OFERCE: Optimized Rule-Based Detection of Malicious URLs

Dhika Ananda Ramadhan
School of Computing, Telkom University, Bandung, Indonesia
dhikaananda@student.telkomuniversity.ac.id (corresponding author)

Vera Suryani
School of Computing, Telkom University, Bandung, Indonesia
verasuryani@telkomuniversity.ac.id

Received: 19 November 2025 | Revised: 1 December 2025, 26 December 2025, and 5 January 2026 | Accepted: 6 January 2026

 DOI: https://doi.org/10.48084/etasr.16366

ABSTRACT

Cyberattacks using bad URLs are on the rise and are quickly becoming a key means of disseminating contemporary online dangers. Numerous cybersecurity agencies have reported a notable increase in Advanced Persistent Threat (APT), malware, and phishing activities that leverage URLs to spread their attacks. This circumstance emphasizes the importance of creating effective and flexible detection systems to thwart increasingly intricate attack patterns. A previous study proposed a malicious URL detection algorithm based on neural networks that achieved 97% accuracy. However, this model's reliability against dynamic attacks is still limited because it has not been verified using actual network data and still has a high false prediction detection rate (18.13%). To address these limitations, this study proposes OFERCE (Optimized Rule-based Detection for Malicious URLs), an optimized rule-based feature extraction framework that integrates adaptive feature selection based on mutual information, data-balancing strategies (SMOTE and class weighting), and comprehensive lexical rule-based features. Additionally, OFERCE incorporates hyperparameter tuning to ensure that the underlying machine-learning models operate at their optimal configuration, enhancing generalization capability and reducing overfitting during real-world evaluation. According to experimental results, OFERCE improves the performance of the Logistic Regression model to 99% accuracy and reduces the average error detection rate by up to 30%, demonstrating consistent and reliable performance across multiple categories of URL-based threats.

Keywords-malicious url; false-detection; optimised feature; real network traffic

I. INTRODUCTION

A Uniform Resource Locator (URL) is a unique address used to access different types of information on the Internet. However, URLs can potentially be exploited by hackers for harmful purposes. Malicious URLs, which are intentionally created to propagate viruses, steal personal information, conduct fraudulent activities, or deceive users into visiting fake websites, represent one of the most prevalent forms of misuse [1]. Security monitoring systems have reported a significant increase in phishing and scam activities, with more than 709 million attempts to access malicious websites blocked in 2023, representing an increase of more than 40% compared to the previous year [2]. Furthermore, the Indonesian Cyber Security Landscape reported 330,527,636 cyber anomalies in 2024, including 2,487,041 Advanced Persistent Threat (APT) activities, 514,508 ransomware incidents, and 26,771,610 phishing activities [3]. These figures indicate that the complexity and volume of cyberattacks have increased substantially across various sectors, including government institutions, the financial industry, public digital services, and private organizations that rely on web-based infrastructure.

Malicious URLs are widely recognized as one of the primary entry points for the distribution of malware, trojan executables, and APTs across various communication channels. Previous studies have demonstrated that malicious links remain highly effective in compromising users through redirection-based attacks and deceptive web pages [4]. In addition, the structural patterns and delivery mechanisms of phishing-oriented URLs continue to evolve, allowing attackers to bypass conventional detection filters and exploit human vulnerabilities [5]. Email remains the most common non-web-based vector for the dissemination of malicious links. Recent threat intelligence reports indicate a significant increase in embedded URL attacks used for malware delivery, credential harvesting, and multi-stage intrusion campaigns [6].

In [7], it was shown that URL-based attacks continue to achieve high success rates even in academic environments with relatively high levels of user awareness. In this study, more than 2,000 users voluntarily disclosed sensitive information through deceptive links distributed through 8,712 phishing emails. These results indicate that significant gaps remain in cybersecurity awareness and in the effectiveness of early

detection mechanisms across multiple sectors. This observation suggests that malicious URLs remain one of the primary entry points for contemporary cyberattacks. Attackers frequently employ URL obfuscation techniques and multi-stage redirection strategies to conceal malicious intent and increase perceived legitimacy [8]. In addition, many fraudulent URLs leverage HTTPS encryption to further reinforce user trust [9]. To mitigate such threats, Network Detection and Response (NDR) systems are designed to monitor and analyze network activities. However, a fundamental limitation of NDR systems lies in their tendency to generate a high volume of false alerts, increasing the risk of overlooking genuine threats [10].

Using feature-based machine learning techniques, several previous studies in the field of malicious URL identification, especially those focusing on phishing, have demonstrated encouraging results. However, the majority of these techniques are still binary classifications that can only differentiate between phishing and authentic URLs; as a result, they are not yet able to identify more intricate threat varieties such as malware, trojan activity, and APTs. Furthermore, many earlier models relied on static features and small datasets, limiting their ability to adjust to changing assault patterns in the real world. Given these drawbacks, a more contextual and flexible method is required to identify malicious URLs; this method should not only focus on phishing classification but also be able to fully differentiate between malware, trojan activities, and APTs. This work suggests a multi-class malicious URL detection strategy that combines mutual information-based adaptive feature selection, data balancing strategies (SMOTE and class weight), and rule-based lexical features.

The main focus of research on malicious URL detection is the use of feature-based machine learning approaches to detect phishing's structural and behavioral tendencies. In [11], a phishing detection method used four machine learning algorithms, namely Random Forest (RF), Decision Tree, SVM (RBF), and XGBoost, in conjunction with lexical, host-based, and content-based data. According to experimental data, XGBoost achieved the highest balanced performance, with an accuracy of 96.6% and an F1-score of 96.7%. To increase real-time phishing detection rates, this study validated the significance of rich URL features and the choice of the right model. However, this study did not assess the model's performance in real-world attack environments, which are often unbalanced and full of variances in the traits of malicious URLs from other categories, such as malware, trojans, or advanced persistent threats, because it only examined a single structured dataset with a balanced distribution.

In [12], a thorough comparison of eight machine learning algorithms, including Naive Bayes (NB), KNN, MLP, Gradient Boosting (GB), and LR, was conducted using GridSearchCV to systematically optimize hyperparameters, leading to exceptionally high performance. After tuning, RF and GB achieved up to 99.97% accuracy. A richer feature set was produced by this study's incorporation of Whois features, JavaScript properties, geolocation, and domain variants (homoglyphs, bitsquatting, hyphenation). However, the experiment was constrained to a binary scenario (Phishing vs. Benign) and was only carried out on a tiny part of a big dataset (12,000 samples out of 1.56 million URLs) without cross-dataset testing or validation against more complicated threat classes. Furthermore, all of the trials were carried out on very clean data, which did not represent the real-world adversarial URLs or noisy situations that frequently arise on real networks.

In [13], a phishing URL detection framework was based on a dual-model architecture, combining a BERT-based Deterministic Neural Network and a Probabilistic Neural Network. The study utilized a large, balanced dataset consisting of 220,000 phishing URLs and 220,000 benign URLs collected from multiple sources. The deterministic model achieved an accuracy of 97% across both validation and production environments; however, it produced a relatively high false prediction rate, with 18.13% of outputs classified incorrectly, indicating a substantial presence of false positives and false negatives. Although the probabilistic model reduced certain errors by incorporating prediction uncertainty, the results highlight an inherent trade-off between accuracy and misclassification risk, emphasizing the need for further research on reducing false predictions without compromising generalization or interpretability.

CTI-MURLD [14] is a malicious URL detection framework that integrates external Cyber Threat Intelligence (CTI) sources, including Google search results and Whois information, to improve detection accuracy beyond conventional lexical-based approaches. This model employs a two-stage ensemble architecture, where multiple RF classifiers trained on URL, Google-based CTI, and Whois-based CTI features are combined using an MLP for final classification. Experimental results on a public dataset of 20,000 URLs demonstrated strong performance, achieving an accuracy of 96.80% and reducing the false positive rate to 3.13%, representing a significant improvement over traditional URL-based models. However, the reliance on external CTI sources introduces potential risks related to outdated or unreliable information, despite the observed reduction in false positives.

Existing methods still have significant false prediction rates and poor generalization across heterogeneous datasets, despite a great deal of work on phishing and malicious URL identification. Moreover, the combination of adaptive feature selection, multi-class balance, and lexical heuristic rules has received little attention. This disparity drives the creation of OFERCE.

II. METHODOLOGY

Through the integration of multi-level optimization at the data, feature, and model stages, this work seeks to enhance the performance of malicious URL detection. The proposed framework integrates data balancing, adaptive feature selection, model parameter optimization, and OFERCE. In NDR systems, the high rate of false positives and negatives, along with data imbalance between classes, are prominent problems that are intended to be addressed by this method.

Data collection, data preprocessing, feature engineering, feature selection, data balancing, and machine learning model training and optimization are the six primary phases of this study. Figure 1 shows the comprehensive workflow of the suggested method.

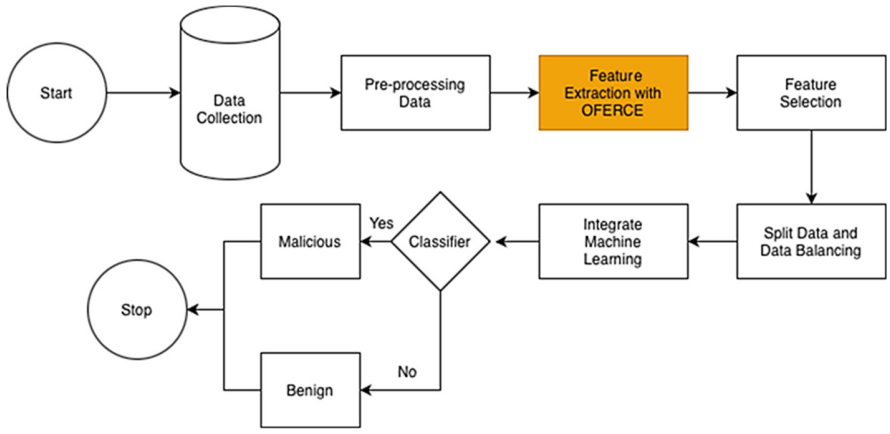

Fig. 1. Proposed method.

A. Dataset

This study utilizes three datasets to evaluate the proposed malicious URL detection framework. The first dataset is the Mendeley Phishing URLs Dataset [15], which contains 450,176 URL samples. The dataset consists of two classes, Benign and Phishing, and is widely used as a benchmark for phishing detection research. The second dataset is a Malicious URLs Dataset [16], consisting of 651,191 URLs categorized into Malware, Phishing, Defacement, and Bnign classes. This dataset aggregates URLs from multiple open-source feeds and blacklist repositories, making it suitable for evaluating multi-class malicious URL classification models.

The third dataset was obtained from the National Cyber and Crypto Agency through an official institutional request [17]. It represents real-world network traffic continuously collected between January and December 2024 and includes 1,177,520 anonymized URL records derived from national DNS sinkhole monitoring systems and threat intelligence correlation engines. Due to national cybersecurity policies, this dataset is not publicly accessible. All entries were fully anonymized and contained only URL strings and associated threat labels. The dataset was validated by agency analysts using multi-source threat intelligence correlation, heuristic inspection, and signature-based filtering techniques. Unlike the public datasets, this dataset reflects inherently imbalanced real-world traffic distributions and includes four categories: Malware, Trojan activity, APTs, and Benign traffic.

B. Data Preprocessing

To ensure data consistency and integrity, all URLs undergo preprocessing before feature extraction. This stage aims to standardize textual representations and reduce noise that may negatively affect model performance. The preprocessing procedures applied in this study include text normalization, stopword removal, stemming, and vectorization. All URL strings were first converted to lowercase, and non-alphabetic characters were removed to achieve a uniform textual format. Subsequently, stopwords and other non-informative terms were eliminated to reduce dimensionality and improve feature relevance. Stemming was performed using the Porter stemming algorithm to normalize word variations into their base forms.

Textual features were then transformed into numerical representations using the Term Frequency-Inverse Document Frequency (TF-IDF) vectorization technique. In this study, TF-IDF was applied using a character-level n-gram range of 3 to 5, which enables the model to capture substring patterns commonly observed in malicious or obfuscated URLs.

C. OFERCE

The OFERCE framework incorporates a compact set of six rule-based lexical and statistical features to capture obfuscation behavior, structural anomalies, and hazardous patterns commonly found in malicious URLs. Conventional feature-based detection methods typically rely on simple lexical indicators, such as the presence of protocol tokens or special characters, which are insufficient to counter modern malware delivery and obfuscation techniques. OFERCE addresses this limitation by introducing context-aware rules that better reflect real-world attack behaviors.

The OFERCE feature set includes indicators for HTTPS usage, combined HTTP and "www" patterns, counts of special characters associated with obfuscation, Shannon entropy to measure randomness, detection of executable file extensions commonly linked to malware distribution, and identification of encoded substrings resembling Base64. These features are designed to reveal deceptive patterns used in phishing, malware, trojan, and APT campaigns, including payload concealment and command-and-control redirection techniques.

Algorithm 1 presents the proposed OFERCE-based detection model. By complementing TF-IDF-based statistical features, OFERCE provides additional semantic context that is particularly effective for short, truncated, or heavily obfuscated URLs, where traditional n-gram representations often become sparse.

```
Algorithm 1: OFERCE
1:  Input: URL dataset D = {URL1, URL2, …,
    URLn}
2:  Output: Feature matrix F representing
    rule-based features for each URL
3:  Initialize empty feature matrix F
4:  For each URL in dataset D:
5:    Let Label ← Label_Mapping[ 'Class' ]
```

```
    if "http" not in URL or "www" not in
    URL then
      http_www ← 0
    else
      http_www ← Label
    if "https" in URL then
      has_https ← 0
    else
      has_https ← Label
    Define special_chars ←"  -_@?%....#+!"
    count ← number of characters in URL
    that belong to special_chars
    count_special_chars ← count
    Define exe_pattern ← { '.exe' ,⋯,.jar' }
    if URL contains any extension in
    exe_pattern then
      has_exec_extension ← Label
    else
      has_exec_extension ← 0
    Define base64_pattern ← sequences
    matching regex ([A-Za-z0-9+/]{20,}={0,2})
    if any valid base64 token exists in
    URL with length multiple of 4 then
      has_encoded_token ← Label
    else
      has_encoded_token ← 0
    Calculate the probability pi of each
    unique character in URL
    entropy ← -Σ(pi * log2(pi)) for
    all characters i
    entropy ← round(entropy, 4)
    F.append([http_www,has_https,
    count_special_chars, has_exec_extension,
    has_encoded_token, entropy])
Return F
```

D. Feature Selection

This study employs the mutual information approach (mutual_info_classif) to minimize feature dimensions while preserving model efficiency. This approach chooses a handful of characteristics with the highest information value after calculating the degree of dependence between each feature and the target label. As a result, only characteristics that are actually pertinent to the classification process are kept for model training, which lowers complexity and the possibility of overfitting. Figure 2 shows a feature selection example.

E. Split Data and Data Balancing

To ensure an objective evaluation, the dataset was divided into training and testing subsets before any balancing procedure, adopting a 70:30 training/testing split ratio. This configuration provides a reliable compromise between learning capacity and generalization performance and is commonly applied in classification-based machine learning studies.

Given the inherent class imbalance in real-world URL traffic, two complementary data balancing strategies were applied to the training set:

- Synthetic Minority Oversampling Technique (SMOTE): This technique generates synthetic samples for minority classes through linear interpolation between a minority instance and one of its k-nearest neighbours, as formulated in (1), where x_{new} denotes the generated synthetic sample, x_i represents a minority-class instance, and x_{zi} refers to one of its k-nearest neighbours ($k = 5$ in this study). The parameter $\lambda \in [0,1]$ is a random interpolation coefficient controlling the position of the synthetic sample. This approach increases minority-class representation without duplicating existing samples, thereby mitigating overfitting while preserving the original data distribution.

$$x_{new} = x_i + \lambda \times (x_{zi} - x_i) \quad (1)$$

- Class Weight Adjustment: To further mitigate bias toward majority classes, class weighting was applied during model training. The weighting scheme is defined in (2), where w_j denotes the weight assigned to class j, $n_{samples}$ represents the total number of samples in the dataset, $n_{classes}$ denotes the total number of classes, and n_j corresponds to the number of samples belonging to class j. This scheme assigns higher importance to underrepresented classes, ensuring proportional contribution during optimisation.

$$w_j = \frac{n_{samples}}{n_{classes} \times n_j} \quad (2)$$

The combined use of SMOTE and class weighting enhances minority-class detection while preserving model stability under highly imbalanced URL traffic conditions.

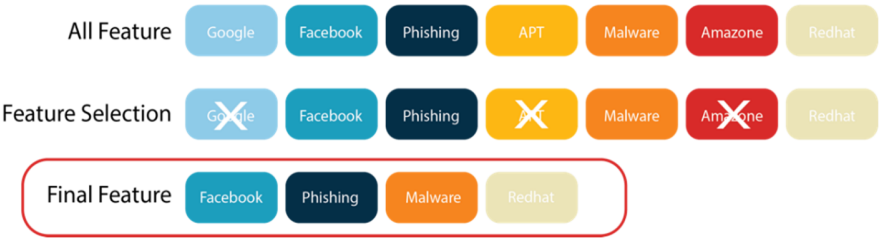

Fig. 2. Feature selection process.

F. Machine Learning

This study employed three machine learning algorithms: Random Forest (RF), Naïve Bayes (NB), and Logistic Regression (LR). To maximize speed and minimize overfitting, model hyperparameters were optimized using the Grid Search Cross Validation (GridSearchCV) technique. The optimization aim is optimized as:

$$\theta^* = \arg\min_{\theta\in\Theta} \frac{1}{k}\sum_{i=1}^{k} L\left(f_\theta\left(X_i^{(val)}\right), y_i^{(val)}\right) \quad (3)$$

In the hyperparameter optimization phase, GridSearchCV with 5-fold cross-validation was used to adjust the model parameters θ. The aim is to find the ideal parameter configuration θ^* that minimizes the average validation loss L. For each classifier, grid search examined a predetermined search space: Random Forest used *n_estimators* = 100, *max_depth* = 500, *min_samples_split* = 50, *min_samples_leaf* = 50, and *max_features* = auto, sqrt, log2; Naïve Bayes used *alpha* = 10, and Logistic Regression used *C* = 0.01, *penalty* = l2, and *solver* = lbfgs, saga. These parameters were chosen to strike a balance between inference efficiency, training stability, and model complexity. θ^* was chosen as the configuration that yielded the best validation performance and the lowest cross-validated loss.

G. Performance Evaluation

The model's performance was assessed using the following metrics:

- Accuracy: Calculates the overall percentage of accurate forecasts compared to the total number of samples.
- F1-score, precision, and recall: Assess each class's classification performance, paying particular attention to the model's capacity to identify minority classes like trojan activity and APT.
- False Detection Rate (FDR): This measure evaluates how well the model minimizes classification errors by taking into account both false positives and false negatives.

III. EXPERIMENTAL RESULTS

This study used three different datasets [15-17] to evaluate the effectiveness of the OFERCE detection method, chosen to assess the system's ability to generalize across a range of threat attributes and data sources. The dataset in [17] was gathered from actual network traffic under a national cyber surveillance environment, and the datasets [15, 16] are publicly available benchmarks frequently utilized in malicious URL detection research. Thus, the assessment outcomes obtained on the dataset in [17] offer a more accurate representation of the system's performance in actual traffic scenarios.

Performance was measured and compared before and after the OFERCE feature engineering module was integrated using NB, RF, and LR. The three datasets were used to train and validate each model, evaluating its efficacy and generalizability across various data distributions. Each dataset was subjected to a methodical feature engineering procedure before training, which included important lexical and statistical characteristics such as *has_https*, *has_http_www*, *count_special_chars*, *has_exec_extension*, *has_encoded_token*, and *entropy*.

A. Mendeley Dataset [15]

As shown in Tables I, II, and III, OFERCE consistently improves model performance on this dataset [15] across all classifiers. Before optimization, the Phishing class only achieved an F1-score of 54-56%, demonstrating instability in identifying malicious URLs, while LR and RF both performed moderately on the Benign class, each reaching about 60% accuracy. After using OFERCE, the Phishing F1-score of LR increased from 54.46 to 98.10, and its accuracy increased from 60.98% to 99.13%. Similarly, the Phishing F1-score of RF increased from 51.80 to 97.58, along with its accuracy from 56.49% to 98.87%. After optimization, NB, which had earlier yielded inconsistent results, especially in the Phishing class, greatly improved, increasing its F1-score from 44.82 to 98.71% and achieving accuracy above 99% for both classes. All three models achieved near-perfect accuracy and F1-scores during optimization, demonstrating that OFERCE successfully improves classification performance on this dataset by increasing feature relevance and decreasing false detections.

B. Kaggle Dataset [16]

As shown in Tables IV, V, and VI, OFERCE significantly enhanced all models. The Benign and Phishing classes experienced high error rates before optimization; for instance, RF reported an F1-score of only 35.37 for Phishing, while LR only managed 56.84% accuracy with an FDR of 43.06% on the Benign class. Following the use of OFERCE, for the Phishing class, RF attained a 94.88 F1-score and just 1.49% FDR, whereas LR improved its Benign accuracy to 99.31% and decreased FDR to 0.69%. After optimization, NB increased its Phishing F1-score from 30.07 to 81.55%. Therefore, OFERCE continuously improves accuracy, increases F1-score in every class, and significantly reduces FDRs, demonstrating its efficacy in bolstering malicious URL classification on [16].

C. The Cyber and Crypto Agency Dataset [17]

Tables VII, VIII, and IX summarize the evaluation using the dataset in [17], demonstrating that OFERCE offers significant performance improvements across all models. Before optimization, all classifiers had severe difficulties, particularly in the Trojan Activity and Malware classes, where FDRs were quite high. For instance, RF recorded 33.23% accuracy, whereas LR only managed 35.01% accuracy and an F1-score of 33.50 on Trojan Activity. Even more instability was displayed by NB, as its Trojan Activity F1-score fell to 2.92. After optimization with OFERCE, across all classes, LR achieved more than 99.9% accuracy and almost 100% F1-score, bringing FDR down to almost 0%. RF also had a significant improvement, with F1-score increasing from 32.92 to 98.40 and accuracy increasing from 33.23% to 99.48%. Despite being the weakest of the three models, NB demonstrated notable improvements in the Benign and Trojan Activity classes, reaching up to 96.78% accuracy and a Benign F1-score of 95.38. These results show that OFERCE works quite well in unbalanced, real-world situations. All classifiers can function more dependably across all malicious URL categories due to the optimized feature set and balancing method, which also dramatically reduces false detections and stabilizes model performance.

TABLE I. LOGISTIC REGRESSION WITH MENDELEY DATASET [15]

Class	ACC Before	ACC After	F1 Before	F1 After	FDR Before	FDR After
Benign	60.98%	99.13%	65.87	99.43	39.02%	0.87%
Phishing	60.98%	99.13%	54.46	98.10	39.02%	0.87%

TABLE II. RANDOM FOREST WITH MENDELEY DATASET [15]

Class	ACC Before	ACC After	F1 Before	F1 After	FDR Before	FDR After
Benign	56.49%	98.87%	60.36	99.26	43.51%	1.13%
Phishing	56.49%	98.87%	51.80	97.58	43.51%	1.13%

TABLE III. NAÏVE BAYES WITH MENDELEY DATASET [15]

Class	ACC Before	ACC After	F1 Before	F1 After	FDR Before	FDR After
Benign	83.09%	99.40%	90.02	99.61	16.91%	0.60%
Phishing	83.09%	99.40%	44.82	98.71	16.91%	0.60%

TABLE IV. LOGISTIC REGRESSION WITH KAGGLE DATASET [16]

Class	ACC Before	ACC After	F1 Before	F1 After	FDR Before	FDR After
Benign	56.84%	99.31%	51.66	99.48	43.16%	0.69%
Defacement	95.58%	98.23%	83.06	93.90	4.42%	1.77%
Phishing	50.87%	97.52%	36.24	91.33	49.13%	2.48%
Malware	97.15%	99.84%	62.38	98.34	2.85%	0.16%

TABLE V. RANDOM FOREST WITH KAGGLE DATASET [16]

Class	ACC Before	ACC After	F1 Before	F1 After	FDR Before	FDR After
Benign	54.77%	99.09%	47.99	99.31	45.23%	0.91%
Defacement	95.48%	99.20%	82.70	95.86	4.52%	0.80%
Phishing	48.77%	98.59%	35.37	94.88	51.23%	1.41%
Malware	97.09%	99.70%	61.89	96.92	2.91%	0.30%

TABLE VI. NAÏVE BAYES WITH KAGGLE DATASET [16]

Class	ACC Before	ACC After	F1 Before	F1 After	FDR Before	FDR After
Benign	80.06%	98.96%	86.46	99.21	19.94%	1.04%
Defacement	95.62%	98.99%	83.08	88.42	4.38%	1.01%
Phishing	86.38%	93.91%	30.07	81.55	13.62%	6.09%
Malware	97.29%	97.30%	65.32	65.32	2.71%	2.70%

TABLE VII. LOGISTIC REGRESSION WITH THE NATIONAL CYBER AND CRYPTO AGENCY DATASET [17]

Class	ACC Before	ACC After	F1 Before	F1 After	FDR Before	FDR After
Benign	78.52%	99.98%	58.16	99.98	21.48%	0.02%
Malware	58.03%	99.99%	14.10	99.99	41.97%	0.01%
Trojan	35.01%	99.99%	33.50	99.99	64.99%	0.01%
APT	98.41%	99.99%	23.02	99.98	1.59%	0.01%

TABLE VIII. RANDOM FOREST WITH THE NATIONAL CYBER AND CRYPTO AGENCY DATASET [17]

Class	ACC Before	ACC After	F1 Before	F1 After	FDR Before	FDR After
Benign	77.06%	96.16%	53.99	94.44	22.94%	3.84%
Malware	57.74%	96.13%	12.94	95.91	42.26%	3.87%
Trojan	33.23%	99.48%	32.92	98.40	66.77%	0.52%
APT	98.42%	99.52%	22.98	88.30	1.58%	0.48%

TABLE IX. NAÏVE BAYES WITH THE NATIONAL CYBER AND CRYPTO AGENCY DATASET [17]

Class	ACC Before	ACC After	F1 Before	F1 After	FDR Before	FDR After
Benign	38.54%	96.78%	52.81	95.38	61.46%	3.22%
Malware	58.03%	81.82%	14.09	77.16	41.97%	18.18%
Trojan	81.71%	83.33%	2.92	65.48	18.29%	16.67%
APT	98.42%	98.21%	22.98	27.72	1.58%	1.79%

IV. CONCLUSION

This study used three diverse malicious URL datasets [15-17] to assess the efficacy of the OFERCE framework across three machine-learning models: LR, RF, and NB. The findings consistently show that OFERCE significantly improves classification performance in both extremely imbalanced and balanced scenarios. Significant flaws were found in the baseline models across all datasets, especially in minority or high-variance classes, including Trojan Activity, Phishing, and Malware. In a number of instances, the FDRs were noticeably high. For instance, NB showed great instability with extremely low F1-scores, while LR and RF recorded FDR values that exceeded 40–60 % for malware-related classes on [16, 17]. All models showed significant improvements with the application of OFERCE. Across almost all classes and datasets, LR consistently reached more than 99% accuracy and 99% F1-score, demonstrating the greatest consistent gains. Additionally, RF made tremendous progress, especially in the Trojan Activity class, where accuracy on the dataset in [17] rose from as low as 33% to over 99%. Despite having the least stable baseline, NB demonstrated significant post-optimization gains, particularly in the datasets in [15, 16], increasing previously low F1-scores to 80–98% and significantly lowering erroneous detections.

Overall, OFERCE's mix of rule-based lexical features, adaptive mutual-information feature selection, explicit data balancing, and hyperparameter optimization successfully improves model dependability, as evidenced by the steady performance gains observed across all three datasets. In addition to lowering FDRs for a variety of threat categories, the approach also stabilizes classifier behavior in the face of massively unbalanced, real-world malicious URL distributions. These findings demonstrate that OFERCE is a reliable and broadly applicable method for detecting malicious URLs in the modern era.

V. NOVELTY AND CONTRIBUTIONS

The novelty of this work lies in the integration of a lightweight rule-based lexical framework, OFERCE, with adaptive mutual-information feature selection and explicit data-balancing strategies for multi-class malicious URL detection. Unlike most existing approaches that focus primarily on binary phishing classification or rely exclusively on statistical n-gram representations, OFERCE introduces interpretable heuristic features designed to capture obfuscation patterns, deceptive structures, and payload delivery indicators commonly observed in malware, trojan, and APT campaigns. The main contributions of this study are threefold: (i) the design of OFERCE as an optimized rule-based lexical feature extraction framework that integrates mutual information-based adaptive feature selection, data-balancing strategies, and systematic hyperparameter tuning to improve model robustness, (ii) a comprehensive evaluation across three heterogeneous datasets, including a large-scale real-world national dataset characterized by extreme class imbalance, and (iii) empirical evidence demonstrating that the proposed framework effectively reduces FDRs and enhances generalization capability while stabilizing the performance of multiple machine-learning classifiers in malicious URL detection.

Overall, the results confirm that OFERCE provides a reliable, interpretable, and scalable solution for malicious URL detection in modern network environments. By improving detection accuracy, reducing false positives, and enhancing robustness under real-world traffic conditions, the proposed framework is well-suited for practical deployment in NDR systems.

ACKNOWLEDGMENT

This study is funded by the Ministry of Communication and Digital of the Republic of Indonesia and Telkom University. The authors sincerely thank the Ministry and Telkom University for their trust and generous support in funding this project. Their assistance was instrumental in the successful completion of this study.

REFERENCES

[1] Q. Abu Al-Haija and M. Al-Fayoumi, "An intelligent identification and classification system for malicious uniform resource locators (URLs)," *Neural Computing and Applications*, vol. 35, no. 23, pp. 16995–17011, Aug. 2023, https://doi.org/10.1007/s00521-023-08592-z.

[2] "Kaspersky reports phishing attacks grow by 40 percent in 2023," *Kaspersky*, Mar. 07, 2024. https://www.kaspersky.com/about/press-releases/kaspersky-reports-phishing-attacks-grow-by-40-percent-in-2023.

[3] *Lanskap Keamanan Siber Indonesia*. Jakarta, Indonesia: Id-SIRTII/CC–BSSN, 2024.

[4] A. Prasad and S. Chandra, "PhiUSIIL: A diverse security profile empowered phishing URL detection framework based on similarity index and incremental learning," *Computers & Security*, vol. 136, Jan. 2024, Art. no. 103545, https://doi.org/10.1016/j.cose.2023.103545.

[5] B. B. Gupta, A. Tewari, A. K. Jain, and D. P. Agrawal, "Fighting against phishing attacks: state of the art and future challenges," *Neural Computing and Applications*, vol. 28, no. 12, pp. 3629–3654, Dec. 2017, https://doi.org/10.1007/s00521-016-2275-y.

[6] "Email Threat Landscape Report: Evolving Threats in Email-Based Attacks," https://www.trendmicro.com/vinfo/us/security/news/threat-landscape/email-threat-landscape-report-evolving-threats-in-email-based-attacks.

[7] D. E. D. Vivas, W. Y. G. Pena, S. P. C. Botero, and A. E. Rojas, "A Controlled Phishing Attack in a University Community: A Case Study," *Journal of Internet Services and Information Security*, vol. 14, no. 3, pp. 98–110, Aug. 2024, https://doi.org/10.58346/JISIS.2024.I2.007.

[8] J. Milletary, "Technical Trends in Phishing Attacks," US-CERT.

[9] S. Udipi, "The event data management problem: getting the most from network detection and response," *Network Security*, Nov. 2021, https://doi.org/10.1016/S1353-4858(21)00008-8.

[10] M. Campfield, "The problem with (most) network detection and response," *Network Security*, Nov. 2021, https://doi.org/10.1016/S1353-4858(20)30104-5.

[11] R. Alzubi, T. Bishtawi, and H. Kassem, "Improving Web Security through Machine Learning: A Feature-Based Methodology for Detecting Phishing URLs," *Engineering, Technology & Applied Science Research*, vol. 15, no. 5, pp. 26845–26851, Oct. 2025, https://doi.org/10.48084/etasr.12015.

[12] A. A. Albishri and M. M. Dessouky, "A Comparative Analysis of Machine Learning Techniques for URL Phishing Detection," *Engineering, Technology & Applied Science Research*, vol. 14, no. 6, pp. 18495–18501, Dec. 2024, https://doi.org/10.48084/etasr.8920.

[13] H. Ghalechyan, E. Israyelyan, A. Arakelyan, G. Hovhannisyan, and A. Davtyan, "Phishing URL detection with neural networks: an empirical study," *Scientific Reports*, vol. 14, no. 1, Oct. 2024, Art. no. 25134, https://doi.org/10.1038/s41598-024-74725-6.

[14] M. Alsaedi *et al.*, "Cyber Threat Intelligence-Based Malicious URL Detection Model Using Ensemble Learning," *Sensors*, vol. 22, no. 9, Apr. 2022, https://doi.org/10.3390/s22093373.

[15] J. K. S. Kaitholikkal and B. Anthi, "Phishing URL dataset." Mendeley Data, Apr. 02, 2024, https://doi.org/10.17632/vfszbj9b36.1.

[16] "Malicious URLs dataset." Kaggle, [Online]. Available: https://www.kaggle.com/datasets/sid321axn/malicious-urls-dataset.

[17] "Cyber threat intelligence dataset 2024." National Cyber and Crypto Agency (private, provided under formal institutional request), 2024.