Journal of Computer Networks, Architecture and High Performance Computing
Volume 08, Number 3, July 2026
https://doi.org/ 10.47709/cnahpc.v8i3.8940

Submitted : 2026-06-21
Accepted : 2026-06-28
Published : 2026-07-01

Comparative Machine Learning Classification for QRIS Quishing Detection Based on URL Features

Permadi Kusuma[*1], Muhammad Yusuf Halim[2], Ruhamah[3]
[1,3]Informatika, Fakultas Teknik Komputer, Universitas Cokroaminoto Palopo, Indonesia
[2]Teknik Informatika, Fakultas Teknik, Universitas Nahdlatul Ulama Kalimantan Timur, Indonesia
[1)] permadikusumauncp@gmail.com, [2)] myusufhalim26@gmail.com, [3)] ruhamah.uma@gmail.com,

ABSTRACT

The increasing adoption of the Quick Response Code Indonesian Standard (QRIS) as a digital payment method has been accompanied by the emergence of quishing, a phishing attack that exploits malicious QR codes to redirect users to fraudulent websites. This study aims to compare the performance of three machine learning classification algorithms Random Forest, Decision Tree, and Naïve Bayes—for detecting phishing URLs in a simulated QRIS quishing environment using URL-based features. The experiments were conducted using a publicly available phishing URL dataset representing simulated URLs that may be encountered after scanning malicious QR codes. Model performance was evaluated using accuracy, precision, recall, and F1-score. The experimental results show that Random Forest achieved the highest accuracy of 96.94%, outperforming Decision Tree 95.32% and Naïve Bayes 65.49%. The superior performance of Random Forest is attributed to its ensemble learning mechanism, which combines multiple decision trees to reduce overfitting, improve robustness, and provide more stable classification performance across diverse URL characteristics. This study contributes a comparative benchmark of machine learning algorithms for URL-based quishing detection and demonstrates that Random Forest is the most effective approach for supporting early phishing detection in QRIS payment systems.

Keywords: decision tree, machine learning, qr code, qris, url phishing

INTRODUCTION

Digital transformation in the financial sector has driven the increasing use of electronic payment systems based on Quick Response Codes, or QR codes. In Indonesia, one form of QR code-based payment standardization is the Quick Response Code Indonesian Standard (QRIS), developed by Bank Indonesia. QRIS is a type of two-dimensional matrix code or barcode capable of storing information both horizontally and vertically (Ajeng Muninggar & Rahardiansah, n.d.). QRIS represents a significant innovation in the digital payment ecosystem as it unifies various payment services under a single QR code standard (Natalia Kristanty, 2024). Although QRIS offers convenience, speed, and transaction efficiency, data security and the potential for fraud remain critical concerns in its implementation (Yudiana, 2023). One of the growing threats in digital transactions is phishing—a form of online fraud in which attackers attempt to obtain a victim's personal information by posing as a trusted entity [4]. Perpetrators even employ social engineering—that is, exploiting cognitive biases, emotions, and interpersonal trust to manipulate victims into handing over access or valuable assets (Marulino Angga, 2025).

Phishing attacks can result in financial losses and the leakage of personal information for both individuals and organizations (Mahmud & Wirawan, 2024). In the context of QR Code-based payments, this threat has evolved into QR phishing, or "quishing"—attacks that use malicious QR Codes to redirect users to fake websites or apps in order to steal sensitive data, such as personal information and login credentials (Njuguna & Ndia, 2025). Code replacement can also be carried out through physical manipulation, where attackers affix a fake QRIS sticker over the merchant's original QRIS (Saputra et al.,

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 08, Number 3, July 2026
https://doi.org/ 10.47709/cnahpc.v8i3.8940

Submitted : 2026-06-21
Accepted : 2026-06-28
Published : 2026-07-01

2026). The global rise in phishing threats indicates that social engineering-based attacks remain one of the dominant forms of cybercrime. According to the APWG's first-quarter 2025 report, there were 1,003,924 phishing attacks recorded—the highest number since the end of 2023. The report also notes that cybercriminals send millions of emails every day containing QR codes to direct users to phishing sites or malware. Additionally, attacks on the online payment and financial sectors, including banking, have increased and account for 30.9% of all phishing attacks (APWG, 2025).

The Indonesia Domain Abuse Data Exchange report, managed by the Indonesian Internet Domain Name Administrator, also shows that the financial sector is one of the primary targets of domain abuse and phishing activities. This situation highlights the need for detection methods capable of identifying suspicious URLs before users complete transactions. One approach that can be used to detect phishing URLs is machine learning. Machine learning is a branch of artificial intelligence that enables systems to learn from data and past experiences to make predictions or decisions without explicit instructions (Mahmud & Wirawan, 2024). In phishing detection, classification algorithms can be used to distinguish between legitimate URLs and phishing URLs based on URL characteristics, such as URL length, use of symbols, domain structure, number of subdomains, and use of the HTTP or HTTPS protocol. One widely used algorithm is Random Forest, a method that combines multiple decision trees to produce a more accurate and stable classification model (Mahendra Alvanof & Kesuma Dinata, 2024).

Several previous studies have examined phishing detection using machine learning algorithms. (Fauzan et al., 2025)compared the Naïve Bayes, Random Forest, and Decision Tree algorithms for detecting phishing websites. The results showed that Random Forest achieved the highest accuracy at 97.2%, followed by Decision Tree at 96.3% and Naïve Bayes at 85.3%. (Alexander et al., 2025) also compared Ensemble Learning, Kernel Methods, and Deep Learning for classifying malicious URLs. The results show that Deep Learning has the highest accuracy, but Random Forest provides the best balance between detection performance and computational efficiency, making it more suitable for real-time implementation. Additionally, [(Kresna Kencana et al., 2022) applied Random Forest to classify phishing and non-phishing websites with an accuracy of 94.36%, and identified key indicators such as the absence of SSL, the number of scripts, and URL length.

On the other hand, research on QRIS security and quishing still focuses primarily on risk analysis and mitigation strategies. (Marulino Angga, 2025) indicates that the implementation of QRIS by businesses still has security vulnerabilities, particularly regarding data protection and QRIS sticker authentication. (Ajhari, 2024) also analyzes quishing threats to QRIS and shows that attacks can occur through the manipulation of QR codes that appear legitimate but redirect transactions to the attacker's account. Recommended solutions include two-factor authentication, transaction location verification, and increased digital security education. Although numerous studies have successfully applied machine learning algorithms to detect phishing websites, most focus on conventional web-based phishing rather than quishing attacks involving malicious QR codes. In contrast, existing QRIS security studies mainly discuss security risks, fraud mitigation, and authentication mechanisms without evaluating the effectiveness of machine learning for detecting phishing URLs generated from QR code scans. Furthermore, comparative studies investigating the performance of different classification algorithms in QRIS-related quishing scenarios remain limited. Therefore, this study addresses this gap by comparing Random Forest, Decision Tree, and Naïve Bayes using URL-based features to establish a benchmark for early quishing detection in QRIS payment systems. Rather than proposing a new algorithm, this study contributes a comparative machine learning framework that demonstrates the feasibility of URL-based classification for enhancing QRIS transaction security.

Although numerous studies have successfully applied machine learning algorithms to detect phishing websites, most focus on conventional web-based phishing rather than quishing attacks involving malicious QR codes. In contrast, existing QRIS security studies mainly discuss security risks, fraud mitigation, and authentication mechanisms without evaluating the effectiveness of machine learning for detecting phishing URLs generated from QR code scans. Furthermore, comparative studies investigating the performance of

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 08, Number 3, July 2026
https://doi.org/ 10.47709/cnahpc.v8i3.8940

Submitted : 2026-06-21
Accepted : 2026-06-28
Published : 2026-07-01

different classification algorithms in QRIS-related quishing scenarios remain limited. The novelty of this study does not lie in proposing a new machine learning algorithm but in establishing a comparative machine learning framework for detecting potential quishing attacks in QRIS payment systems through URL feature analysis. Unlike previous studies that focus either on conventional phishing website detection or on conceptual QRIS security analysis, this research systematically evaluates the performance of Random Forest, Decision Tree, and Naïve Bayes under a simulated QRIS quishing scenario. The study provides a benchmark for identifying the most effective classification algorithm and demonstrates the feasibility of URL-based machine learning as an early detection mechanism for enhancing QRIS transaction security.

LITERATURE REVIEW

Research on phishing attack detection has grown rapidly alongside the increasing use of digital services and internet-based transactions. Various approaches have been used to identify malicious URLs, ranging from rule-based methods to machine learning and deep learning. However, research specifically examining the detection of "Quishing" (QR Code Phishing) attacks on the Quick Response Code Indonesian Standard (QRIS) payment system remains relatively limited. (Kresna Kencana et al., 2022) implemented the Random Forest algorithm to detect phishing websites using a dataset of 2,457 phishing and non-phishing URLs obtained from Kaggle. The study utilized 30 URL and website features, such as URL Length, HTTPS Token, Age of Domain, Web Traffic, and Google Index. Test results showed an accuracy rate of 94.36% and a validation rate of 94.77%, proving that Random Forest is capable of classifying phishing URLs with a high degree of accuracy. Additionally, the model was successfully implemented as a browser extension to detect phishing websites in real time.

However, this study focuses only on the detection of conventional website-based phishing and has not yet considered phishing attacks that exploit QR codes. In "quishing" attacks, users do not immediately see the destination URL before scanning the QR code, so the risk of fraud is higher than in typical website-based phishing. Research conducted by (Ajhari, 2024) discusses the security aspects of the QRIS digital payment system and identifies "quishing" as one of the main threats to the digital payment ecosystem in Indonesia. The study explains that criminals can replace the genuine QRIS with a fake one that directs victims to make transactions to the perpetrator's account. To address this issue, the study proposes additional authentication mechanisms, merchant location verification using the Haversine method, and a rule-based early warning system (rule-based detection).

However, the approach used is still rule-based and relies on location verification, so it has not yet implemented machine learning-based classification techniques to detect malicious URLs contained in QR codes. As a result, the system's ability to recognize new, previously undefined attack patterns remains limited. Another study by (Natalia Kristanty, 2024) examined security trends and challenges in transactions using QRIS in the era of digital transformation. The results indicate that security threats such as phishing, social engineering, and QR code forgery remain major challenges in the implementation of QRIS. The study emphasizes the importance of authentication mechanisms, data encryption, monitoring of suspicious activity, and user education as measures to mitigate digital transaction security risks. Although it provides a comprehensive overview of QRIS security risks, the study employs a qualitative, literature-based approach and therefore has not yet produced an automated detection model capable of identifying "Quishing" attacks in real time based on the URL characteristics contained within QR codes.

Based on previous research studies, it can be concluded that research on phishing URLs has shown good performance through the application of classification algorithms such as Random Forest, while research on QRIS security and quishing is still dominated by approaches such as risk analysis, location verification, and user education. To date, there remains a research gap in the form of the absence of a classification model specifically designed to detect Quishing attacks on the QRIS payment system through the analysis of URL characteristics derived from QR codes. Therefore, this study proposes the application of a classification algorithm to detect Quishing attacks on the QRIS payment system based on URL analysis.

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 08, Number 3, July 2026
https://doi.org/ 10.47709/cnahpc.v8i3.8940

Submitted : 2026-06-21
Accepted : 2026-06-28
Published : 2026-07-01

This approach is expected to automatically identify malicious URLs based on URL patterns and characteristics, thereby providing an additional layer of security for QRIS transactions and helping users avoid potential Quishing attacks before transactions are carried out.

METHOD

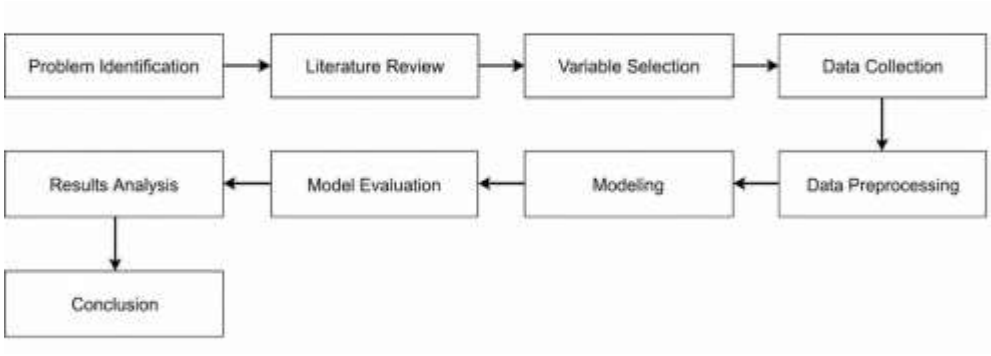


Figure 1. Steps in Building a Machine Learning Model.

The first stage is problem formulation. In this stage, we identify issues related to the rise in QR code-based phishing attacks, or "quishing," on the QRIS payment system. The focus is on how to detect malicious URLs generated from QR code scans using a machine learning approach. The second stage is a literature review, which involves examining previous research related to phishing detection, the application of the Random Forest, Decision Tree, and Naïve Bayes algorithms, and the security of the QRIS payment system. The literature review is used to strengthen the theoretical foundation and identify research gaps that can be explored.

The third stage is variable selection. The independent variables in this study are the classification algorithm and URL features, while the dependent variables are the classification results and model performance metrics. The classification results consist of two categories: phishing and legitimate. Model performance is measured using accuracy, precision, recall, F1-score, macro average, and weighted average. The fourth stage is data collection. The dataset used was obtained from the Kaggle website via the Phishing Websites Dataset. This dataset contains 88,647 phishing and legitimate websites, represented as numerical features extracted from URL characteristics, domains, and website behavior. Each feature represents a specific indicator, such as the number of symbols in the URL, the presence of SSL, redirect activity, and domain structure characteristics.

The fifth stage is data preprocessing. During this stage, the dataset is examined, class labels are adjusted, features are selected, and the data is split into training and test sets. The features used focus on URL characteristics, such as the use of IP addresses in URLs, the presence of the "@" symbol, excessively long URLs, suspicious subdomain or subpage structures, redirection patterns using "//," the insertion of HTTP/HTTPS protocols in domain names, and the use of URL shortening services. The preprocessing process aims to ensure the data is ready for use in the machine learning modeling stage.

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 08, Number 3, July 2026
https://doi.org/ 10.47709/cnahpc.v8i3.8940

Submitted : 2026-06-21
Accepted : 2026-06-28
Published : 2026-07-01

The sixth stage is modeling. In this stage, classification models are built using the Random Forest, Decision Tree, and Naïve Bayes algorithms. Random Forest is used to classify websites as phishing or non-phishing based on the available features. To determine the classification algorithms' ability to detect phishing URLs, this study also conducted a comparative test using Decision Tree and Naïve Bayes. The design of the phishing link detection classification model is shown in Figure 2. The best algorithm was selected based on the results of the model performance evaluation. The algorithm with the best performance was used as the primary model in this study.

Figure 2. Phishing Link Detection Classification Model.

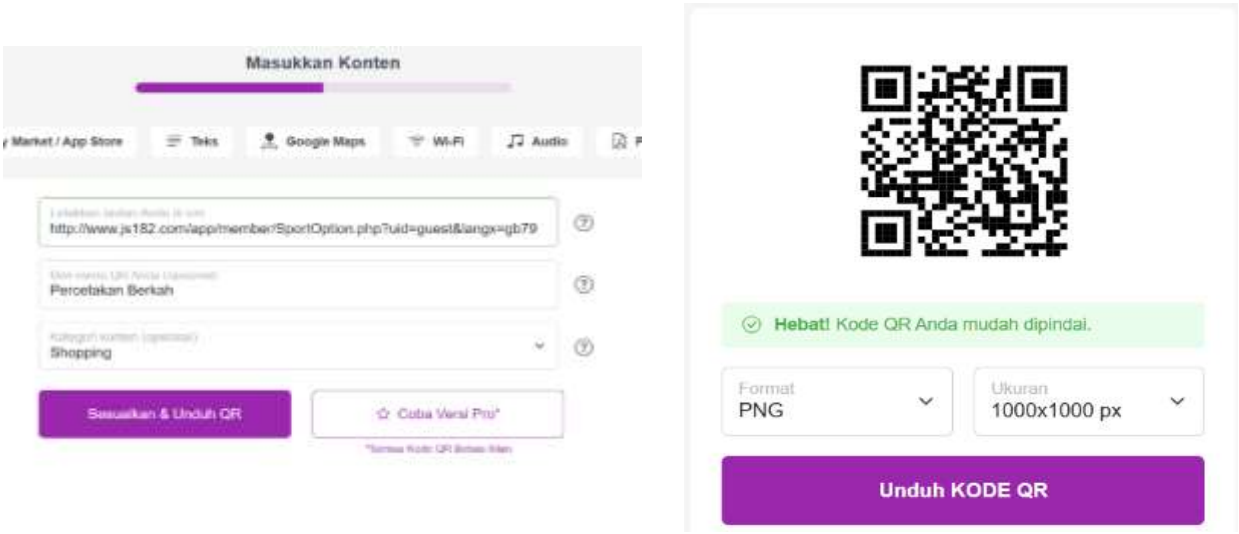


Figure 3. The Process of Embedding a Phishing Link in a QR Code.

An example of a URL used in the simulation is http://www.js182.com/app/member/SportOption.php?uid=guest&langx=gb79. This URL is assumed to be a malicious URL obtained from a QR code scan. Once the URL is obtained, the system performs an analysis based on the URL's characteristics and classifies it using a pre-built machine learning model. Thus, this simulation is used to illustrate how the model can serve as an early-warning system against potential phishing attacks on QRIS payments.

The seventh stage is model evaluation. The classification model that has been built is evaluated using test data to determine the algorithm's performance in detecting phishing URLs. The evaluation is based on

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 08, Number 3, July 2026
https://doi.org/ 10.47709/cnahpc.v8i3.8940

Submitted : 2026-06-21
Accepted : 2026-06-28
Published : 2026-07-01

a confusion matrix consisting of True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). True Positive refers to the number of phishing data points successfully predicted as phishing, while True Negative refers to the number of legitimate data points successfully predicted as legitimate. False Positive refers to the number of legitimate data points incorrectly predicted as phishing, while False Negative refers to the number of phishing data points incorrectly predicted as legitimate.

The evaluation metrics used in this study include accuracy, precision, recall, and F1-score. Accuracy is used to measure the model's overall accuracy in classifying phishing and legitimate URLs. Precision is used to measure the model's accuracy in identifying phishing URLs. Recall is used to measure the model's ability to detect all actual phishing data. The F1-score is used to measure the balance between precision and recall.

In addition to these four metrics, this study also uses the macro average and weighted average. The macro average is used to calculate the average precision, recall, and F1-score for each class without considering the amount of data in each class. This metric is used to assess the model's performance across all classes uniformly, particularly when evaluating the model's ability to detect minority classes. The weighted average is used to calculate the average precision, recall, and F1-score while taking into account the proportion of data in each class. This metric is particularly relevant for imbalanced datasets because it provides a more realistic picture of the model's performance based on the actual data distribution (Hinojosa Lee et al., 2024).

The eighth stage is the analysis of results. The evaluation results for each algorithm were analyzed to determine the effectiveness of Random Forest, Decision Tree, and Naïve Bayes in detecting phishing URLs. The analysis was conducted by comparing the accuracy, precision, recall, F1-score, macro average, and weighted average values. Based on the results of this comparison, the algorithm with the best performance was selected as the primary classification model. The final stage is drawing conclusions, which involves summarizing the research results based on the experiments conducted and addressing issues related to URL-based phishing detection in the context of quishing attacks within the QRIS payment system.

RESULT

This study uses the Phishing Websites Dataset obtained from the Kaggle website. The dataset contains data on phishing and legitimate websites, represented as numerical features extracted from URLs, domains, and website behavior. Although the dataset focuses on website URLs in general, this study adapts it to the QRIS context, assuming that the URLs originate from QR code scans. This approach is relevant because phishing attacks on QRIS typically exploit fake QR codes to redirect users to malicious URLs. The detection process in this study begins with the URL obtained from scanning a QR code. These URLs are then analyzed based on numerical features that represent URL characteristics, such as the number of characters, the presence of SSL, domain structure, and redirect activity. These features are subsequently used as input for a classification model to determine whether a URL is phishing or non-phishing. Thus, the developed model can serve as an initial approach for detecting potential phishing attacks in QRIS-based digital payment transactions. The dataset used consists of 88,647 URLs, comprising 58,000 non-phishing URLs and 30,647 phishing URLs. The composition of the dataset is shown in Table 1.

Table 1. Dataset by Class

No	Klasifikasi	Jumlah *Record Dataset*
1	*Non-phishing*	58.000
2	*Phishing*	30.647
	Jumlah	88.647

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 08, Number 3, July 2026
https://doi.org/ 10.47709/cnahpc.v8i3.8940

Submitted : 2026-06-21
Accepted : 2026-06-28
Published : 2026-07-01

The volume of non-phishing data is greater than that of phishing data. This indicates an imbalance in the amount of data across classes, so model evaluation is conducted not only using accuracy but also using precision, recall, F1-score, macro average, and weighted average. The use of these metrics is necessary to analyze model performance more comprehensively, particularly in assessing the model's ability to detect the phishing class. It should be noted that the dataset used in this study is not derived from actual QRIS transactions or real QRIS payment systems. Instead, a publicly available phishing URL dataset is employed to simulate URLs that may be encountered after scanning malicious QR codes in quishing scenarios. Therefore, the experimental results represent a simulation of QRIS-related phishing detection rather than an evaluation using real-world QRIS transaction data. This approach enables a controlled comparison of machine learning algorithms while providing preliminary evidence of the feasibility of URL-based quishing detection.

The phishing URL dataset contains URL-based features representing lexical and host-related characteristics commonly used in phishing detection. These features include indicators such as URL length, the use of an IP address instead of a domain name, URL shortening services, the presence of special characters, HTTPS status, DNS records, domain registration information, and redirection behavior. Each feature was extracted from the original dataset and used as an input variable for the classification models. In this study, all available URL features were retained during the preprocessing stage, and no feature elimination technique was applied. This decision was made to preserve the complete representation of phishing-related characteristics contained in the dataset. Consequently, the Random Forest, Decision Tree, and Naïve Bayes classifiers were trained using the same set of URL-based features to ensure a fair comparison of model performance.

DISCUSSIONS

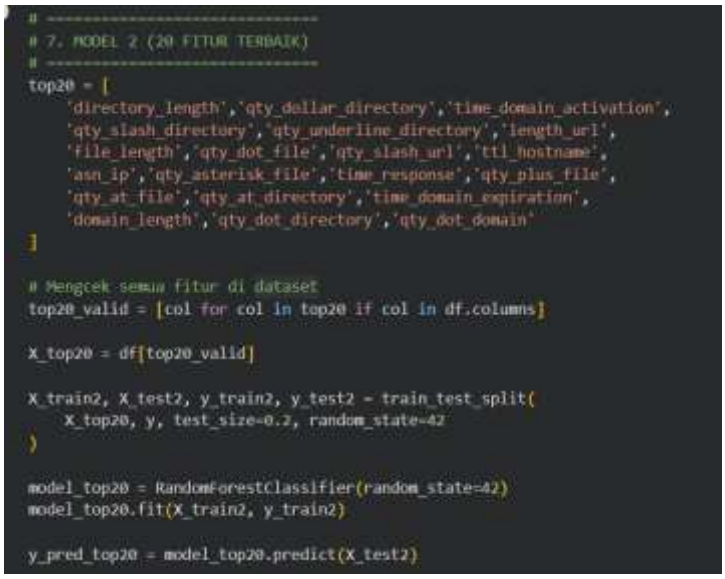
DATA PREPROCESSING DAN FEATURE SELECTION

The preprocessing stage is conducted to ensure the dataset is ready for use in the classification model training process. The dataset is first analyzed to identify the data structure, feature types, and available classification labels. The dataset used contains various numerical features extracted from URL characteristics, domains, directories, files, parameters, and network security information such as SSL, DNS, and domain validity periods. At this stage, data validation, class label adjustment, and the selection of relevant features are performed. The features used in the classification process include URL characteristics, such as URL length, the number of hyphens (-), the number of periods (.), the use of the HTTPS protocol, domain type, subdomain length, the presence of SSL, and redirect activity. Since the features are in numerical form, the data can be processed by machine learning algorithms after format adjustments and data partitioning have been performed. This study also performed a feature selection process to identify the features that contribute most to the classification results. Feature selection was performed using the Python library Scikit-learn. The method used was feature importance from the Random Forest algorithm, as this method is capable of measuring the degree of contribution of each feature to the classification results. The feature selection process is shown in Figure 6.

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 08, Number 3, July 2026
https://doi.org/ 10.47709/cnahpc.v8i3.8940

Submitted : 2026-06-21
Accepted : 2026-06-28
Published : 2026-07-01



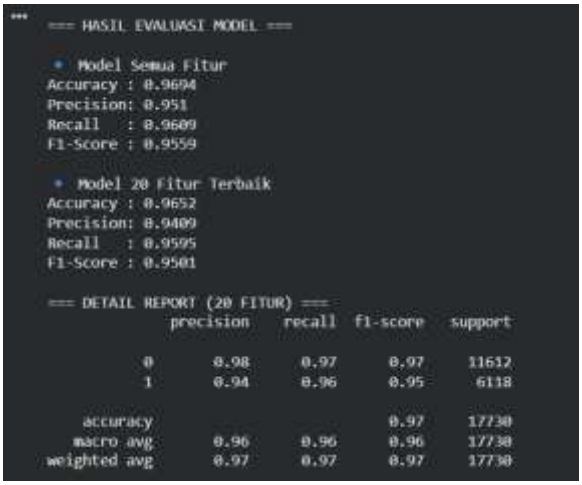
```
# ============================
# 7. MODEL 2 (20 FITUR TERBAIK)
# ============================
top20 = [
    'directory_length','qty_dollar_directory','time_domain_activation',
    'qty_slash_directory','qty_underline_directory','length_url',
    'file_length','qty_dot_file','qty_slash_url','ttl_hostname',
    'asn_ip','qty_asterisk_file','time_response','qty_plus_file',
    'qty_at_file','qty_at_directory','time_domain_expiration',
    'domain_length','qty_dot_directory','qty_dot_domain'
]

# Mengcek semua fitur di dataset
top20_valid = [col for col in top20 if col in df.columns]

X_top20 = df[top20_valid]

X_train2, X_test2, y_train2, y_test2 = train_test_split(
    X_top20, y, test_size=0.2, random_state=42
)

model_top20 = RandomForestClassifier(random_state=42)
model_top20.fit(X_train2, y_train2)

y_pred_top20 = model_top20.predict(X_test2)
```

Figure 4. The Phishing URL Feature Selection Process

Based on the results of feature selection, a comparison was made between a model using all features and a model using the top 20 features. The evaluation results showed that the Random Forest model using all features performed slightly better than the model using the top 20 features. Therefore, the model using all features was selected as the primary model because it provided more optimal phishing detection performance.

Results of the Random Forest Model Evaluation

The first test was conducted using the Random Forest algorithm. Based on the evaluation results, the Random Forest model that used all features achieved an accuracy of 96.94%, a precision of 95.10%, a recall of 96.09%, and an F1-score of 95.59%. These results indicate that Random Forest is capable of classifying phishing and non-phishing URLs with a low error rate. The Random Forest evaluation results are shown in Figure 7, and the Random Forest classification report results are shown in Table 2.



```
=== HASIL EVALUASI MODEL ===

 • Model Semua Fitur
Accuracy : 0.9694
Precision: 0.951
Recall   : 0.9609
F1-Score : 0.9559

 • Model 20 Fitur Terbaik
Accuracy : 0.9652
Precision: 0.9489
Recall   : 0.9595
F1-Score : 0.9501

=== DETAIL REPORT (20 FITUR) ===
              precision    recall  f1-score   support

           0       0.98      0.97      0.97     11612
           1       0.94      0.96      0.95      6118

    accuracy                           0.97     17730
   macro avg       0.96      0.96      0.96     17730
weighted avg       0.97      0.97      0.97     17730
```

Figure 7. Results of the Random Forest Selection Evaluation

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 08, Number 3, July 2026
https://doi.org/ 10.47709/cnahpc.v8i3.8940

Submitted : 2026-06-21
Accepted : 2026-06-28
Published : 2026-07-01

Table 2, the precision value

Label	***Precision***	***Recall***	***F1-score***	***Support***
Non-phishing (0)	0.98	0.97	0.97	11612
Phishing (1)	0.94	0.96	0.95	6118
Macro Average	0.96	0.96	0.96	17730
Weighted Average	0.97	0.97	0.97	17730

Based on Table 2, the precision value of 0.98 for the non-phishing class indicates that the model is highly effective at predicting safe URLs. Meanwhile, the phishing class achieved a precision of 0.94, meaning that most URLs predicted as phishing actually fall into the phishing category. The recall value for the phishing class is 0.96, indicating that the model is capable of detecting the majority of phishing URLs effectively. This is crucial in the context of digital security because errors in identifying phishing URLs can increase the risk of users being redirected to malicious sites. The F1-score for the non-phishing class is 0.97 and for the phishing class is 0.95, showing that the Random Forest model maintains a balance between precision and recall across both classes. Additionally, the macro average of 0.96 indicates that the model's performance is relatively balanced across all classes, while the weighted average of 0.97 shows that the model's overall performance remains high when accounting for the distribution of data counts across each class. Thus, Random Forest demonstrates excellent, stable, and effective performance in detecting phishing URLs.

Results of the Naïve Bayes Model Evaluation

The second test was conducted using the Naïve Bayes algorithm. Based on the evaluation results, the Naïve Bayes algorithm achieved an accuracy of 65.49%. However, the precision, recall, and F1-score values for the phishing class were all 0.00. This indicates that the model was unable to accurately identify the phishing class. The Naïve Bayes evaluation results are shown in Figure 8, and the Naïve Bayes classification report results are shown in Table 3.

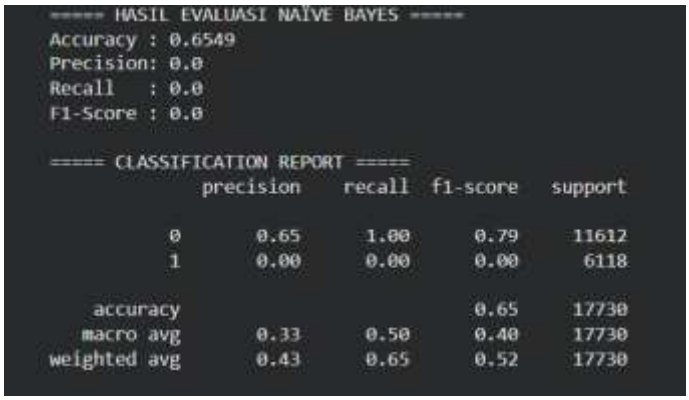


Figure 8. Results of the Naïve Bayes Selection Evaluation

Table 3. Results of the Naïve Bayes Algorithm Testing

Label	***Precision***	***Recall***	***F1-score***	***Support***
Non-phishing (0)	0.65	1.00	0.79	11612
Phishing (1)	0.00	0.00	0.00	6118
Macro Average	0.33	0.50	0.40	17730
Weighted Average	0.43	0.65	0.52	17730

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 08, Number 3, July 2026
https://doi.org/ 10.47709/cnahpc.v8i3.8940

Submitted : 2026-06-21
Accepted : 2026-06-28
Published : 2026-07-01

Based on Table 3, the Naïve Bayes model was able to identify all non-phishing data with a recall of 1.00. However, the model failed to detect the phishing class because the precision, recall, and F1-score values for the phishing class were 0.00. This indicates that the model tends to classify data into the non-phishing class and exhibits a bias toward the majority class. The macro average precision of 0.33, recall of 0.50, and F1-score of 0.40 indicate that the model's average performance across both classes remains low. Meanwhile, the weighted average precision of 0.43, recall of 0.65, and F1-score of 0.52 indicate that the model's overall performance is still suboptimal. Thus, the Naïve Bayes classifier is not yet effective for detecting phishing URLs in the dataset used in this study.

D. Results of the Decision Tree Model Evaluation

The third test was conducted using the Decision Tree algorithm. Based on the evaluation results, the Decision Tree algorithm achieved an accuracy of 95.32%, a precision of 92.97%, a recall of 93.53%, and an F1-score of 93.25%. These results indicate that the Decision Tree algorithm is capable of classifying phishing and non-phishing URLs with fairly good performance. The evaluation results for the Decision Tree are shown in Figure 9, and the Decision Tree classification report is shown in Table 4.

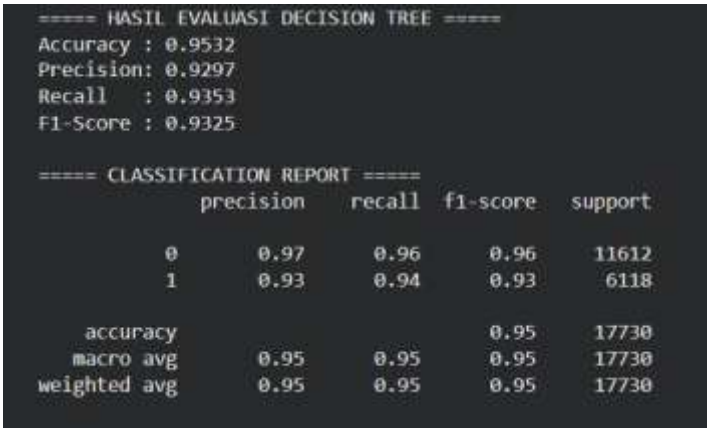


Figure 9. Selection Decision Tree Evaluation Results

Table 4 Results of the Decision Tree Algorithm Testing

Label	***Precision***	***Recall***	***F1-score***	***Support***
Non-phishing (0)	0.97	0.96	0.96	11612
Phishing (1)	0.93	0.94	0.93	6118
Macro Average	0.95	0.95	0.95	17730
Weighted Average	0.95	0.95	0.95	17730

Based on Table 4, the non-phishing class achieved a precision of 0.97, a recall of 0.96, and an F1-score of 0.96. This indicates that the Decision Tree is capable of accurately identifying safe URLs. For the phishing class, a precision of 0.93 and a recall of 0.94 indicate that most phishing URLs were correctly identified, although there were still a small number of misclassifications. The macro average and weighted average values, each at 0.95 for precision, recall, and F1-score, show that the Decision Tree performs consistently across both classes. Overall, the Decision Tree can be used as a model for classifying phishing URLs because it delivers high performance, even though its scores are still slightly below those of the Random Forest. After testing the three algorithms, the next step was to compare the performance of Random Forest, Naïve Bayes, and Decision Tree. The comparison was based on accuracy, precision, recall, F1-score, macro average, and weighted average. The results of the accuracy comparison are shown in Table 5.

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 08, Number 3, July 2026
https://doi.org/ 10.47709/cnahpc.v8i3.8940

Submitted : 2026-06-21
Accepted : 2026-06-28
Published : 2026-07-01

Table 5. Comparison of Algorithm Accuracy

Algoritma	***Accuracy***
Random Forest	96,94%
Naïve Bayes	65,49%
Decision Tree	95,32%

Based on Table 5, Random Forest achieved the highest accuracy of 96.94%. Decision Tree ranked second with an accuracy of 95.32%, while Naïve Bayes achieved the lowest accuracy of 65.49%. These results indicate that Random Forest has the best classification performance compared to the other two algorithms. Furthermore, a comparison of the precision, recall, and F1-score metrics is shown in Table 6.

Table 6. Comparison of Algorithm Evaluation Metrics

Metrik	***Random Forest***	***Naïve Bayes***	***Decision Tree***
Precision (Macro Average)	96%	33%	95%
Recall (Macro Average)	96%	50%	95%
F1-score (Macro Average)	96%	40%	95%
Precision (Weighted Average)	97%	43%	95%
Recall (Weighted Average)	97%	65%	95%
F1-score (Weighted Average)	97%	52%	95%

Based on Table 6, Random Forest demonstrated the best performance across all evaluation metrics. The macro average of 96% indicates that the model is capable of delivering balanced performance for both the phishing and non-phishing classes. The weighted average of 97% shows that the model's performance remains high when accounting for the amount of data in each class. This indicates that Random Forest not only has high accuracy but is also capable of maintaining a balanced classification across both classes.

Decision Tree also performed very well, with an average of 95% across all metrics. These results show that the Decision Tree is capable of consistently classifying phishing and non-phishing URLs. However, its performance is still slightly below that of the Random Forest. Meanwhile, Naïve Bayes showed the lowest performance. The low macro average and weighted average values indicate that this algorithm is not yet capable of optimally distinguishing between phishing and non-phishing URLs, primarily because it fails to recognize the phishing class. Overall, Random Forest emerged as the best algorithm in this study. The advantage of Random Forest stems from its ability to build multiple decision trees and combine the prediction results from each tree to produce a more stable final decision. This characteristic makes Random Forest better equipped to handle feature variations in the phishing URL dataset compared to a single Decision Tree or Naïve Bayes. Thus, Random Forest can be used as the primary model for detecting potential quishing attacks on QRIS payment systems based on URL analysis.

CONCLUSION

This study successfully applied a machine learning classification algorithm to detect phishing attacks on the QRIS payment system based on URL analysis by comparing three algorithms, namely Random Forest, Decision Tree, and Naïve Bayes. Based on the test results, the Random Forest algorithm obtained the best performance with precision, recall, and F1-score values in the range of 96%–97%, thus being able to detect phishing URLs with a high and stable level of accuracy. The Decision Tree algorithm also showed good performance with an average value of 95% in all test metrics, although still slightly below Random Forest. Meanwhile, the Naïve Bayes algorithm showed the lowest performance because it was not able to detect phishing classes optimally. The results of this study indicate that the URL analysis approach using the Random Forest algorithm is effective as an early detection model for potential quishing attacks on the QRIS payment system, thereby supporting increased security for QR Code-based digital transactions. For

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 08, Number 3, July 2026
https://doi.org/ 10.47709/cnahpc.v8i3.8940

Submitted : 2026-06-21
Accepted : 2026-06-28
Published : 2026-07-01

further research, it is recommended to use a more diverse and more specific URL quishing dataset for the QRIS context so that machine learning models can recognize broader attack patterns and approach real-world conditions. In addition, the development of a real-time detection system integrated with digital payment applications, browsers, or QR Code scanning systems can also be carried out to increase the effectiveness of quishing attack detection.

REFERENCES

Ajeng Muninggar, R., & Rahardiansah, T. (n.d.). *Pemberdayaan Hukum Pembayaran Digital melalui Penggunaan Teknologi Quick Response Code Indonesian Standar di Masyarakat* (Vol. 6).

Ajhari, A. A. (2024). *Analisis Keamanan Sistem Pembayaran Digital Quick Response Code Indonesian Standard (QRIS)*.

Alexander, A. D., Warta, J., Lubis, H., Mahbub, A. R., & Rasim, R. (2025). Analisis Komparatif Algoritma Machine Learning Untuk Mendeteksi Malicious Url Berbasis FITUR GANDA. *Jurnal Manajamen Informatika Jayakarta*, *5*(3), 302. https://doi.org/10.52362/jmijayakarta.v5i3.2101

APWG. (2025). Phishing Activity Trends Report APWG. *Anti-Phishing Working Group*. https://docs.apwg.org/reports/apwg_trends_report_q1_2025.pdf

Fauzan, R., Vitianingsih, A. V., Cahyono, D., Maukar, A. L., & Suprio, Y. A. B. (2025). Penerapan Algoritma Klasifikasi pada Machine Learning untuk Deteksi Phishing. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, *5*(2), 531–540. https://doi.org/10.57152/malcom.v5i2.1968

Kresna Kencana, A., Dwi Ananda, F., & Dwi Hartanto, A. (2022). Implementasi Metode Random Forest Klasifikasi untuk Phishing Link Detection. *Information Technology Journal*, *4*(2).

Mahendra Alvanof, M., & Kesuma Dinata, R. (2024). Penerapan Algoritma Random Forest dalam Deteksi dan Klasifikasi Ransomware. In *Jurnal Elektronika dan Teknologi Informasi* (Vol. 5, Number 2).

Mahmud, A. F., & Wirawan, S. (2024). *Deteksi Phishing Website menggunakan Machine Learning Metode Klasifikasi* (Vol. 13, Number 4). http://sistemasi.ftik.unisi.ac.id

Marulino Angga. (2025). Pemalsuan Qris Pelaku Usaha Dalam Transaksi Sistem Pembayaran Digital Melalui Pola Kejahatan Phishing. 1–66.

Natalia Kristanty, D. (2024). Tren dan Tantangan Keamanan Bertransaksi dengan Qris dalam Era Transformasi Sistem Pembayaran Digital. In *Syntax Admiration* (Vol. 5, Number 10).

Njuguna, D., & Ndia, J. (2025). Quick Response Code Security Attacks and Countermeasures: A Systematic Literature Review. *Journal of Cyber Security*, *7*(1), 1–20. https://doi.org/10.32604/jcs.2025.059398

Saputra, I. P. G. D., Kesiman, M. W. A., & Sunarya, I. M. G. (2026). Deteksi Pemalsuan QRIS MPM Statis Menggunakan YOLO, PaddleOCR dan Metode Berbasis Aturan. *Bulletin of Computer Science Research*, *6*(2), 763–775. https://doi.org/10.47065/bulletincsr.v6i2.1020

Yudiana, A. A. (2023). Pengaruh Penggunaan Qris, Pendapatan, Tabungan Dan Gaya Hidup Terhadap Perilaku Konsumtif. *Contemporary Studies in Economic, Finance and Banking*, *2*(4), 739–746. https://doi.org/10.21776/csefb.2023.02.4.15

* Corresponding author