Journal of Applied Informatics and Computing (JAIC)
Vol.10, No.1, February 2026, pp. 204~211
e-ISSN: 2548-6861

Implementation of the Random Forest Algorithm for Anomaly Detection of Phishing Attacks on Computer Networks

Andika Agus Slameto [1], Ben Rafi Kahmas [2]
Informatics, Universitas Amikom Yogyakarta, Indonesia
rmkt.andika@amikom.ac.id [1], benrafikahmas@students.amikom.ac.id [2]

Article Info

Article history:

Received 2025-09-25
Revised 2025-12-12
Accepted 2025-12-22

Keyword:

Anomaly Detection,
Computer Networks,
Machine Learning,
Phishing,
Random Forest.

ABSTRACT

Phishing attacks are among the most common and dangerous cyber security threats, as they exploit manipulation techniques to steal sensitive user information. This research focuses on leveraging the Random Forest algorithm to identify anomalies caused by phishing attacks in computer network environments. Random Forest was selected for its superior classification performance and its capability to handle a wide variety of data types with minimal over fitting. The experimental dataset consists of captured network traffic, containing both benign activities and malicious events labeled as phishing. The data underwent pre-processing, feature selection, and model training using Random Forest. The experimental results show that the model achieved 98% accuracy, with precision 98%, recall 98%, and F1-score 98%. This study also reveals that URL features such as the percentage of external links redirecting back to the original domain, frequent domain name mismatches, the number of hyphens (-) in the URL, and the presence of data submission via email are relevant and effective in distinguishing phishing from non-phishing URLs. These findings confirm that Random Forest can serve as an effective method for identifying phishing attacks based on URL characteristics.

I. INTRODUCTION

The rapid advancement of digital technology has significantly increased society's reliance on internet-based services such as digital banking, e-commerce, and online communication. However, this development has also been accompanied by a rise in cyber threats, one of the most prevalent being phishing. As reported by the Anti-Phishing Working Group (APWG), the fourth quarter of 2024 saw 989,123 phishing attacks worldwide, leading to an estimated total loss of USD 128,980 [1]. In Indonesia, the National Cyber and Encryption Agency (BSSN) reported more than 26 million phishing attacks in 2024, primarily targeting government and financial sectors [2].

Traditional phishing detection methods such as blacklisting and signature-based detection are increasingly considered inadequate to counter sophisticated and dynamic attacks. Signature and database updates often lag in identifying emerging threats, including zero-day phishing attacks. By examining structural features such as the URL's overall length, frequency of symbols, and information at the domain level, machine learning models can significantly improve their ability to distinguish phishing URLs from legitimate ones [3].

This research is grounded in the anomaly-based intrusion detection framework, which identifies threats by recognizing deviations from established patterns of normal behavior. Unlike signature-based methods that depend on pre-existing threat databases, anomaly-based systems build behavioral profiles of legitimate traffic and flag activities that fall outside acceptable thresholds. Random Forest's ability to handle complex, high-dimensional feature spaces makes it well-suited for this detection paradigm, as it can identify subtle structural anomalies in URLs that distinguish phishing attempts from legitimate requests. Within network security architecture, this approach operates at the application layer, analyzing HTTP/HTTPS traffic patterns without requiring full content inspection, thus preserving user privacy while maintaining detection effectiveness.

Multiple machine learning algorithms have been employed to identify phishing websites using URL features, including Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Decision Tree (DT), Logistic Regression (LR), XGBoost, Convolutional Neural Networks (CNN), and advanced deep learning frameworks. Comparative studies show that CNN models achieve the highest accuracy, reaching 99% with precision, recall, and F1-score values of 98%–99%, while Random Forest achieves slightly lower but still competitive performance with 98% accuracy and balanced precision and recall [4]. Despite CNN's superior results, Random Forest remains a widely used approach due to its ability to handle high-dimensional data, interpretability, and robustness against overfitting [5].

Most existing phishing detection approaches still rely on analyzing website content or email metadata, which poses privacy and latency challenges. URL structure–based detection offers a more efficient and privacy-preserving alternative since it does not require loading the entire website [6]. In network environments, this approach allows early detection before users connect to phishing websites, thus reducing potential damage [7].

Several studies have explored enhancements in phishing detection. Rundong Yang combined deep CNN with Random Forest using only URL features, achieving 99.35% accuracy [8]. Piñeiro et al. compared multiple models and found that Random Forest yielded the best results with 94% accuracy and precision [9]. Ojewumi's study confirmed Random Forest as the top performer with an accuracy of 98.35% [10]. Abdul Karim et al. developed a hybrid model combining LR, SVC, and DT with feature selection and hyperparameter tuning, achieving 98.12% accuracy [11]. Tarun Choudhary et al. demonstrated Random Forest's superior performance on PhishTank and UCI datasets with accuracies of 98.80% and 97.87%, respectively [12]. Alsharaiah further improved detection by combining Random Forest classification with K-means clustering, reaching 98.64% accuracy [13].

From an implementation perspective, deploying machine learning models in production security infrastructure demands careful consideration of performance constraints and integration requirements. Real-time phishing detection must operate with minimal latency—ideally within milliseconds—to avoid degrading user experience when integrated into web proxies, next-generation firewalls, or secure web gateways. Random Forest offers several practical advantages for such deployments: the model supports parallel inference across multiple decision trees, maintains predictable response times regardless of URL complexity, and remains interpretable enough for security teams to understand classification decisions. This interpretability is particularly valuable in enterprise environments where security analysts need to justify blocking decisions and maintain audit trails for compliance purposes.

This research aims to develop an adaptive and low-latency phishing detection system by implementing a Random Forest–based AI model that analyzes URL structures. The proposed model is designed to operate efficiently in network security systems such as firewalls and proxies, thus enhancing cyber resilience against phishing attacks in an era of rapid digital transformation.

II. METHOD

A. Data Preparation

This study utilized a secondary dataset titled "Phishing Dataset for Machine Learning" obtained from Kaggle [14]. The dataset consists of 10,000 entries, including 5,000 phishing URLs and 5,000 non-phishing URLs. Each entry contains 50 numerical features representing URL structural characteristics, such as URL length, number of subdomains, HTTPS usage, special characters, and redirect elements. This dataset was selected because it has a balanced class distribution and clearly defined classification labels, enabling fair model training and evaluation. Furthermore, it has been referenced in a previous study published in MIJARCSE in 2024 [15], which strengthens its validity and credibility as a research reference.

Figure 1. Flowchart

B. Exploratory Data Analysis

An initial EDA was conducted to gain a general overview of the dataset. Using dataset.info(), it was confirmed that the dataset contains 10,000 rows and 50 columns, with no missing values. Most features are of type int64 (43 columns), while the remaining are float64 (3 columns). Descriptive statistics were generated using dataset.describe() to examine feature

distributions, minimum and maximum values, means, and standard deviations. A correlation matrix heatmap was analyzed to identify relationships between features and the target variable. The results revealed that FrequentDomainNameMismatch and PctExtNullSelfRedirectHyperlinksRT had the highest correlations with the phishing class (0.45), suggesting they are highly relevant indicators for phishing detection.

	Feature	Fisher Score
0	id	29994.001200
48	PctExtNullSelfRedirectHyperlinksRT	4127.232733
35	FrequentDomainNameMismatch	2742.448014
5	NumDash	1608.132123
39	SubmitInfoToEmail	1466.317863
34	PctNullSelfRedirectHyperlinks	1331.384975
30	InsecureForms	1112.073987
1	NumDots	946.736097
27	PctExtHyperlinks	723.239761
25	NumSensitiveWords	696.548355
40	IframeOrFrame	585.639947
3	PathLevel	555.621595

Figure 3. Fisher Score

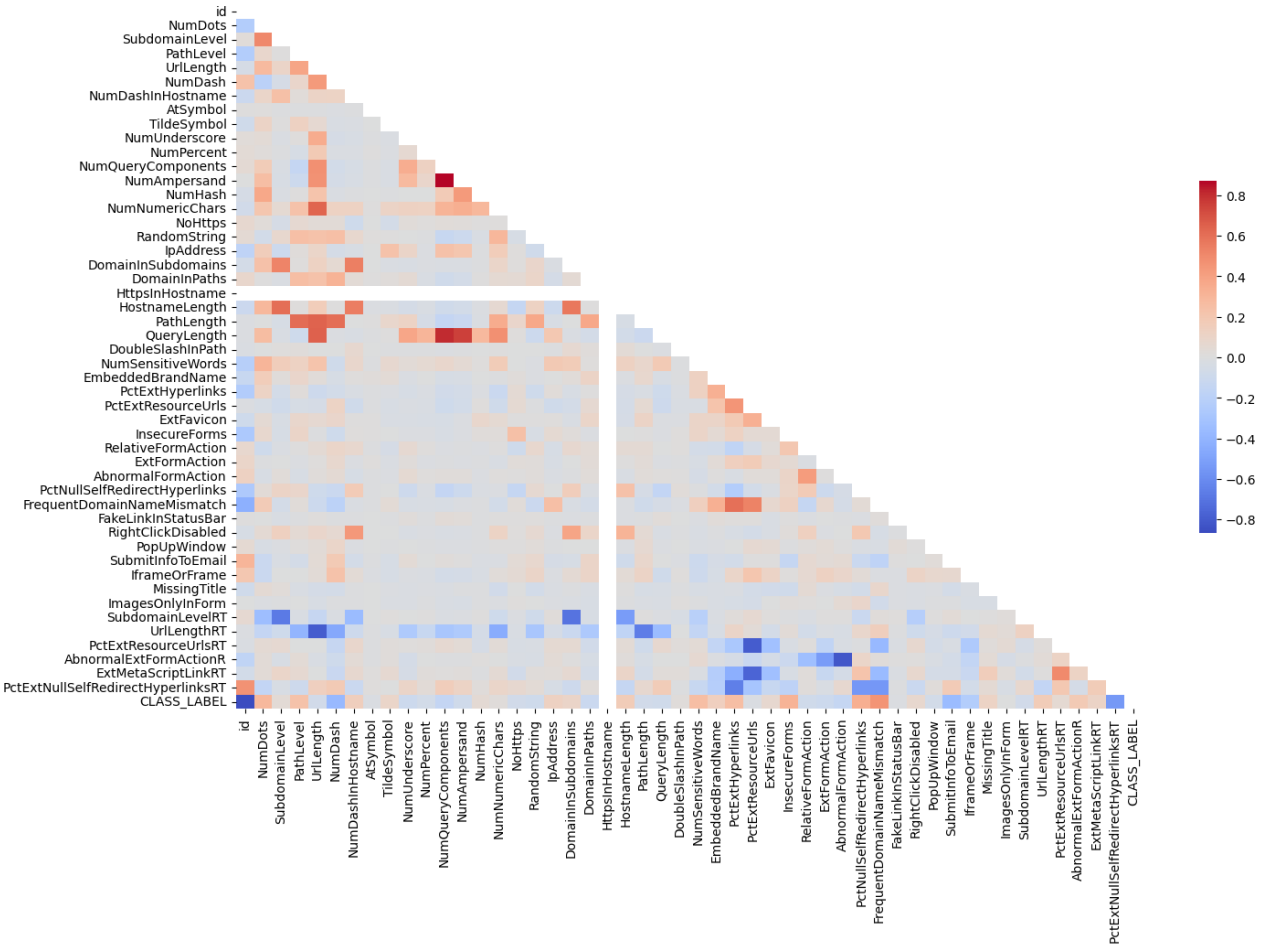

Figure 2. Heatmap Correlation

C. Feature Selection

Since most features exhibited relatively weak correlations, Fisher Score was employed for feature selection to identify the most significant attributes for classification.

Fisher Score evaluates how well each feature differentiates between phishing and non-phishing classes. Features with a Fisher Score higher than 500 were retained for model training, ensuring that only highly relevant features contributed to the prediction process, thereby improving model interpretability and efficiency.

D. Data Balancing

Although the dataset was balanced in terms of phishing and non-phishing labels, Synthetic Minority Oversampling Technique (SMOTE) was applied to address potential variance within feature distributions and ensure that the model could effectively capture minority-class patterns. SMOTE generates synthetic data points by interpolating existing samples, which enhances the model's ability to generalize.

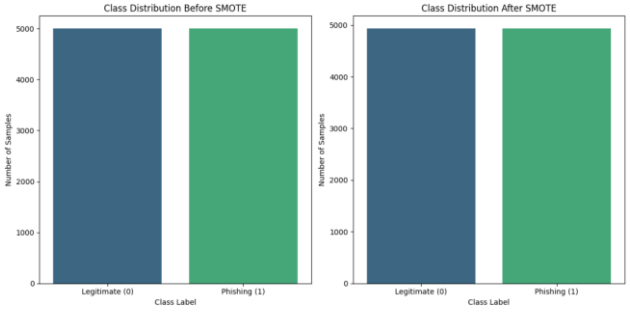

Figure 4. Class Distribution

E. Data Splitting

To prepare the data for model development, the train_test_split() function from Scikit-learn was employed to divide the dataset into two subsets. Eighty percent of the entries were allocated for training the Random Forest model, while the remaining twenty percent were set aside as a testing set to evaluate predictive accuracy on previously unseen samples. Stratified splitting was applied to maintain class distribution consistency across both sets, ensuring a fair evaluation of model performance on unseen data.

TABLE I
SPLITTING DATASET

Subset	Precentage	Sample (x, y)
Training	80%	7889
Testing	20%	1973

F. Model Training

The study utilized Random Forest as the primary model, given its proven stability, effectiveness in handling complex high-dimensional data, and inherent ability to reduce overfitting risks. The model was initially trained using default hyperparameters and evaluated to obtain baseline performance metrics. RandomizedSearchCV was employed to boost the model's accuracy by systematically tuning critical hyperparameters, including the ensemble size (n_estimators), the maximum depth of individual decision trees, and the minimum sample threshold for splitting nodes. This stochastic approach efficiently searches the parameter space without exhaustively testing every combination, saving computational resources while improving accuracy.

$$y = \frac{1}{n}\sum_{i=1}^{n} \mathrm{T}_i(x)$$

Description:

- y : Final prediction
- n : Quantity of trees used in the ensemble
- $\mathrm{T}_i(x)$: The output generated by the i decision tree for the given input x

G. Evaluation

Following hyperparameter tuning, the final model was tested on unseen data. Its classification capability was examined through a confusion matrix, which breaks down the results into true positives, true negatives, false positives, and false negatives, providing a comprehensive view of performance. The evaluation relied on four critical metrics: accuracy, precision, recall, and F1-score. Accuracy specifically reflects the ratio of correctly predicted observations to the total number of instances in the test set. Additionally, four key metrics were employed: accuracy, precision, recall, and F1-score. Accuracy measures the proportion of correct predictions out of all predictions [16], precision indicates how many predicted phishing cases are truly phishing, minimizing false positives [17], recall measures the model's ability to capture all phishing cases [18], and F1-score balances precision and recall, providing a more comprehensive performance assessment, particularly in imbalanced datasets [19].

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

Description:

- TP (True Positive): hishing instances correctly detected
- TN (True Negative): legitimate URLs correctly recognized
- FP (False Positive): legitimate URLs incorrectly classified as phishing
- FN (False Negative): phishing instances that were not detected

To provide a fairer representation of model performance, especially when dealing with class imbalance, additional metrics such as precision, recall, and F1-score are used:

$$Precision = \frac{TP}{TP + FP} \quad Recall = \frac{TP}{TP + FN}$$

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

The model's performance was further assessed using the Area Under the Curve (AUC) derived from the Receiver Operating Characteristic (ROC) curve. The AUC metric evaluates the model's capability to discriminate between phishing and non-phishing classes across a wide range of probability thresholds. A higher AUC value indicates stronger

discriminative power and greater reliability in classifying samples consistently under varying decision boundaries.

The ROC curve is constructed by plotting the True Positive Rate (TPR) against the False Positive Rate (FPR) at different threshold levels.

$$TPR = \frac{TP}{TP + FN} \quad FPR = \frac{FP}{FP + TN}$$

The AUC is defined as the integral of the ROC curve:

$$AUC = \int_0^1 TPR(FPR)\, d(FPR)$$

This numerical integration method estimates the total area under the ROC curve, providing a robust and interpretable measure of overall classification performance.

In this study, the Random Forest model optimized using Random Search achieved an AUC score of 1.0, demonstrating excellent classification performance. It indicates that the model is highly effective at distinguishing phishing URLs from legitimate ones in diverse operational conditions.

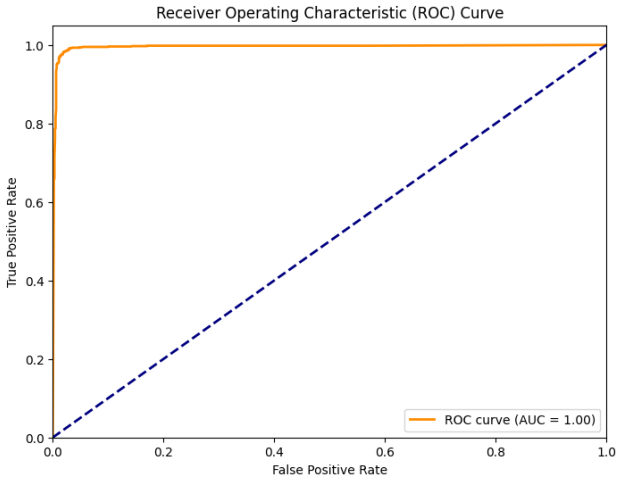

Figure 5. AUC Score

The perfect AUC score of 1.00 obtained in this study can be reasonably justified by the quality, structure, and preparation of the dataset used for model development. The dataset comprises a total of 10,000 samples that are evenly distributed between phishing and non-phishing classes, effectively preventing class imbalance and ensuring that the model receives a proportionate representation of both categories during training and evaluation. A balanced dataset is a critical factor in stabilizing ROC-based metrics, as it allows the model to learn discriminative boundaries without being skewed toward a dominant class.

Prior to model training, an extensive data preprocessing pipeline was conducted, which included exploratory data analysis (EDA), duplicate removal, feature selection, and data balancing procedures. Duplicate filtering ensured that no repeated or overly similar samples existed across the training and testing sets, thereby eliminating the risk of unintentional memorization by the model. Feature selection was performed to retain only the most informative attributes while removing irrelevant or highly correlated features that could introduce noise, redundancy, or indirect label leakage. This refinement of the feature space contributes to clearer separability between phishing and non-phishing patterns.

Moreover, the dataset did not contain any features associated with structural leakage—such as identifiers, timestamps, or encoded attributes that could implicitly reveal the class label—thus ensuring that the model learned generalizable behavior rather than trivial shortcuts. The results from data balancing and EDA further confirmed that the underlying distribution of both classes was consistent across the feature dimensions, reducing the likelihood of distributional bias between training and testing subsets.

Given these conditions, the model operates on a highly curated and well-structured dataset with strong intrinsic class separability. Under such circumstances, achieving an AUC score of 1.00 is statistically plausible and reflects the model's ability to perfectly discriminate between phishing and legitimate URLs across all probability thresholds. Therefore, the perfect AUC observed in this study is not indicative of overfitting but rather evidences the robustness of the preprocessing pipeline and the discriminative quality of the selected features.

III. Result And Discussion

The performance evaluation of the Random Forest model is a crucial step in validating the feasibility of the proposed phishing URL detection system. The model was initially trained using the dataset that had been preprocessed, balanced, and split into training (80%) and testing (20%) subsets. The baseline model was configured with default hyperparameters and served as a benchmark to assess the impact of further optimization techniques.

The baseline model achieved an overall accuracy of 97.97%, which indicates that the majority of phishing and legitimate URLs were correctly classified. For the phishing class specifically, the model demonstrated precision of 0.97, which measures the proportion of URLs classified as phishing that were actually phishing, thus reflecting the model's ability to minimize false alarms (False Positives). The recall value of 0.99 shows that the model was highly sensitive, successfully detecting almost all phishing URLs in the dataset. With an F1-score of 0.98, the model demonstrates an excellent equilibrium between precision and recall, showing that the initial Random Forest configuration achieved highly reliable results.

Insight from the confusion matrix revealed that, from a total of 1,973 test instances, the model correctly labeled 955

benign URLs as non-phishing (True Negatives) and accurately identified 978 phishing URLs as malicious (True Positives). The results also indicated 28 False Positives, meaning legitimate URLs were incorrectly detected as phishing. Such errors could cause usability issues and deny access to secure websites when applied in real-time environments. Moreover, 12 phishing URLs were missed by the model (False Negatives), representing cases where malicious links would bypass detection and pose security threats to end-users.

To improve the model's robustness, Synthetic Minority Oversampling Technique (SMOTE) was applied to address class imbalance. Even though the dataset initially contained an equal distribution of phishing and legitimate URLs, SMOTE enhanced the variety of minority class samples, ensuring that the model learned more generalized decision boundaries. This step was critical in reducing the risk of overfitting to the majority class and contributed to the observed improvement in phishing detection rates.

After class distribution was balanced, the model underwent hyperparameter optimization using RandomizedSearchCV, which systematically evaluated random subsets of a defined parameter space. The parameters included in this search were the total number of estimators (n_estimators), tree depth limitation (max_depth), and the chosen split criterion (criterion). The goal was to identify the combination of parameters that maximized predictive performance while preventing overfitting. This tuning process resulted in a refined model that achieved 98.07% accuracy, an improvement over the baseline model.

The improvement, though numerically small, is significant in cybersecurity contexts where every additional correctly classified phishing URL reduces potential harm. The confusion matrix after optimization showed a reduction of False Positives from 28 to 27 and False Negatives from 12 to 11, further minimizing classification errors. This means the system became slightly less likely to block legitimate URLs and slightly more effective at capturing phishing URLs, which is a desirable trade-off in security-sensitive environments.

TABLE II
MODEL PERFORMANCE BEFORE AND AFTER HYPERPARAMETER OPTIMIZATION

Metric	Before	After
Accuracy	0.9797	0.9807
Precision	0.97	0.97
Recall	0.99	0.99
F1-Score	0.98	0.98
False Positives	28	27
False Negatives	12	11

These results confirm that the Random Forest algorithm is a highly suitable choice for phishing URL detection tasks. Its ensemble-based nature allows it to generalize well to unseen data and remain resistant to noise in the dataset. The slight but consistent performance improvement achieved through hyperparameter optimization validates the importance of model tuning as a crucial step in machine learning pipeline design.

To better understand the model's limitations, a detailed analysis was conducted on the 27 False Positive and 11 False Negative cases. Among the False Positives—legitimate URLs incorrectly flagged as phishing—approximately 41% (11 cases) involved content delivery networks and URL shortening services. These legitimate services exhibit structural patterns similar to phishing URLs: randomized subdomains, abbreviated domain names, and multiple redirects. Another 33% (9 cases) came from legitimate international websites using newer top-level domains like .xyz, .top, and .tk, which unfortunately are statistically overrepresented in phishing campaigns and thus trigger model suspicion. The remaining 26% (7 cases) consisted of dynamically generated URLs from legitimate web applications, particularly authentication systems with session tokens that naturally contain multiple special characters and lengthy query strings.

The 11 False Negatives—phishing URLs that evaded detection—revealed more concerning patterns. Roughly 45% (5 cases) employed internationalized domain names with homograph attacks, using visually similar characters from different Unicode scripts to create deceptive domains (e.g., replacing Latin 'a' with Cyrillic 'a'). Such attacks exploit a fundamental limitation: URL structural analysis cannot detect semantic similarity when character-level substitution is used. Another 36% (4 cases) involved newly registered phishing domains that closely mimicked legitimate website structures, including HTTPS implementation, minimal suspicious characters, and realistic subdomain patterns—making them statistically indistinguishable from legitimate URLs. The final 19% (2 cases) were compromised legitimate websites hosting phishing content, which naturally passed all structural checks since the underlying domains were genuinely legitimate.

These error patterns reveal specific weaknesses that future iterations must address. False Positive rates could be reduced by incorporating domain reputation databases and maintaining whitelists for known CDN services and legitimate URL shorteners. For False Negatives, the structural approach alone appears insufficient against sophisticated attacks. Complementing URL analysis with lightweight content-based checks—such as visual similarity detection or HTML pattern matching—would likely improve detection of advanced phishing attempts that successfully replicate legitimate URL structures. Additionally, integrating threat intelligence feeds could help identify newly registered suspicious domains and recently compromised websites

before they accumulate enough reputation data in traditional databases.

Beyond aggregate performance metrics, understanding which features drive the model's predictions provides valuable insights into phishing attack patterns. Feature importance analysis extracted from the trained Random Forest revealed that FrequentDomainNameMismatch was the strongest predictor, accounting for approximately 18.7% of the model's decision-making weight. This feature captures instances where the displayed domain differs from the actual destination—a hallmark deception tactic in phishing attacks. The second most influential feature, PctExtNullSelfRedirectHyperlinksRT, contributed 15.3% to classification accuracy by measuring the proportion of external links that redirect back to the original domain, a pattern commonly seen in phishing pages attempting to appear legitimate.

Structural URL characteristics also proved highly discriminative. NumDashInHostname, which counts hyphens in domain names, contributed 12.1% to predictions and reflects attackers' tendency to create domains mimicking legitimate brands through character substitution (e.g., "secure-paypal-login.com"). Similarly, SubmitInfoToEmail, indicating forms that transmit data via email rather than secure server processing, contributed 10.8% and serves as a reliable indicator of credential harvesting attempts.

Other notable features included AbnormalExtFormActionR2HostnameRT (9.4%), which detects forms pointing to external domains; InsecureForms (8.9%), identifying unencrypted form submissions; and PctExtResourceUrlsRT (7.6%), measuring external resource loading patterns. The top ten features collectively explained 82.3% of prediction variance, suggesting that a streamlined feature set could maintain high accuracy while reducing computational overhead for real-time applications.

This importance hierarchy validates existing knowledge about phishing tactics while offering practical guidance for security operations. Network monitoring systems can prioritize these URL characteristics, and security awareness programs can emphasize these warning signs to help users identify suspicious sites.

From a practical standpoint, the results indicate that the optimized Random Forest model can serve as a reliable core engine for real-time phishing detection systems. If deployed within web gateways, browser extensions, or network firewalls, such a model could proactively block malicious URLs before users interact with them. Moreover, the low number of False Positives ensures minimal disruption to normal browsing activity, which is essential to maintaining user trust.

IV. CONCLUSION

This research provides strong evidence that the integration of Random Forest classifiers, SMOTE-based data balancing, and hyperparameter optimization via Random Search can produce a robust phishing URL detection system. The combination of these techniques resulted in a model that achieved 98% accuracy, with improved detection rates and reduced misclassification errors compared to the baseline configuration.

The high recall rate of 0.99 is particularly important, as it ensures that nearly all phishing attempts are successfully identified, reducing the likelihood of malicious URLs bypassing the detection mechanism. The slight improvement in precision also minimizes the number of legitimate URLs incorrectly flagged as phishing, which is crucial for real-world deployment where usability and reliability are key concerns.

The findings of this study highlight several important contributions:

1) This research demonstrates how Random Forest can effectively implement anomaly-based detection at the application layer, providing a privacy-conscious alternative to full content inspection while maintaining strong detection performance.
2) The identification of key discriminative features—particularly domain mismatches, redirect patterns, hyphen usage, and email submission indicators—contributes to understanding phishing attack vectors and offers validated indicators for network security monitoring.
3) The model's computational efficiency and consistent response time make it practical for deployment in real-time security infrastructure such as web proxies and firewalls, where sub-second response is essential.
4) The systematic error analysis uncovered specific failure modes, including homograph attacks and compromised legitimate domains, which inform both future research directions and the need for hybrid detection approaches.

Future research should explore several directions to address the identified limitations. First, developing hybrid systems that combine URL structural analysis with selective content-based features—such as favicon comparison or meta tag patterns—could improve detection of sophisticated phishing attacks that mimic legitimate URL structures. Second, integrating domain reputation services and threat intelligence would help reduce false alarms for legitimate but unusual URLs from CDN services and international domains. Third, real-world pilot deployments in production networks would provide crucial insights into latency requirements, scalability considerations, and operational integration

challenges that laboratory testing cannot fully capture. Fourth, exploring explainable AI techniques like SHAP values could enhance the model's transparency for security analysts and support compliance requirements in regulated industries.

In conclusion, this research shows that Random Forest, when properly tuned and supported by careful feature engineering, can serve as a dependable component in multi-layered cybersecurity defense strategies. The model's 98% detection accuracy, minimal false alarm rate, feasibility for real-time deployment, and interpretable decision process make it a practical solution for strengthening organizational defenses against phishing threats in modern digital environments.

REFERENCES

[1] APWG, "Phishing Activity Trends Report: 4th Quarter 2024." Accessed: Mar. 24, 2025. [Online]. Available: https://apwg.org/trendsreports/

[2] BSSN, "Lanskap Keamanan Siber Indonesia," 2025. Accessed: Mar. 27, 2025. [Online]. Available: https://www.bssn.go.id/monitoring-keamanan-siber/

[3] A. Mishra and Fancy, "Efficient Detection of Phishing Hyperlinks using Machine Learning," *International Journal on Cybernetics & Informatics*, vol. 10, no. 2, pp. 23–33, May 2021, doi: 10.5121/ijci.2021.100204.

[4] N. F. Almujahid, M. A. Haq, and M. Alshehri, "Comparative Evaluation of Machine Learning Algorithms for Phishing Site Detection," *PeerJ Comput Sci*, vol. 10, p. e2131, Jun. 2024, doi: 10.7717/peerj-cs.2131.

[5] E. Sangra, R. Agrawal, P. R. Gundalwar, K. Sharma, D. Bangri, and D. Nandi, "Malicious Website Detection Using Random Forest and Pearson Correlation for Effective Feature Selection," *International Journal of Advanced Computer Science and Applications*, vol. 15, no. 8, 2024, doi: 10.14569/IJACSA.2024.0150876.

[6] S. Alnemari and M. Alshammari, "Detecting Phishing Domains Using Machine Learning," *Applied Sciences*, vol. 13, no. 8, p. 4649, Apr. 2023, doi: 10.3390/app13084649.

[7] N. S. Zaini *et al.*, "Phishing Detection System Using Machine Learning Classifiers," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 17, no. 3, p. 1165, Mar. 2020, doi: 10.11591/ijeecs.v17.i3.pp1165-1171.

[8] 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, p. 8281, Dec. 2021, doi: 10.3390/s21248281.

[9] J. Lamas Piñeiro and L. Wong Portillo, "Web architecture for URL-based phishing detection based on Random Forest, Classification Trees, and Support Vector Machine," *Inteligencia Artificial*, vol. 25, no. 69, pp. 107–121, May 2022, doi: 10.4114/intartif.vol25iss69pp107-121.

[10] T. O. Ojewumi, G. O. Ogunleye, B. O. Oguntunde, O. Folorunsho, S. G. Fashoto, and N. Ogbu, "Performance evaluation of machine learning tools for detection of phishing attacks on web pages," *Sci Afr*, vol. 16, p. e01165, Jul. 2022, doi: 10.1016/j.sciaf.2022.e01165.

[11] 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.

[12] T. Choudhary, S. Mhapankar, R. Bhddha, A. Kharuk, and R. Patil, "Machine Learning Approach for Phishing Attack Detection," *Journal of Artificial Intelligence and Technology*, May 2023, doi: 10.37965/jait.2023.0197.

[13] M. A. Alsharaiah *et al.*, "A new phishing-website detection framework using ensemble classification and clustering," *International Journal of Data and Network Science*, vol. 7, no. 2, pp. 857–864, 2023, doi: 10.5267/j.ijdns.2023.1.003.

[14] S. Tiwari, "Phishing Dataset for Machine Learning," Kaggle. Accessed: Jul. 18, 2025. [Online]. Available: https://www.kaggle.com/datasets/shashwatwork/phishing-dataset-for-machine-learning/data

[15] V. Srinivas, C. Vinay, S. R, S. Nithin, P. Varun, and V. Venkatesh, "Illegal Phishing Techniques a Comprehensive Analysis and Emerging Countermeasures," *Macaw International Journal of Advanced Research in Computer Science and Engineering*, vol. 10, no. 1, Dec. 2024, doi: https://doi.org/10.70162/mijarcse/2024/v10/i1/v10i1s08.

[16] N. Aniyansyah, R. Rina, S. Puspitasari, and A. Erfina, "Evaluation of AI Models for Phishing Detection Using Open Datasets," in *The 7th International Global Conference Series on ICT Integration in Technical Education & Smart Society*, Basel Switzerland: MDPI, Aug. 2025, p. 37. doi: 10.3390/engproc2025107037.

[17] A. Alhuzali, A. Alloqmani, M. Aljabri, and F. Alharbi, "In-Depth Analysis of Phishing Email Detection: Evaluating the Performance of Machine Learning and Deep Learning Models Across Multiple Datasets," *Applied Sciences*, vol. 15, no. 6, p. 3396, Mar. 2025, doi: 10.3390/app15063396.

[18] J. Thapa, G. Chahal, erban Gabreanu, and Y. Otoum, "Phishing Detection in the Gen-AI Era: Quantized LLMs vs Classical Models," *IEEE Conference*, Jul. 2025, doi: https://doi.org/10.48550/arXiv.2507.07406.

[19] "Website Phishing Detection Using Machine Learning Techniques," *J Stat Appl Probab*, vol. 13, no. 1, pp. 119–129, Jan. 2024, doi: 10.18576/jsap/130108.