an IP address instead of a standard domain name (1 = Yes, 0 = No).
4	TLDLength	Length-based	Numeric	Calculates the character length of the Top-Level Domain (TLD).
5	NoOfSubDomain	Domain-based	Numeric	Counts the number of subdomains in the URL.
6	HasObfuscation	Obfuscation-based	Boolean	Marks whether the URL uses obfuscation techniques to hide the original address.
7	NoOfObfuscatedChar	Obfuscation-based	Numeric	Counts the total number of intentionally obfuscated or encoded characters.
8	ObfuscationRatio	Obfuscation-based	Float	Calculates the percentage of obfuscated characters to the total URL length.
9	NoOfLettersInURL	Character-based	Numeric	Counts the number of alphabetical letters (a-z, A-Z) in the URL.
10	LetterRatioInURL	Character-based	Float	Calculates the percentage of alphabetical letters in the total URL length.
11	NoOfDigitsInURL	Character-based	Numeric	Counts the number of digits (0-9) in the URL.
12	DigitRatioInURL	Character-based	Float	Calculates the percentage of digits in the total URL length.
13	NoOfEqualsInURL	Special Character	Numeric	Counts the frequency of the equals symbol ('=') in the URL.
14	NoOfQMarkInURL	Special Character	Numeric	Counts the frequency of the question mark symbol ('?') in the URL.
15	NoOfAmpersandInURL	Special Character	Numeric	Counts the frequency of the ampersand symbol ('&') in the URL.
16	NoOfOtherSpecialCharsInURL	Special Character	Numeric	Counts the occurrence of other special characters in the URL.
17	SpecialCharRatioInURL	Special Character	Float	Counts the occurrence of other special characters in the URL.
18	IsHTTPS	Protocol-based	Boolean	Identifies whether the URL uses the secure HTTPS protocol (1 = Yes, 0 = No).
19	URLSimilarityIndex	Similarity-based	Float	Measures the structural similarity index of the URL with URLs from legitimate web domains.
20	CharContinuationRate	Pattern-based	Float	Measures the longest sequence rate of alphabetical, digit, and consecutive character patterns.
21	TLDLegitimateProb	Probability-based	Float	The empirical probability value of the used TLD compared to the legitimate TLD list.
22	URLCharProb	Probability-based	Float	The probability value is based on the character occurrence distribution across URLs.

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 4, Juni 2026
e-ISSN 2685-998X
DOI 10.30865/json.v7i4.9777

Note: Minor typographical adjustments have been made to the feature names in this manuscript for readability and standard English spelling (e.g., correcting 'Degit' to 'Digit' and 'Spacial' to 'Special'), whereas their original identifiers in the raw PhiUSIIL dataset remain intact.

The numerical features subsequently underwent a preprocessing stage using the standard scaling technique (StandardScaler) to achieve a uniform scale distribution. This normalization step is crucial to minimize redundancy (collinearity) among features, prevent the dominance of large-valued features, and avoid weight bias in the Logistic Regression algorithm during the training phase.

2.3 Mathematical Formulation

The classification system employs a hybrid approach combining Logistic Regression (LR) and Random Forest (RF) to strike an optimal balance between low computational expense for a browser extension and a high capability to capture non-linear feature patterns [21]

2.3.1 Logistic Regression

Logistic Regression acts as the frontline detection model due to its exceptionally fast inference time. This algorithm predicts the probability of a URL being a phishing attack by mapping a linear combination of the URL features into a value range of 0 to 1 using the Sigmoid activation function. The mathematical equation is shown in equation (1):

$$P(Y = 1|X) = \frac{1}{1 + e^{-(\beta_0+\beta_1 X_1+\beta_2 X_2+\cdots+\beta_n X_n)}} \quad (1)$$

Where $P(Y = 1|X)$ is the probability of the link being classified as phishing, β_0 is the intercept (bias) value, $\beta_1 \ldots \beta_n$ are the trained coefficient weights, and $X_1 \ldots X_n$ represent the 22 normalized URL features.

2.3.2 Random Forest

To minimize the false-negative rate, Random Forest is utilized as an advanced verification layer. At each tree node, the model utilizes the Gini Impurity metric to evaluate optimal feature separation [[11]. The Gini Impurity metric is shown in equation (2):

$$Gini = 1 - \sum p_i^2 \quad (2)$$

Where p_i is the probability of a sample belonging to a specific class at that node. The final classification is determined through a majority voting mechanism from all trees (C) in the forest (m) shown in equation (3):

$$\hat{y} = mode(C_1(X), C_2(X), \ldots, C_m(X) \quad (3)$$

2.3.2 Hybrid Scoring Mechanism

To generate the final decision on the client-side, the system combines the confidence probability scores from both LR and RF predictions into a Hybrid Averaging mechanism shown in equation (4):

$$Score_{hybrid} = \frac{P_{LR} + P_{RF}}{2} \times 100\% \quad (4)$$

Where P_{LR} is the threat probability percentage from the LR model, and P_{RF} is the confidence percentage from the RF model. Based on the system parameters, the detection results are categorized into three real-time warning classes shown in Table 2.

Table 2. Risk Indicator Classification

No	Class	Score Range	Class Description
1	Safe	0% - 25%	The extension will not initiate blocking and provides a green indicator
2	Caution	26% - 50%	The extension displays a yellow indicator warning that the link exhibits suspicious characteristics, and the user is advised against entering sensitive data.
3	High Risk	51% - 100%	The extension displays a bright red blocking screen with a severe warning that the link is strongly indicated as phishing.

2.4 Model Evaluation

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 4, Juni 2026
e-ISSN 2685-998X
DOI 10.30865/json.v7i4.9777

To ensure objective evaluation and avoid overfitting, the processed PhiUSIIL dataset was partitioned using a stratified train-test split technique, with a ratio of 80% for the training set and 20% for the testing set. To guarantee the stability and reliability of the model's performance, a 5-fold cross-validation evaluation method was applied to the training data. The entire algorithm initialization was configured using a fixed random seed value (random_state = 42).

In addition to static dataset evaluation, this research conducted real-world testing within the user's browser environment. This testing aims to measure the reliability of the Google Chrome extension in providing visual warnings when processing Live URLs in real-time.

3. RESULT AND DISCUSSION

In this study, the testing and implementation process of a machine learning-based phishing detection system applied in a web browser environment was conducted. The system was built to analyze link characteristics to distinguish between phishing and legitimate links.

3.1 Machine Learning Model Training

Model training was executed using a cross-validation approach to build a hybrid architecture. First, Logistic Regression was trained as a base learner to map the linear decision boundary between phishing and legitimate classes with high computational efficiency. Next, the Random Forest algorithm was trained on the same feature space to capture more complex non-linear feature interactions that might be missed by the linear model. The training process stability of both algorithms was rigorously evaluated through a cross-validation mechanism. This was to ensure that the model converged well in recognizing threat patterns and did not memorize the training data, which could trigger a severe overfitting phenomenon.

3.2 Model Performance Evaluation

Model performance evaluation was comprehensively conducted using standard classification metrics, namely Accuracy, Precision, Recall, F1-Score, and Area Under Curve (AUC). To ensure that the high accuracy achieved by the model was not caused by overfitting, stability testing was performed through a 5-fold cross-validation mechanism during the training phase. The visualization of the accuracy score distribution across iterations showed that the variance of data deviation was very small, proving the model's consistency when faced with different data portions, as shown in Figure 2.

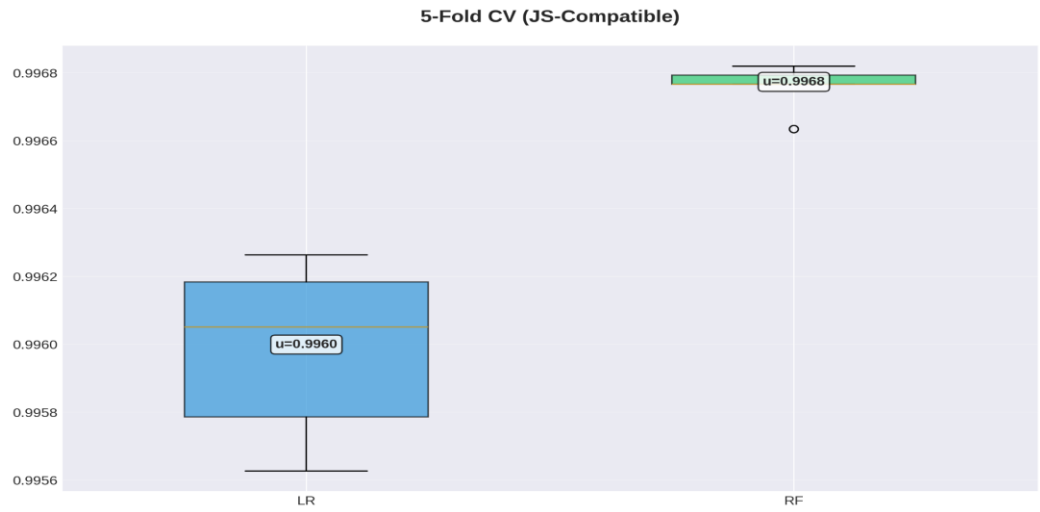

Figure 2. 5 Fold Cross-Validation Result

After the stability of the base model was validated, testing continued on 47,159 unseen testing set samples. A Confusion Matrix was used to map the absolute number of correct and incorrect predictions, including True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) shown in Figure 3.

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 4, Juni 2026
e-ISSN 2685-998X
DOI 10.30865/json.v7i4.9777

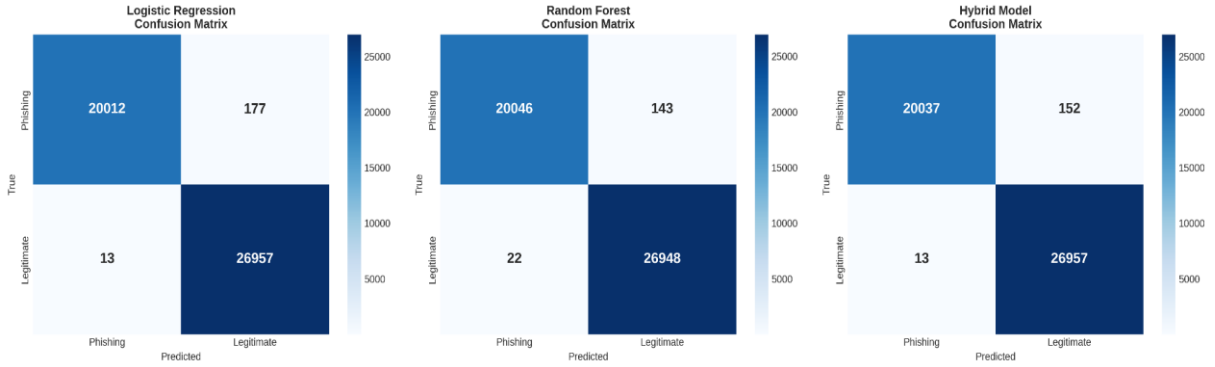

Figure 3. Confusion Matrix: (a) Logistic Regression,(b) Random Forest,(c) Hybrid Model

In the aspect of phishing detection, the Hybrid approach successfully detected 20,037 malicious links accurately (TP) and produced 152 detection failures (FN). Although independently, Random Forest recorded a marginally higher Recall rate (FN = 143), the Hybrid model proved to be the most robust "middle-ground" architecture. The average scoring mechanism (Hybrid Averaging) successfully suppressed the main weakness of Logistic Regression (which produced 177 FNs), while simultaneously distributing the inference workload so it is not as expensive as pure Random Forest execution. These characteristics make the Hybrid architecture highly ideal to be executed seamlessly, lightly, and in real-time in a client-side environment. Based on these absolute Confusion Matrix values, the models' overall performance was mathematically translated into percentage scores for Accuracy, Precision, Recall, and F1-Score. The comprehensive summary and comparative analysis of these metrics are elaborated further in the Discussion section.

3.3 System Implementation

After being evaluated, the Logistic Regression and Random Forest models were converted into TensorFlow.js and JSON formats to be directly integrated into the Google Chrome extension. Through the JavaScript programming language, the hybrid inference and decision-making mechanisms were fully executed on the client-side. The system works in real-time by intercepting URLs from the user's active tab, extracting its 22 features locally, and calculating the probability score without needing to send browsing history to third-party servers. The details of the system architecture will be shown in Figure 4.

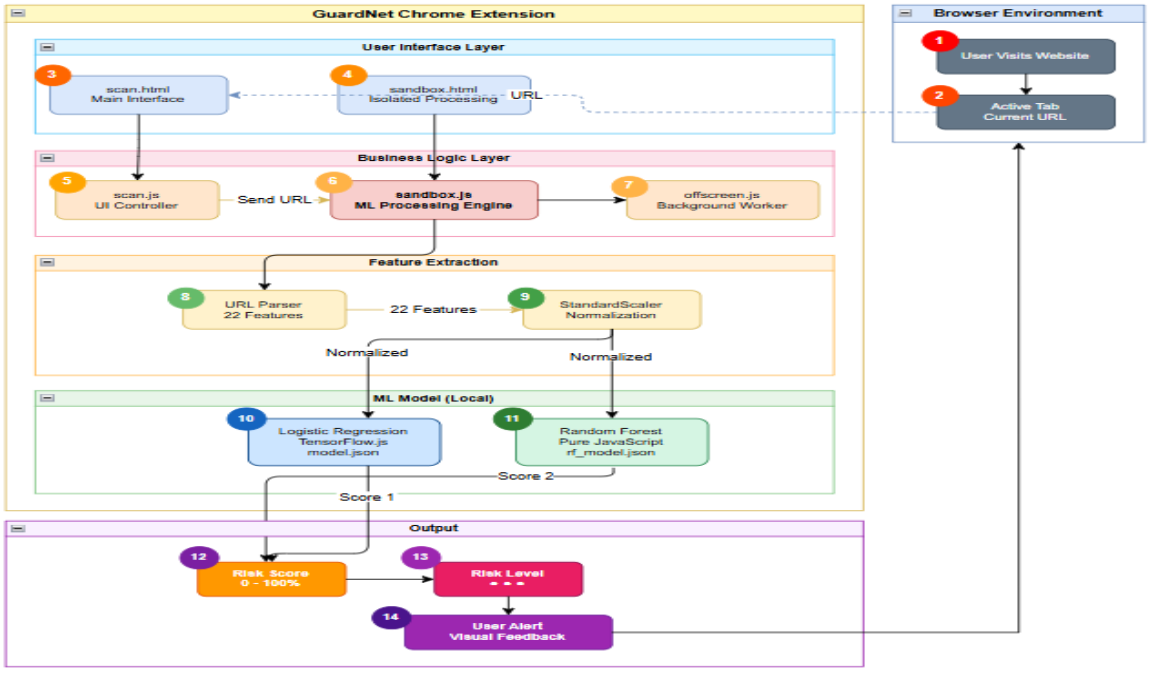

Figure 4. The Details Of System Architecture

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 4, Juni 2026
e-ISSN 2685-998X
DOI 10.30865/json.v7i4.9777

To validate the operational reliability of the proposed model from an end-user perspective, a live implementation and functional testing were conducted directly within the browser environment. During this live implementation phase, the system was actively tested against 100 live URL samples. The extension's User Interface then converts the hybrid classification results into three intuitive visual risk indicators: a green indicator for Safe status (0-25% score), a yellow indicator for Caution (26-50% score), and a bright red blocking screen for High Risk warning (51-100% score). The live implementation results proved that the extension achieved 92% accuracy rate in classifying the live samples, successfully presenting instant protection that is responsive and easily understood by users, without causing latency that disrupts web browsing smoothness. The result of the extension is shown in Figure 5.

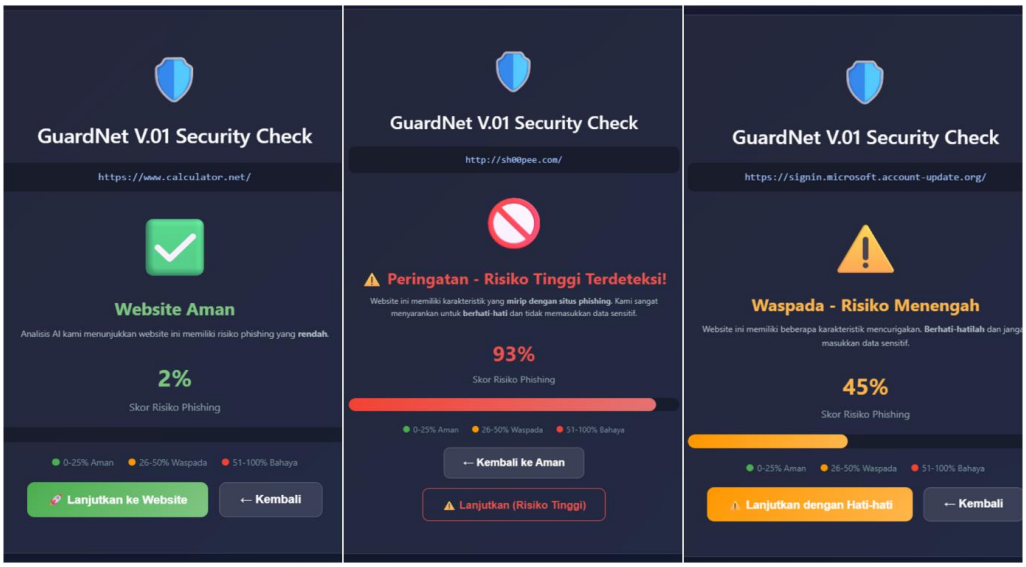

Figure 5. Visual Risk Indicators and Extension Results

3.4 Discussion

The strategic integration of Logistic Regression (LR) and Random Forest (RF) into a hybrid architecture demonstrates a significant advantage for client-side phishing detection. The 5-fold cross-validation results confirm high operational stability, with Random Forest achieving a slightly higher mean stability (μ=0.9968) compared to Logistic Regression (μ=0.9960). This minimal variance proves that the 22 selected lexical URL features are highly discriminatory and allow the models to generalize threat patterns effectively without overfitting to the training data

The core strength of the proposed system lies in its ability to resolve the critical trade-off between user accessibility and security. In a browser extension environment, a False Positive (FP)—wrongfully blocking a legitimate site—severely degrades the user experience, while a False Negative (FN)—missing a phishing link—poses a direct security threat. The specific performance metrics of the standalone models compared to the hybrid approach are summarized in Table 3.

Table 3. Summary of Model Performance Matrix

No	Model	TP	TN	FP	FN	Accuracy	Precision	Recall	F1-Score
1	Logistic Regression	20012	26957	13	177	99.60%	99.94%	99.12%	99.53%
2	Random Forest	20046	26948	22	143	99.65%	99.89%	99.29%	99.59%
3	Hybrid Machine Learning	20037	26957	13	152	99.65%	99.94%	99.25%	99.59%

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 4, Juni 2026
e-ISSN 2685-998X
DOI 10.30865/json.v7i4.9777

The standalone Logistic Regression model is highly conservative, producing only 13 FPs but missing 177 phishing threats. Conversely, while Random Forest improves threat detection, it results in a higher FP rate of 22. As detailed in Table 3, the Hybrid mechanism successfully bridges this gap, inheriting LR's near-perfect Precision (99.94%) while boosting Recall to 99.25%. By achieving an outstanding F1-Score of 99.59% and 99.65% Accuracy, the model mathematically resolves the trade-off between user accessibility and security.

Furthermore, the live implementation on 100 real-world URL samples validated the extension's functional reliability, achieving 92% operational accuracy. Compared to recent server-side Deep Learning models that achieve accuracy exceeding 99% but suffer from high computational latency and privacy risks, our hybrid approach delivers a comparable 99.65% accuracy entirely locally. By executing inference purely on the client-side without transmitting user data, it eliminates network delays, providing a highly responsive and privacy-preserving defense layer.

Despite these advantages, by strictly avoiding HTML content extraction to maintain fast inference speeds, the system relies entirely on lexical URL features. Consequently, the model may struggle with highly obfuscated pages hosted on compromised legitimate domains or sophisticated URL shorteners, presenting a critical area for future optimization

4. CONCLUSION

This study has successfully implemented a real-time, client-side phishing detection system through a Google Chrome extension. By integrating Logistic Regression and Random Forest into a hybrid architecture, the system achieves an optimal balance between computational efficiency and robust pattern recognition using only 22 lexical URL features. Stability evaluation via 5-fold cross-validation yielded high mean performance scores $\mu = 0.9960$ for Logistic Regression and $\mu = 0.9968$ for Random Forest, confirming the model's reliability and resistance to overfitting.The hybrid approach effectively resolved the critical trade-off between user accessibility and security. Experimental results on the PhiUSIIL dataset showed the model successfully identified 20,037 phishing links (TP) and 26,957 legitimate URLs (TN), while suppressing false alarms to only 13 False Positives and reducing missed threats to 152 False Negatives. Furthermore, functional testing on 100 live URL samples validated the system's practical utility with 92% operational accuracy. Although minor false-positive (6%) and false-negative (2%) rates were observed in dynamic environments, the system provides a lightweight, privacy-preserving defense layer that eliminates the network latency inherent in server-side heuristic analysis.Despite these promising outcomes, the system's reliance on structural URL features remains a limitation against highly obfuscated short-link techniques. Future research should focus on integrating lightweight contextual analysis from the Document Object Model (DOM) and evaluating the model's resilience against adversarial machine learning to further strengthen defenses against emerging zero-day cyber threats

REFERENCES

[1] F. N. Motlagh, M. Hajizadeh, M. Majd, P. Najafi, F. Cheng, and C. Meinel, "Large language models in Cybersecurity: Sate-of-the-Art," *arXiv (Cornell University)*, Jan. 2024, doi: 10.48550/arXiv.2402.00891.

[2] M. Mijwil, I. E. Salem, and M. M. Ismaeel, "The Significance of machine learning and deep learning techniques in Cybersecurity: A Comprehensive review," *Iraqi Journal for Computer Science and Mathematics*, pp. 87-101, Jan. 2023, doi: 10.52866/ijcsm.2023.01.01.008.

[3] A. Safi and S. Singh, "A systematic literature review on phishing website detection techniques," *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 2, pp. 590–611, Feb. 2023, doi: 10.1016/j.jksuci.2023.01.004.

[4] O. K. Sahingoz, E. Buber, and E. Kugu, "DEPHIDES: Deep Learning Based Phishing Detection System," *IEEE Access*, vol. 12, pp. 8052–8070, 2024, doi: 10.1109/ACCESS.2024.3352629.

[5] R. MRM, N. F.A.M, S. A.M, and S. K.A.S, "A comparative analysis of machine learning models for URL-Based phishing detection," *Research Square*, Apr. 15, 2025. doi: 10.21203/rs.3.rs-6439154/v1.

[6] B. C. Ujah-Ogbuagu, O. N. Akande, and E. Ogbuju, "A hybrid deep learning technique for spoofing website URL detection in real-time applications," *Journal of Electrical Systems and Information Technology*, vol. 11, no. 1, Jan. 2024, doi: 10.1186/s43067-023-00128-8.

[7] G. S. Nayak, B. Muniyal, and M. C. Belavagi, "Enhancing Phishing Detection: a machine learning approach with feature selection and deep learning models," *IEEE Access*, vol. 13, pp. 33308–33320, Jan. 2025, doi: 10.1109/ACCESS.2025.3543738.

[8] Q. E. U. Haq, M. H. Faheem, and I. Ahmad, "Detecting Phishing URLs based on a deep learning approach to prevent Cyber-Attacks," *Applied Sciences (Switzerland)*, vol. 14, no. 22, Nov. 2024, doi: 10.3390/app142210086.

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 4, Juni 2026
e-ISSN 2685-998X
DOI 10.30865/json.v7i4.9777

[9] U. Daniel, E. Bartholomew, and F. Egbono, "Phishing URL Attack Detection using Logistic Regression and Convolutional Neural Network,' *International Journal of Computer Applications*, vol.187, no. 1, pp. 8-14, May 2025, doi: 10.5120/ijca2025924611.

[10] R. Yang, K. Zheng, B. Wu, C. Wu, and X. Wang, "Phishing website detection based on deep convolutional neural network and random forest ensemble learning," *Sensors*, vol. 21, no. 24, Dec. 2021, doi: 10.3390/s21248281.

[11] D. Kalla and S. Kuraku, "Phishing Website URL's Detection Using NLP and machine learning techniques," *Journal on Artificial Intelligence*, vol. 5, no. 0, pp. 145–162, Jan. 2023, doi: 10.32604/jai.2023.043366.

[12] A. Almomani *et al.*, "Phishing website detection with semantic features based on machine learning classifiers,"*International Journal on Semantic Web and Innformation Systems*, vol. 18, no. 1, Feb. 2022, doi: 10.4018/ijswis.297032.

[13] W. Purba, M. Turnip, and E. Indra, "Algorithm analysis comparison of naïve bayes and logistic regression methods for predicting diabetes," Jun. 01, 2024. https://internetworkingindonesia.org/index.php/iij/article/view/33

[14] V. Vajrobol, B. B. Gupta, and A. Gaurav, "Mutual information based logistic regression for phishing URL detection," *Cyber Security and Applications*, vol. 2, p. 100044, Jan. 2024, doi: 10.1016/j.csa.2024.100044.

[15] F. Felix, D. Sitanggang, Y. Laia, A. -, M. Radhi, and E. S. Barus, "Application of Data Mining using the random forest method to predict heart disease", *JUSIKOM PRIMA*, vol. 7, no. 2, pp. 35-47, Feb. 2024.

[16] N. F. Almujahid, M. A. Haq, and M. Alshehri, "Comparative evaluation of machine learning algorithms for phishing site detection," *PeerJ Computer Science*, vol. 10, p. e2131, Jun 2024, doi: 10.7717/peerj-cs.2131.

[17] E. Kocyigit, M. Korkmaz, O. K. Sahingoz, and B. Diri, "Enhanced feature selection using genetic algorithm for Machine-Learning-Based phishing URL detection," *Applied Sciences*, vol. 14, no. 14, p. 6081, Jul. 2024, doi: 10.3390/app14146081.

[18] M. Ahmed *et al.*, "PhishCatcher: Client-Side Defense Against Web Spoofing Attacks Using Machine Learning," *IEEE Access*, vol. 11, pp. 61249–61263, 2023, doi: 10.1109/ACCESS.2023.3287226.

[19] Sagar Aghera, "Enhancing Predictive Accuracy in Phishing Attack Detection: A Study on the impact of Collinearity and Feature Selection in ML-based Logistic Regression Models", *Int J Intell Syst Appl Eng*, vol. 12, no. 4, pp. 723-728, Jun. 2024.

[20] V. A. Onih, "Phishing Detection using Machine Learning: a model development and integration," *International Journal of Scientific and Management Research*, vol. 07, no. 04, pp. 27–63, Jan. 2024, doi: 10.37502/ijsmr.2024.7403.

[21] M. Sannigrahi and R. Thandeeswaran, "Predictive analysis of Network-Based attacks by hybrid machine learning algorithms utilizing bayesian optimization, logistic regression, and random forest algorithm," *IEEE Access*, vol. 12, pp. 142721–142732, Jan. 2024, doi: 10.1109/ACCESS.2024.3464866.

[22] M. Tamal, "Phishing Detection Dataset," *Data Archiving and Networked Services (DANS)*. Jun. 07, 2023. doi: 10.17632/6tm2d6sz7p.1.