



Modern Phishing URL Detection Using Feature Selection and Comparative Classification Models

M Teguh Prastyo¹, Febri Vahlevie²

^{1,2} Magister of Informatics, Institut Informatika dan Bisnis Darmajaya, Indonesia

ARTICLE INFO	ABSTRACT
Article history: Received May 25, 2026 Revised Jun 26, 2026 Accepted Jun 30, 2026 Keywords: phishing URL detection; feature selection; Information Gain; comparative classification; cybersecurity.	Phishing URLs remain a critical cybersecurity threat because attackers increasingly exploit domain similarity, webpage imitation, and structural manipulation to deceive users and bypass conventional blacklist-based detection. This study proposes an engineering-oriented phishing URL detection pipeline using feature selection and comparative classification models implemented in RapidMiner. The PhiUSIIL Phishing URL Dataset was used, and after preprocessing, 234,903 URL records were retained, consisting of 134,834 legitimate URLs and 100,069 phishing URLs. Non-predictive attributes were removed, invalid target labels were filtered, missing predictor values were handled, and the target label was transformed into a binominal class, where phishing was treated as the positive class. Information Gain was applied to identify the most discriminative attributes, and the top-20 features were used for model comparison. Five classification models were evaluated using stratified 10-fold cross-validation: Decision Tree, Random Forest, Naive Bayes, Logistic Regression, and Gradient Boosted Trees. The results show that all models achieved accuracy above 99.95%, indicating strong class separability within the selected-feature scenario. Random Forest produced the most balanced performance, achieving 100.00% accuracy, 100.00% precision, 100.00% recall, 100.00% F1-score, and AUC of 1.000, with only three phishing URLs misclassified as legitimate. The findings demonstrate that selected URL similarity and webpage structural features can support efficient and interpretable phishing detection. However, the near-perfect performance should be interpreted as strong internal validation, and future work should include external dataset validation and ablation testing of dominant features.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Febri Vahlevie,
Master of Informatics, IIB Darmajaya,
Jl. Zainal Abidin Pagar Alam No. 93, Gedong Meneng, Rajabasa, Bandar Lampung City, Lampung 35142,
Indonesia.
Email: febri.mahasiswa@darmajaya.ac.id

INTRODUCTION

Phishing URL detection has become a critical engineering problem in modern cybersecurity because malicious actors increasingly exploit web infrastructure, domain similarity, and content imitation to deceive users into submitting credentials or sensitive information. Unlike traditional malware-based attacks, phishing does not always require system exploitation; instead, it manipulates human trust through visually and structurally convincing

web pages [1]. From a system-engineering perspective, phishing URL detection must therefore operate as an early-layer defensive mechanism capable of classifying URLs before users interact with harmful pages. Conventional blacklist-based mechanisms remain useful but are inherently reactive and limited against newly generated, short-lived, or obfuscated phishing URLs. Recent studies indicate that machine learning-based detection provides a more adaptive approach by learning discriminative patterns from lexical URL structures, domain properties, webpage source-code attributes, and content-based indicators [2].

The state-of-the-art in phishing detection has evolved from rule-based and heuristic techniques toward data-driven classification models that exploit heterogeneous feature representations [3]. Classical machine learning algorithms such as Decision Tree, Logistic Regression, Naive Bayes, Support Vector Machine, and Random Forest have been widely used because they are computationally feasible, interpretable to varying degrees, and suitable for tabular cybersecurity datasets. Meanwhile, deep learning approaches have attempted to learn representations directly from raw URLs and HTML content, reducing dependency on handcrafted features but increasing computational complexity and deployment cost [4]. Recent phishing detection studies also emphasize the importance of combining URL-based features with webpage content characteristics because phishing behavior is not always captured by URL syntax alone. Therefore, practical detection systems must balance predictive performance, computational efficiency, interpretability, and deployability within real-time security architectures [5].

Despite the high accuracy reported in previous studies, several research gaps remain unresolved. First, many studies prioritize model performance without sufficiently analyzing whether a reduced subset of features can maintain comparable detection capability. This is important because feature-heavy models may increase preprocessing overhead, memory consumption, and inference latency in real-world deployment. Second, comparative model evaluation is often presented without adequate interpretability analysis, limiting the ability of security engineers to understand why a URL is classified as phishing or legitimate. Third, phishing datasets may contain dominant features that strongly separate classes, producing optimistic results if model validation is not carefully designed. Recent work also highlights that phishing detection features can be dataset-dependent, meaning that high performance on one dataset does not always guarantee generalization across different phishing environments [6]. These issues indicate the need for a systematic evaluation framework that integrates model comparison, feature selection, interpretability, and robust validation.

This study addresses the above gaps by proposing a structured machine learning evaluation framework for Modern Phishing URL Detection Using Feature Selection and Comparative Classification Models. The research uses the PhiUSIIL Phishing URL Dataset, which consists of 134,850 legitimate URLs and 100,945 phishing URLs, with features extracted from URL structures and webpage source code, including derived attributes such as CharContinuationRate, URLTitleMatchScore, URLCharProb, and TLDLegitimateProb [2]. The novelty of this study lies in the integration of four components: comparative classification modeling, feature selection, interpretability-oriented feature analysis, and 10-fold cross-validation. Instead of merely identifying the highest accuracy model, this study evaluates how selected features influence classification effectiveness and how dominant attributes explain phishing behavior from a cybersecurity engineering perspective [7].

The objectives of this study are fourfold. First, this research compares several machine learning algorithms, including Decision Tree, Random Forest, Naive Bayes, Logistic Regression, and Gradient Boosted Trees, for phishing URL classification. Second, it applies feature selection to identify the most influential attributes and assess whether reduced feature subsets can preserve model performance. Third, it evaluates models using accuracy, precision, recall, F1-score, AUC, and confusion matrix under 10-fold cross-validation, with

particular emphasis on phishing recall because false negatives represent phishing URLs incorrectly classified as legitimate. Fourth, it interprets dominant features to explain their technical relevance to phishing detection and their implications for practical deployment. The expected contribution is a compact, interpretable, and engineering-oriented detection framework that can support real-time cybersecurity systems such as browser extensions, email filtering engines, network security gateways, and institutional web protection platforms.

RESEARCH METHODS

System Architecture

This study adopts an experimental machine learning architecture for modern phishing URL detection. The proposed system is designed as a structured classification pipeline consisting of five main layers: data acquisition, preprocessing, feature selection, classification modeling, and performance evaluation [8]. The input layer receives the PhiUSIIL Phishing URL Dataset, which contains URL-based and webpage source-code-based attributes. The preprocessing layer transforms the raw dataset into a structured analytical format by assigning the target label, removing non-predictive attributes, checking missing values, and validating attribute types. The feature selection layer identifies the most discriminative attributes to reduce computational complexity and improve model interpretability [9]. The classification layer compares several supervised machine learning algorithms, while the evaluation layer measures model performance using 10-fold cross-validation.

The system architecture is illustrated conceptually as follows:

$$D_{\text{raw}} \rightarrow D_{\text{prep}} \rightarrow F_{\text{selected}} \rightarrow M_i \rightarrow E_i$$

where D_{raw} denotes the original dataset, D_{prep} denotes the preprocessed dataset, F_{selected} represents the selected feature subset, M_i represents the (i) – th classification model, and (E_i) denotes the evaluation result of each model.

Table 1. Proposed System Architecture

Layer	Process	Output
Data acquisition	Import PhiUSIIL dataset into RapidMiner	Raw URL dataset
Preprocessing	Set label, remove non-predictive attributes, validate data type	Clean dataset
Feature selection	Rank and select dominant attributes	Selected feature subset
Classification	Train comparative machine learning models	Prediction model
Validation	Apply 10-fold cross-validation	Generalized model performance
Evaluation	Compute accuracy, precision, recall, F1-score, AUC, and confusion matrix	Comparative performance result

This architecture follows an engineering-oriented approach because the model is not only evaluated based on prediction accuracy but also based on computational efficiency and interpretability. Such a structure is relevant for real-world implementation in browser security modules, URL filtering engines, email gateways, and institutional cybersecurity monitoring systems.

Dataset Description

The dataset used in this study is the PhiUSIIL Phishing URL Dataset, which contains a substantial collection of legitimate and phishing URLs. The dataset consists of 235,795 URL records, including 134,850 legitimate URLs and 100,945 phishing URLs. The features were extracted from URL structures and webpage source code [10]. Several attributes are derived from existing features, such as CharContinuationRate, URLTitleMatchScore, URLCharProb, and TLDLegitimateProb. These features provide technical indicators related to URL lexical

patterns, domain characteristics, similarity score, webpage structure, and source-code behavior.

The target class is represented by a binary label, where phishing and legitimate URLs are treated as two classification classes. In the implementation stage, the original numeric label is transformed into a nominal class to improve interpretability in RapidMiner. The phishing class is treated as the positive class because, in cybersecurity applications, the most critical error is a phishing URL being incorrectly classified as legitimate.

Table 2. Dataset Characteristics

Component	Description
Dataset	PhiUSIIL Phishing URL Dataset
Total records	235,795 URLs
Legitimate URLs	134,850
Phishing URLs	100,945
Total attributes	56 attributes
Feature source	URL structure and webpage source code
Original label	0 and 1
Classification task	Binary classification
Tool	RapidMiner

The dataset includes several non-predictive attributes such as FILENAME, URL, Domain, and Title. These attributes are removed before model training because they function as identifiers or raw textual fields rather than structured predictive attributes. The remaining numerical and binary attributes are used as input features for the classification models [11].

Algorithm Design

The algorithmic design of this study consists of two scenarios: classification using full features and classification using selected features. The full-feature scenario evaluates model performance using all structured predictive attributes after preprocessing. The selected-feature scenario evaluates whether a reduced subset of dominant features can maintain comparable performance [12]. This design is important for determining whether phishing URL detection can be implemented using a lighter and more efficient model.

Let the dataset be defined as:

$$D = (x_i, y_i)_{i=1}^n$$

where (x_i) represents the feature vector of the $(i) - \text{th}$ URL, $(y_i \in \text{phishing, legitimate})$ represents the class label, and (n) is the total number of URL records. Each feature vector consists of multiple attributes:

$$x_i = f_1, f_2, f_3, \dots, f_m$$

where (m) denotes the number of predictive features. Feature selection is applied to assign a relevance weight to each feature. In this study, Information Gain is recommended as the primary feature selection method because it measures the contribution of each attribute in reducing class uncertainty. Information Gain is defined as:

$$IG(Y, X) = H(Y) - H(Y|X)$$

where $(H(Y))$ is the entropy of the class label and $(H(Y|X))$ is the conditional entropy of the class after observing feature (X) . Entropy is computed as:

$$H(Y) = - \sum_{c \in C} p(c) \log_2 p(c)$$

where $(p(c))$ is the probability of class (c) , and (C) is the set of class labels. Features with higher Information Gain are considered more relevant for distinguishing phishing URLs from legitimate URLs.

Table 3. Experimental Scenarios

Scenario	Feature Setting	Purpose
Scenario 1	Full features	Evaluate baseline model performance using all predictive attributes
Scenario 2	Selected features	Evaluate model efficiency and performance after feature selection

The classification models compared in this study include Decision Tree, Random Forest, Naive Bayes, Logistic Regression, and Gradient Boosted Trees. Decision Tree is used because it produces rule-based structures that are easy to interpret. Random Forest is used because it improves generalization by combining multiple decision trees. Naive Bayes is included as a probabilistic baseline model [13]. Logistic Regression is used as a linear baseline classifier, while Gradient Boosted Trees is included as a boosting-based ensemble model capable of capturing complex feature interactions.

Table 4. Classification Algorithms

Algorithm	Model Type	Main Role in Experiment
Decision Tree	Rule-based classifier	Interpretable baseline model
Random Forest	Bagging ensemble	Robust non-linear classifier
Naive Bayes	Probabilistic classifier	Simple statistical baseline
Logistic Regression	Linear classifier	Linear decision boundary comparison
Gradient Boosted Trees	Boosting ensemble	High-performance ensemble comparison

Implementation Steps in RapidMiner

The experiment is implemented using RapidMiner because it supports visual machine learning workflows, cross-validation, feature selection, and comparative model evaluation. The implementation begins by importing the CSV dataset using the Read CSV operator. After importing the dataset, the Set Role operator is used to define the class label [14]. If needed, Generate Attributes is applied to transform the original binary label into nominal labels, namely phishing and legitimate.

The preprocessing stage removes non-predictive attributes using the Select Attributes operator. The attributes removed include FILENAME, URL, Domain, and Title. The remaining structured attributes are retained as predictor variables. After preprocessing, feature selection is conducted using Weight by Information Gain, followed by Select by Weights to select the most influential features. The selected feature subset is then passed into the model training stage.

The model validation process is implemented using the Cross Validation operator with 10 folds and stratified sampling. In each fold, nine subsets are used for training and one subset is used for testing. The process is repeated until each subset has been used once as testing data. The training subprocess contains the selected classification algorithm, while the testing subprocess contains Apply Model and Performance (Binominal Classification).

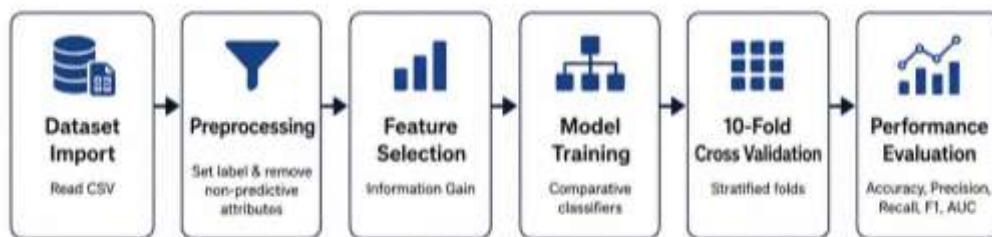


Figure 1. RapidMiner Implementation Workflow

To prevent optimistic bias, feature selection should be placed inside the training subprocess of cross-validation when evaluating selected-feature scenarios. This ensures that feature weights are computed only from training data in each fold, rather than from the entire dataset before validation.

Evaluation Metrics

Model performance is evaluated using accuracy, precision, recall, F1-score, AUC, and confusion matrix. In phishing detection, recall for the phishing class is particularly important because false negatives indicate phishing URLs that are incorrectly classified as legitimate. Such errors are highly critical because they may allow malicious URLs to bypass the detection system. The confusion matrix is defined as follows:

Table 4. Confusion Matrix for Phishing URL Detection

Actual / Predicted	Predicted Phishing	Predicted Legitimate
Actual Phishing	True Positive (TP)	False Negative (FN)
Actual Legitimate	False Positive (FP)	True Negative (TN)

Accuracy measures the overall proportion of correctly classified URLs:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision measures the proportion of predicted phishing URLs that are truly phishing:

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall measures the ability of the model to detect actual phishing URLs:

$$\text{Recall} = \frac{TP}{TP + FN}$$

F1-score represents the harmonic mean between precision and recall:

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

The Area Under the Curve (AUC) is used to evaluate the ability of the model to separate phishing and legitimate classes across different classification thresholds [15]. A higher AUC indicates stronger discriminative capability.

The final model is selected based not only on the highest accuracy but also on phishing recall, F1-score, AUC, and confusion matrix analysis. This evaluation strategy ensures that the selected model is technically accurate, practically reliable, and suitable for cybersecurity-oriented deployment.

RESULTS AND DISCUSSION

Preprocessing and Dataset Distribution Results

After the dataset was imported into RapidMiner, several preprocessing stages were performed before model development. The original PhiUSIIL dataset consisted of 235,795 URL records and 56 attributes. Non-predictive textual attributes, including FILENAME, URL, Domain, and Title, were removed because they mainly function as identifiers or raw textual fields rather than structured predictor variables. Records with missing target labels were also removed to ensure that all examples used in supervised learning had valid class information. After filtering invalid labels, the final dataset used for model training and evaluation contained 234,903 URL records.

The numerical target label was transformed into a binominal class attribute. In RapidMiner, the transformed label was represented as false and true, where false denotes legitimate URLs and true denotes phishing URLs. The final class distribution consisted of 134,834 legitimate URLs and 100,069 phishing URLs. This distribution indicates that the dataset remained relatively balanced after preprocessing, with legitimate URLs representing 57.40% of the dataset and phishing URLs representing 42.60%.

Table 5. Dataset Distribution After Preprocessing

Class Label	Actual Class	Number of Records	Percentage
false	Legitimate	134,834	57.40%
true	Phishing	100,069	42.60%

Total	–	234,903	100.00%
--------------	---	---------	---------

The preprocessing stage successfully produced a clean dataset with valid class labels and complete predictor attributes. Missing values in predictor variables were handled using the Replace Missing Values operator before feature selection and model evaluation. Therefore, the resulting dataset was suitable for supervised binary classification.

Feature Selection Results

Feature selection was performed using Information Gain to identify the most discriminative attributes for phishing URL classification. The top-ranked features indicate that phishing detection in this dataset is strongly influenced by URL similarity, webpage structure, source-code characteristics, and security-related attributes. The most influential feature was URLSimilarityIndex, which obtained the highest normalized Information Gain weight of 1.000. This result shows that URL similarity is the strongest discriminator between phishing and legitimate URLs in the selected-feature scenario.

Table 6. Top Features Based on Information Gain

Rank	Feature	Information Gain Weight
1	URLSimilarityIndex	1.000
2	NoOfExternalRef	0.791
3	LineOfCode	0.758
4	NoOfSelfRef	0.757
5	NoOfImage	0.698
6	NoOfJS	0.655
7	HasSocialNet	0.581
8	NoOfCSS	0.580
9	HasCopyrightInfo	0.481
10	HasDescription	0.420
11	LargestLineLength	0.373
12	IsHTTPS	0.345
13	NoOfOtherSpecialCharsInURL	0.300
14	HasSubmitButton	0.284
15	DomainTitleMatchScore	0.277

The ranking shows that the dominant attributes are not limited to lexical URL characteristics. Several webpage structural features, such as NoOfExternalRef, LineOfCode, NoOfSelfRef, NoOfImage, NoOfJS, and NoOfCSS, also contributed strongly to classification. This indicates that the distinction between legitimate and phishing URLs in the dataset is represented by both URL-level and webpage-level feature patterns. Based on the Information Gain ranking, the top 20 features were selected for the comparative classification experiment.

Comparative Model Performance

Five machine learning algorithms were evaluated using the top-20 Information Gain features: Decision Tree, Random Forest, Naive Bayes, Logistic Regression, and Gradient Boosted Trees. Each model was evaluated using 10-fold cross-validation with stratified sampling. The phishing class was treated as the positive class because the most critical error in phishing URL detection is a phishing URL being classified as legitimate.

The comparative results are presented in Table VI. Overall, all evaluated models achieved very high classification performance, with accuracy values above 99.95%. Random Forest achieved the highest overall performance, with 100.00% accuracy, 100.00% precision, 100.00% recall, 100.00% F1-score, and an AUC value of 1.000. Naive Bayes also achieved 100.00% recall for the phishing class, although its precision was slightly lower at 99.89%. Logistic Regression obtained the lowest recall among the evaluated models, although its performance remained very high at 99.94%.

Table 7. Comparative Performance of Classification Models

Model	Accuracy	Precision	Recall	F1-score	AUC
Decision Tree	99.99%	100.00%	99.98%	99.99%	0.500
Random Forest	100.00%	100.00%	100.00%	100.00%	1.000
Naive Bayes	99.95%	99.89%	100.00%	99.94%	1.000
Logistic Regression	99.97%	99.98%	99.94%	99.96%	1.000
Gradient Boosted Trees	99.99%	100.00%	99.97%	99.98%	1.000

Based on accuracy and F1-score, Random Forest produced the strongest result among the evaluated models. Decision Tree and Gradient Boosted Trees also showed highly competitive performance, while Naive Bayes demonstrated strong phishing detection capability with perfect recall but slightly lower precision. Logistic Regression remained effective but produced the highest number of phishing false negatives among the evaluated models.

Confusion Matrix Analysis

To provide a more detailed quantitative assessment, the confusion matrix of each model was analyzed. In phishing detection, the false negative value is particularly important because it represents phishing URLs that are incorrectly predicted as legitimate. A lower false negative value indicates a stronger ability to prevent phishing URLs from bypassing the detection system.

Table 8. Confusion Matrix Summary of Evaluated Models

Model	TN	FP	FN	TP
Decision Tree	134,834	0	24	100,045
Random Forest	134,834	0	3	100,066
Naive Bayes	134,724	110	2	100,067
Logistic Regression	134,817	17	59	100,010
Gradient Boosted Trees	134,833	1	33	100,036

The Random Forest model produced only 3 false negatives and no false positives, indicating the most balanced error distribution. Naive Bayes produced the lowest false negative value, with only 2 phishing URLs misclassified as legitimate. However, it also produced 110 false positives, meaning that more legitimate URLs were incorrectly classified as phishing compared with the other models. Logistic Regression produced 59 false negatives, which was the highest among all evaluated models. Gradient Boosted Trees produced 33 false negatives and 1 false positive, while Decision Tree produced 24 false negatives and no false positives.

From the perspective of phishing detection, the false negative count is a critical indicator because it directly reflects the number of phishing URLs that may escape detection. The comparison shows that all models were able to suppress false negatives to a very low level. However, Random Forest provided the most favorable balance between false negative reduction and false positive control.

Quantitative Model Ranking

The models were ranked based on multiple criteria, including accuracy, phishing recall, F1-score, AUC, and confusion matrix behavior. Although Naive Bayes achieved the lowest false negative count, its false positive count was higher than that of Random Forest. Therefore, Random Forest can be considered the most stable model because it achieved near-perfect performance across all metrics while maintaining very low classification errors.

Table 9. Model Ranking Based on Phishing Detection Performance

Rank	Model	Main Quantitative Justification
1	Random Forest	Highest overall performance; only 3 FN and 0 FP
2	Naive Bayes	Lowest FN value, but higher FP count
3	Decision Tree	High precision and low FN with interpretable structure
4	Gradient Boosted Trees	Very high precision, but higher FN than Decision Tree

5	Logistic Regression	Strong performance, but highest FN among evaluated models
---	---------------------	---

The results indicate that ensemble-based and probabilistic models performed particularly well in detecting phishing URLs using the selected top-20 features. Random Forest achieved the most consistent performance because it combined very high accuracy, perfect precision, perfect recall, perfect F1-score, and the lowest balanced error profile. Meanwhile, Naive Bayes demonstrated that a lightweight probabilistic classifier can also provide excellent phishing recall, although with a higher tendency to classify legitimate URLs as phishing.

The experimental results show that the top-20 Information Gain features provide strong discriminative capability for phishing URL classification. The very high performance across multiple algorithms indicates that the selected features contain substantial information for separating phishing and legitimate URLs within the dataset. However, the interpretation of near-perfect performance should be further elaborated in the discussion section by considering feature dominance, internal validation characteristics, and the need for future external validation.

Discussion

The experimental results indicate that the selected top-20 Information Gain features provide strong discriminative capability for modern phishing URL detection. The dominance of URLSimilarityIndex, webpage reference attributes, source-code size, and structural indicators supports previous findings that phishing detection benefits from combining URL-level and webpage-level features rather than relying only on lexical URL patterns [16]. Compared with earlier feature-based studies, which reported strong performance using handcrafted URL and webpage attributes [17], the present result confirms that feature selection can preserve high detection capability while reducing input dimensionality. This is technically relevant because compact feature subsets reduce preprocessing cost, memory usage, and inference complexity in real-time detection systems.

Among the evaluated models, Random Forest produced the most balanced quantitative result, achieving near-perfect accuracy, precision, recall, F1-score, and AUC, with only three phishing URLs misclassified as legitimate. This finding is consistent with benchmark and hybrid-feature studies showing that ensemble-based classifiers are robust for phishing detection because they capture non-linear interactions among heterogeneous features [18]. Naive Bayes also achieved excellent phishing recall, but its higher false-positive count indicates a more conservative classification tendency. Logistic Regression performed strongly but produced the highest false-negative count, suggesting that linear boundaries may be less suitable when feature interactions are complex. Gradient Boosted Trees and Decision Tree remained competitive, confirming that tree-based learning is highly compatible with structured phishing datasets [19].

However, the near-perfect performance should be interpreted cautiously. Recent studies emphasize that phishing detection models may be dataset-dependent and can degrade when evaluated on different distributions or newer attack patterns. Deep learning approaches based on raw URL, HTML, embeddings, recurrent-convolutional architectures, graph neural networks, and attention mechanisms have been proposed to improve representation learning and robustness. Nevertheless, such models often require higher computational resources and are less transparent than selected-feature machine learning models [20]. Therefore, the practical implication of this study is the feasibility of deploying a lightweight and interpretable phishing detection pipeline in browser extensions, email filtering systems, network gateways, and institutional security monitoring platforms. Future work should include external validation, ablation testing of dominant features, and adversarial robustness evaluation to further assess generalization [21].

CONCLUSION

This study presented an engineering-oriented phishing URL detection pipeline by integrating preprocessing, Information Gain-based feature selection, comparative classification, and 10-fold cross-validation within RapidMiner. After preprocessing, 234,903 URL records were used, consisting of 134,834 legitimate URLs and 100,069 phishing URLs. The feature selection stage showed that URLSimilarityIndex, NoOfExternalRef, LineOfCode, NoOfSelfRef, NoOfImage, and NoOfJS were among the most discriminative attributes, indicating that phishing detection is strongly influenced by both URL similarity and webpage structural characteristics. Using the top-20 selected features, all evaluated classifiers achieved accuracy above 99.95%, demonstrating that the selected attributes provide strong class separability within the dataset.

Among the evaluated models, Random Forest produced the most balanced performance, achieving 100.00% accuracy, 100.00% precision, 100.00% recall, 100.00% F1-score, and AUC of 1.000, with only three phishing URLs misclassified as legitimate and no legitimate URLs misclassified as phishing. Naive Bayes achieved the lowest false-negative count, but produced higher false positives, while Logistic Regression generated the highest number of phishing false negatives. These results indicate that the proposed selected-feature detection pipeline can support efficient and interpretable URL security filtering.

This study is limited to internal validation on a single dataset and selected-feature evaluation without external dataset testing. Therefore, the near-perfect performance should be interpreted as strong internal validation rather than universal generalization. Future research should conduct external validation using newer phishing datasets, perform ablation testing on dominant features such as URLSimilarityIndex, and evaluate deployment robustness in real-time browser, email gateway, or network security environments.

REFERENCES

- [1] A. Mughaid, S. AlZu'bi, A. Hnaif, S. Taamneh, A. Alnajjar, and E. A. Elsoud, "An intelligent cyber security phishing detection system using deep learning techniques," *Cluster Comput.*, vol. 25, pp. 3819-3828, 2022, doi: 10.1007/s10586-022-03604-4.
- [2] Y. Liang, "Robust detection of malicious URLs with self-paced wide and deep learning," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 2, pp. 717-730, 2022, doi: 10.1109/TDSC.2021.3121388.
- [3] S. Ariyadasa, S. Fernando, and S. Fernando, "Combining long-term recurrent convolutional and graph convolutional networks to detect phishing sites using URL and HTML," *IEEE Access*, vol. 10, pp. 82355-82375, 2022, doi: 10.1109/ACCESS.2022.3196018.
- [4] T. Wu, M. Wang, Y. Xi, and Z. Zhao, "Malicious URL detection model based on bidirectional gated recurrent unit and attention mechanism," *Applied Sciences*, vol. 12, no. 23, p. 12367, 2022, doi: 10.3390/app122312367.
- [5] W. Wang, F. Zhang, X. Luo, and S. Zhang, "PDRCNN: Precise phishing detection with recurrent convolutional neural networks," *Security and Communication Networks*, vol. 2019, pp. 1-15, 2019, doi: 10.1155/2019/2595794.
- [6] A. S. Bozkir, F. C. Dalgic, and M. Aydos, "GramBeddings: A new neural network for URL based identification of phishing web pages through n-gram embeddings," *Comput. Secur.*, vol. 124, p. 102964, 2023, doi: 10.1016/j.cose.2022.102964.

- [7] C. Opara, Y. Chen, and B. Wei, "Look before you leap: Detecting phishing web pages by exploiting raw URL and HTML characteristics," *Expert Syst. Appl.*, vol. 236, p. 121183, 2024, doi: 10.1016/j.eswa.2023.121183.
- [8] D. Sahoo, C. Liu, and S. C. H. Hoi, "Malicious URL detection using machine learning: A survey," *ACM Comput. Surv.*, vol. 54, no. 4, pp. 1-36, 2021, doi: 10.1145/3446871.
- [9] Z. Alkhalil, C. Hewage, L. Nawaf, and I. Khan, "Phishing attacks: A recent comprehensive study and a new anatomy," *Front. Comput. Sci.*, vol. 3, 2021, doi: 10.3389/fcomp.2021.563060.
- [10] A. Karim, M. Shahroz, K. Mustofa, S. B. Belhaouari, and S. R. K. Joga, "Phishing detection system through hybrid machine learning based on URL," *IEEE Access*, vol. 11, pp. 36805-36822, 2023, doi: 10.1109/ACCESS.2023.3252366.
- [11] A. Aljofey, "An effective detection approach for phishing websites using URL and HTML features," *Sci. Rep.*, vol. 12, p. 8842, 2022, doi: 10.1038/s41598-022-10841-5.
- [12] S. D. Gupta, K. T. Shahriar, H. Alqahtani, D. Alsalman, and I. H. Sarker, "Modeling hybrid feature-based phishing websites detection using machine learning techniques," *Annals of Data Science*, vol. 11, pp. 217-242, 2024, doi: 10.1007/s40745-022-00379-8.
- [13] A. Almomani, "Phishing website detection with semantic features based on machine learning classifiers: A comparative study," *Int. J. Semant. Web Inf. Syst.*, vol. 18, no. 1, pp. 1-24, 2022, doi: 10.4018/IJSWIS.297032.
- [14] R. S. Rao and A. R. Pais, "Detection of phishing websites using an efficient feature-based machine learning framework," *Neural Comput. Appl.*, vol. 31, pp. 3851-3873, 2019, doi: 10.1007/s00521-017-3305-0.
- [15] F. Rashid, B. Doyle, S. C. Han, and S. Seneviratne, "Phishing URL detection generalisation using unsupervised domain adaptation," *Computer Networks*, vol. 245, p. 110398, 2024, doi: 10.1016/j.comnet.2024.110398.
- [16] A. Hannousse and S. Yahiouche, "Towards benchmark datasets for machine learning based website phishing detection: An experimental study," *Eng. Appl. Artif. Intell.*, vol. 104, p. 104347, 2021, doi: 10.1016/j.engappai.2021.104347.
- [17] A. Prasad and S. Chandra, "PhiUSIL: A diverse security profile empowered phishing URL detection framework based on similarity index and incremental learning," *Comput. Secur.*, vol. 136, p. 103545, 2024, doi: 10.1016/j.cose.2023.103545.
- [18] 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, 2023, doi: 10.1016/j.jksuci.2023.01.004.
- [19] X. Zhang, T. Ge, S. Yin, W. Chen, and X. Yang, "Digital transformation and green agricultural development: Evidence from agricultural production systems," *Technol. Forecast. Soc. Change*, vol. 190, p. 122439, 2023.
- [20] Z. Zhai, J. F. Martínez, V. Beltran, and N. L. Martínez, "Decision support systems for agriculture 4.0: Survey and challenges," *Comput. Electron. Agric.*, vol. 170, p. 105256, 2020, doi: 10.1016/j.compag.2020.105256.

- [21] M. L. Yeo and C. M. H. Keske, "From profitability to trust: Factors shaping digital agriculture adoption," *Front. Sustain. Food Syst.*, vol. 8, p. 1456991, 2024, doi: 10.3389/fsufs.2024.1456991.