ISSN: 2473-4901
v12 n6662

Zero-Day Phishing Detection with Lightweight Machine Learning Models

Samuel Aidoo
sa19104@georgiasouthern.edu
Georgia Southern University
GA USA

Hayden Wimmer
hwimmer@georgiasouthern.edu
Georgia Southern University
GA USA

Loreen Powell
lpowell@maryu.marywood.edu
Marywood University
PA USA

Abstract

Phishing attacks continue to represent a persistent and evolving cybersecurity threat, particularly as adversaries increasingly leverage zero-day phishing domains to circumvent traditional blacklist-based detection mechanisms. This study examines the effectiveness of machine learning models in detecting phishing websites using engineered features derived from URLs and domain-level attributes. Three supervised classifiers, Logistic Regression, Random Forest, and Gradient Boosting, were evaluated using two complementary strategies: a conventional 80/20 stratified random train–test split and an activation-time-based zero-day approximation designed to assess model generalization to more recently activated domains. Under the stratified evaluation, Random Forest achieved the highest overall performance, with an accuracy of 95.94%, an F1-score of 95.65%, and a ROC-AUC of 99.52%. Under the activation-time-based evaluation, Random Forest maintained the strongest performance, achieving an accuracy of 86.44%, an F1-score of 89.69%, a ROC-AUC of 95.50%, and a Zero-Day Detection Rate (ZDDR) of 82.26%. All three models exhibited reduced performance under the activation-time-based evaluation compared with the conventional stratified evaluation, highlighting the challenges of generalizing phishing detection models to more recently activated domains. The findings demonstrate the effectiveness of lightweight URL- and domain-level features for phishing detection while emphasizing the importance of evaluation strategies that assess model robustness beyond conventional random data partitioning. Transformer-based architectures are identified as a direction for future research, particularly for scenarios involving raw URL sequences.

Keywords: Machine learning, Phishing Website Detection, Zero-Day Attacks

ISSN: 2473-4901
v12 n6662

Zero-Day Phishing Detection with Lightweight Machine Learning Models

Samuel Aidoo, Hayden Wimmer, and Loreen Powell

1. INTRODUCTION

Phishing remains one of the most persistent and damaging cybersecurity threats, frequently serving as an entry point for fraud schemes that contribute to billions of dollars in annual cybercrime losses (Federal Bureau of Investigation [FBI], 2025). Modern phishing attacks increasingly rely on rapidly generated, short-lived domains designed to evade traditional defenses such as blacklists and signature-based systems. Domains that have been more recently activated can present additional detection challenges because their characteristics may differ from those represented in historical training data because they operate during the window in which most defensive systems have no prior knowledge of them. As adversaries automate domain generation and employ sophisticated obfuscation techniques, the phishing detection problem becomes more complex, requiring approaches capable of generalizing beyond previously observed patterns (Géron, 2022; Haq et al., 2024; Safi & Singh, 2023).

Today, machine learning (ML) techniques have emerged as effective tools for addressing this challenge (Haq et al., 2024; Safi & Singh, 2023). Specifically, ML techniques analyze structural patterns embedded within URLs and domain metadata, such as lexical features, token structures, entropy measures, WHOIS timestamps, and domain age (Bahnsen et al.; Haq et al., 2024; Marchal et al., 2016). Thus, these engineered features capture subtle anomalies indicative of malicious URLs, enabling early-stage detection without requiring webpage retrieval. However, a common methodological limitation in the phishing detection literature is the reliance on random train–test splits for model evaluation, which can artificially inflate performance metrics and fail to reflect real-world deployment conditions (Bahnsen et al.; Ma et al., 2009). Given the rapidly evolving nature of phishing attacks, models trained on historical data should ideally be evaluated using temporally distinct or cross-source datasets to more rigorously assess their generalization to emerging phishing threats (Haq et al., 2024; Marchal et al., 2016). Despite the demonstrated effectiveness of ML models using engineered URL- and domain-level features, there remains a need for studies that systematically evaluate model generalization under more challenging evaluation conditions, particularly when only lightweight, precomputed features are available. This research gap limits the practical applicability of existing models and underscores the need for approaches that prioritize generalization to unseen and evolving phishing threats.

To address this gap, this study evaluates three supervised ML models, Logistic Regression, Random Forest, and Gradient Boosting, using both a standard 80/20 stratified random train–test split and an activation-time-based zero-day approximation based on the time_domain_activation attribute. The goal is to understand how well lightweight models, including Logistic Regression, Random Forest, and Gradient Boosting, can detect previously unseen phishing URLs and whether domain metadata contributes meaningful predictive value. Finally, this study provides insight into the performance of traditional ML models under an activation-time-based approximation of zero-day evaluation and establishes a foundation for future research involving raw URL representations and transformer-based architectures.

2. LITERATURE REVIEW

Phishing attacks continue to be a global cybersecurity concern, targeting millions of users through deceptive websites designed to steal credentials, financial information, or personal data. Traditional defense mechanisms, especially blacklists such as Google Safe Browsing and PhishTank, rely on known reports of malicious websites. While effective for previously detected phishing URLs, blacklists routinely fail against zero-day attacks and new malicious domains that have not yet been reported or indexed. Several studies have shown that blacklists detect less than half of zero-day phishing URLs, highlighting the need for adaptive, predictive systems capable of generalizing beyond known threats.

ML has become a promising alternative because models can learn structural patterns indicative of phishing activity. Early studies such as Ma et al. (2009) Demonstrated that lexical URL features (length,

token composition, character diversity) and domain-level attributes (WHOIS registration, domain age, hosting patterns) could outperform blacklists without requiring webpage content downloads. Subsequent research expanded the feature space to include entropy measures, subdomain depth, DNS TTL values, and registrar behavior, demonstrating that engineered URL and domain features can capture attacker behavior at multiple layers.

Similarly, Marchal et al. (2016) emphasized the role of domain registration behavior and DNS anomalies in phishing detection, arguing that attackers often exploit domain-age patterns and registrar inconsistencies. Their goal was to assess whether domain-level features such as WHOIS attributes, DNS TTL values, and registrar profiles could enhance detection performance. They constructed datasets from phishing feeds and Alexa top-ranked domains, then applied classifiers including SVM and Random Forest. Their results showed that domain-based models achieved 92% accuracy, outperforming URL-only techniques and excelling on zero-day samples. This paper directly supports the current study's inclusion of domain-age, activation timestamps, and WHOIS fields in evaluating zero-day generalization.

Sahoo et al. (2017) provided a comprehensive survey of malicious URL detection methods and identified fundamental limitations in evaluation techniques commonly used in the literature. Their objective was to compare ML, deep learning, blacklist-based, and hybrid methodologies while analyzing challenges such as temporal drift and cross-source variability. Through extensive review, they demonstrated that models achieving above 97% accuracy under random splits often dropped to 85–88% when tested using temporal splits. This finding is crucial to the present study because it justifies adopting temporal zero-day evaluation and demonstrates why random data splits inflate performance metrics.

Additionally, Bahnsen et al. (2017) applied DL to phishing URL detection by training character-level recurrent neural networks (RNNs) that learn patterns directly from raw URL strings. Their objective was to evaluate whether RNNs could detect phishing URLs without manual feature engineering. Using millions of URLs and temporal test splits, they trained RNN models and benchmarked them against classical methods. Their results demonstrated strong performance with a 93% F1-score under temporal evaluation, showing robustness against obfuscation techniques such as homograph attacks.

Today, more recent literature has highlighted the importance of evaluating ML systems under realistic conditions. Temporal drift changes in phishing patterns over time significantly impact detection accuracy. Specifically, Safi and Singh (2023) conducted a systematic literature review of 120 phishing detection studies to assess methodological soundness across the field. Their objective was to evaluate dataset practices, features, and evaluation strategies used in phishing research from 2015 to 2022. Their review found that many studies relied heavily on random train-test splits, leading to inflated accuracy metrics and poor real-world applicability. Their results emphasized the need for temporal, cross-source, and zero-day validation frameworks.

Additionally, research by Mahdaouy et al. (2024) introduced DomURLs-BERT, a transformer-based model capable of detecting malicious URLs using domain-aware segmentation and contextual attention mechanisms. Their objective was to leverage BERT's self-attention architecture to learn semantic relationships between URL components. They trained the model on large URL corpora and compared its performance to CNN and LSTM baselines. Results showed that DomURLs-BERT achieved 98% accuracy and offered interpretable attention maps.

Additionally, Rashid et al. (2024) focused on the challenge of cross-source generalization, noting that models trained on one phishing dataset frequently underperform on another due to distribution shifts. Their objective was to apply unsupervised adversarial domain adaptation to align feature distributions between datasets such as OpenPhish and PhishTank. They trained deep models before and after adaptation and evaluated cross-source performance. Their results reduced the accuracy loss between sources from 25% to just 10%. Thus, they offered insights into potential future improvements using domain adaptation.

Moreover, Haq et al. (2024) developed a DL model using a convolutional neural network (CNN) trained on multi-source phishing URLs and Tranco-ranked benign URLs. Their objective was to improve phishing detection accuracy using character-level and token-based representations extracted from raw URLs. Their method involved collecting data from multiple phishing feeds, preprocessing URLs, and training CNN models. Their results showed an accuracy of 98.2% and an F1-score of 0.96, outperforming Random Forest and SVM classifiers.

ISSN: 2473-4901
v12 n6662

Finally, Opara et al. (2024) proposed WebPhish, a hybrid phishing detection framework combining raw URL attributes with lightweight HTML-based features. Their objective was to enhance real-world phishing detection by integrating multi-modal signals while maintaining computational efficiency. Using Random Forest and Gradient Boosting models, they evaluated the system on hybrid URL–HTML datasets. Their results demonstrated 97% precision and strong generalization to unseen phishing URLs. Thus, they also illustrated how hybrid approaches can strengthen detection robustness and inform future directions for integrating additional feature modalities.

Given today's landscape, URL and domain-level engineered features remain an important and efficient avenue for phishing detection, especially for systems that must operate in real time and at scale. The goal of this study is to build on the strengths of these features while addressing methodological weaknesses in prior work by incorporating both stratified and activation-time-based evaluations to examine model generalization under more challenging conditions.

3. METHODOLOGY

This study employs a comparative experimental methodology to examine the effectiveness of supervised machine learning approaches for phishing domain detection using engineered URL and domain-level features. The methodological process progresses through several interconnected stages, beginning with data acquisition and preprocessing to ensure the consistency and usability of the dataset. This is followed by correlation analysis and feature reduction to eliminate redundant predictors and reduce the dimensionality of the feature space. Three supervised classifiers, Logistic Regression, Random Forest, and Gradient Boosting, are subsequently developed and evaluated under two complementary experimental conditions: a standard 80/20 stratified random train–test split and an activation-time-based zero-day approximation. The latter uses the time_domain_activation attribute to evaluate model generalization from longer-established domains to more recently activated domains. The goal of this methodology is to determine how effectively lightweight machine learning models operate on structured lexical and domain-level indicators generalize under conventional and activation-time-based evaluation conditions. Collectively, these procedures support the assessment of phishing detection performance and robustness while providing an activation-time-based approximation of zero-day generalization.

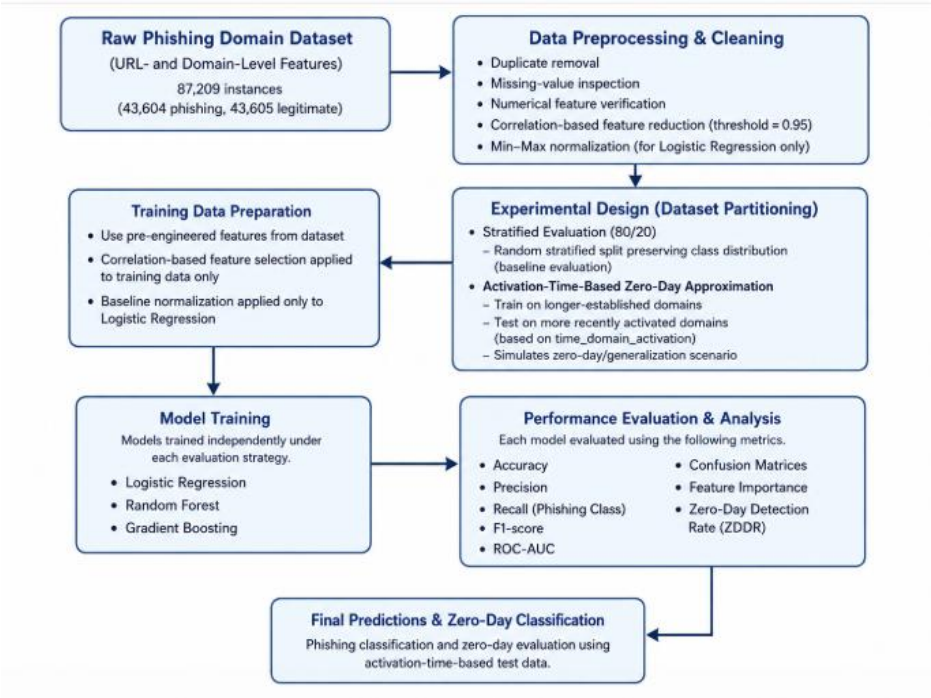

Figure 1. Workflow diagram

Figure 1 summarizes the experimental workflow used in this study. The workflow begins with dataset acquisition and preprocessing, including data cleaning, duplicate removal, correlation-based feature reduction, and numerical feature normalization where applicable. The cleaned dataset is then evaluated using two complementary strategies: an 80/20 stratified random train–test split and an activation-time-based zero-day approximation based on the time_domain_activation attribute. Three supervised machine learning classifiers Logistic Regression, Random Forest, and Gradient Boosting are trained and evaluated using accuracy, precision, recall, F1-score, ROC-AUC, confusion matrices, and feature-importance analysis. The activation-time-based evaluation additionally uses the Zero-Day Detection Rate (ZDDR) to assess model generalization to more recently activated domains.

3.1 Research Design

A quantitative and comparative experimental research design forms the foundation of this study. The research evaluates the performance of three supervised machine learning models, Logistic Regression, Random Forest, and Gradient Boosting, using engineered URL and domain-level features. A transformer-based architecture was not evaluated in the present study and is discussed as a direction for future research. To assess model performance and generalization, two evaluation strategies were implemented. The first employed an 80/20 stratified random train–test split to establish baseline classification performance under conventional experimental conditions. The second employed an activation-time-based zero-day approximation using the time_domain_activation attribute to evaluate generalization from longer-established domains to more recently activated domains. Together, these evaluation strategies provide complementary evidence of model performance under conventional and more challenging deployment-oriented conditions.

3.2 Data Description

The dataset utilized in this study is obtained from Michelle VP's (2024) publicly available *"Phishing Domain Detection | CyberSecurity"* dataset hosted on Kaggle. The downloaded dataset contained 129,698 labeled observations and 112 columns, comprising 111 predictor variables and one binary target variable (phishing). The target variable represents legitimate websites as 0 and phishing websites as 1. Before preprocessing, the dataset contained 77,546 legitimate observations (59.79%) and 52,152 phishing observations (40.21%).

The dataset consists of engineered features describing URL structure, directory and file characteristics, domain and DNS information, domain registration characteristics, SSL/TLS certificate status, redirects, indexing information, and other attributes relevant to phishing detection. The feature structure corresponds to the phishing website, which contains 111 engineered predictors. Their documentation defines time_domain_activation as the number of days since domain activation and time_domain_expiration as the number of days until domain expiration.

The original dataset's authors report that the underlying website observations were obtained from publicly available lists of phishing and legitimate websites. Verified phishing websites were obtained from the PhishTank registry, whereas legitimate websites were obtained from publicly available community-maintained lists and Alexa-ranked websites.

During data-quality inspection, 42,489 exact duplicate observations were identified and removed. The resulting modeling dataset contained 87,209 unique observations, comprising 56,712 legitimate websites (65.03%) and 30,497 phishing websites (34.97%).

3.3 Data Preprocessing

The dataset was first examined for duplicate observations, conventional missing values, feature data types, class distribution, and unavailable temporal values. No conventional missing (NaN) values were identified in the downloaded data. However, 42,489 exact duplicate observations were detected and removed, reducing the dataset from 129,698 to 87,209 unique observations. The cleaned dataset contained 56,712 legitimate observations and 30,497 phishing observations. No over-sampling, under-sampling, or artificial class balancing was performed; consequently, the cleaned dataset's natural class distribution was preserved.

All 111 predictor variables were numerically represented in the downloaded dataset; therefore, additional one-hot encoding was not required in the implemented experiment. The domain temporal

variables were also retained in their original numerical representation. In the cleaned dataset, time_domain_activation ranged from -1 to 17,775, and time_domain_expiration ranged from −1 to 22,574. The value −1 was treated as unavailable temporal information when constructing the activation-time-based evaluation.

Feature reduction was conducted independently on the training partition of each experiment to prevent test-set information from influencing predictor selection. Pearson correlation analysis was applied to the training predictors, and pairs with an absolute correlation coefficient greater than 0.90 were identified. For highly correlated pairs, Random Forest feature-importance values were used to retain the more informative predictor. In the stratified experiment, this procedure reduced the predictor space from 111 to 71 features.

Min-Max normalization was fitted exclusively to the training partition for Logistic Regression and subsequently applied to the corresponding test partition. Random Forest and Gradient Boosting were trained using the reduced numerical predictors without Min–Max scaling. Fitting preprocessing operations only on training data prevented information leakage from the test set.

3.4 Min–Max Normalization

Continuous numerical features were normalized using:

$$x \quad = \frac{x - min(x)}{max(x) - min(x) + \varepsilon}$$

This ensured scale uniformity across attributes such as URL length, entropy, and domain age.

3.5 Experimental Design

A quantitative comparative experimental design was adopted to evaluate the performance of three lightweight supervised machine learning classifiers, Logistic Regression, Random Forest, and Gradient Boosting, for phishing domain detection under two complementary evaluation settings. The first evaluation employed a conventional 80/20 stratified random train–test split to establish baseline classification performance. The second evaluation employed an activation-time-based zero-day approximation, designed to evaluate the models' ability to generalize from longer-established domains to more recently activated domains. Using two evaluation strategies provides a more comprehensive assessment of model performance under both conventional classification conditions and a deployment-oriented testing scenario.

For the stratified random evaluation, the cleaned dataset containing 87,209 unique observations was partitioned using an 80/20 stratified random train–test split with a fixed random seed (random_state = 42) to ensure reproducibility. Stratified sampling preserved the class distribution of the cleaned dataset in both partitions. Consequently, the training comprised 69,767 observations, including 45,369 legitimate and 24,398 phishing domains, while the independent test set contained 17,442 observations, consisting of 11,343 legitimate and 6,099 phishing domains. Both subsets therefore maintained the overall class distribution of approximately 65.03% legitimate and 34.97% phishing observations. To reduce feature redundancy, correlation-based feature reduction was performed exclusively on the training partition. Predictor pairs with an absolute Pearson correlation coefficient greater than 0.90 were identified, and the less informative feature within each correlated pair was removed using Random Forest feature-importance scores. This procedure reduced the predictor space from 111 to 71 features before model training. Conducting feature selection solely on the training data prevented information leakage into the independent test set.

For the activation-time-based zero-day evaluation, the time_domain_activation attribute was used as an activation-age indicator. According to the original dataset documentation, this feature represents domain activation time measured in days rather than an absolute calendar timestamp. Consequently, the second experiment is interpreted as an activation-time-based approximation of zero-day detection rather than a strict chronological evaluation based on observation dates. Observations with unavailable activation information (time_domain_activation = −1) were excluded, leaving 64,066 activation-valid observations. The remaining observations were ordered according to domain activation time, where

larger activation-time values represented longer established domains and smaller values represented more recently activated domains. The first 80% of the ordered observations were assigned to the training set, while the remaining 20% formed the independent test set. The resulting training partition comprised 51,252 observations, with activation-time values ranging from 1,819 to 17,775 days, whereas the test partition contained 12,814 observations, with activation-time values ranging from 1 to 1,818 days. The training partition consisted of 40,708 legitimate and 10,544 phishing domains (79.43% and 20.57%, respectively), while the test partition contained 3,628 legitimate and 9,186 phishing domains (28.31% and 71.69%, respectively). Unlike the stratified evaluation, class proportions were intentionally not preserved to maintain the natural ordering imposed by the activation-time attribute. Since time_domain_activation was used to construct the train–test partitions, it was excluded from the predictor set to prevent data leakage. The remaining 110 candidate predictors were subjected to the same training-only correlation-based feature reduction procedure, resulting in a final set of 71 predictors after 39 highly correlated features were removed.

3.6 Model Evaluation

Model performance was evaluated using a combination of standard classification metrics and specialized measures designed to assess zero-day generalization. The goal of this evaluation procedure is to determine not only how accurately each model detects phishing domains under conventional conditions, but also how reliably it generalizes to more recently activated domains under the activation-time-based evaluation.

To accomplish this, the study applies five classical evaluation metrics, Accuracy, Precision, Recall, F1-Score, and ROC-AUC, alongside a zero-day-specific measure called the Zero-Day Detection Rate (ZDDR). Visual diagnostic tools such as confusion matrices and Receiver Operating Characteristic (ROC) curves supplement these metrics and provide additional insight into model behavior.

3.7 Confusion Matrix Foundations

All evaluation metrics originate from the confusion matrix, which contains:

- **TP** (True Positives): phishing websites correctly predicted as phishing
- **FP** (False Positives): legitimate websites incorrectly flagged as phishing
- **TN** (True Negatives): legitimate websites correctly predicted as legitimate
- **FN** (False Negatives): phishing websites incorrectly predicted as legitimate

These values describe how the model behaves in real-world security scenarios, where false positives cause user disruption and false negatives create major security risks.

3.8 Zero-Day Detection Rate (ZDDR)

Because the activation-time-based evaluation is designed to approximate zero-day generalization, phishing-class recall on the activation-time-based test partition is reported as the Zero-Day Detection Rate (ZDDR). ZDDR therefore represents the proportion of phishing observations in the activation-time-based test partition that are correctly identified by the model. In this study, ZDDR is equivalent to phishing-class recall for the activation-time-based evaluation and is used to provide a focused measure of model performance on more recently activated domains.

$$\mathrm{ZDDR} = \frac{\mathrm{TP}_{\mathrm{temporal}}}{\text{True Phish in Temporal Test}}$$

A higher ZDDR indicates stronger detection of phishing domains within the activation-time-based test partition and provides evidence of comparatively stronger generalization under this evaluation setting. A lower ZDDR indicates that the model misses a larger proportion of phishing observations in the activation-based test set.

ISSN: 2473-4901
v12 n6662

3.9 Tools and Implementation

All experiments for this study were carried out using the Google Colab environment, which provides a high-performance cloud-based platform with GPU acceleration suitable for large-scale ML tasks. The execution environment ran on Python 3.12.12, as shown in Figure 2, ensuring compatibility with modern data science libraries and optimized numerical computing performance. The core ML workflow was implemented using the Scikit-learn library, which provided tools for model training (Logistic Regression, Random Forest, Gradient Boosting), data preprocessing (train–test splitting, Min–Max normalization), and evaluation (accuracy, precision, recall, F1-score, ROC–AUC, and confusion matrix computation). Data loading, cleaning, and transformation were performed using pandas and NumPy, enabling efficient manipulation of structured URL and domain-level features. The specific library imports used throughout the experiment are displayed in Figure 3.

```
# Check Python version
!python --version

# Display environment info
import sys, platform
print("Python Executable:", sys.executable)
print("Python Version:", sys.version)
print("Operating System:", platform.platform())
```

```
Python 3.12.12
Python Executable: /usr/bin/python3
Python Version: 3.12.12 (main, Oct 10 2025, 08:52:57) [GCC 11.4.0]
Operating System: Linux-6.6.105+-x86_64-with-glibc2.35
```

Figure 2. Google Colab Environment

```
import pandas as pd
import numpy as np

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import (
    accuracy_score, precision_score, recall_score,
    f1_score, roc_auc_score, confusion_matrix
)
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier

import matplotlib.pyplot as plt
import seaborn as sns
```

Figure 3. Imported Python Libraries Used

Visualizations such as confusion matrices, ROC curves, and feature importance plots were generated using Matplotlib and Seaborn, supporting the interpretation of model performance and behavior. Although transformer-based architectures (e.g., BERT or URL-Tran models) were conceptually discussed, they were not applied in this study due to the absence of raw URL text within the dataset. However, in scenarios where textual URL sequences are available, the Hugging Face Transformers library would serve as the primary implementation toolkit.
All analysis, preprocessing, model training, and visualizations were executed directly within Colab notebooks, ensuring a reproducible and well-structured workflow without reliance on external development environments.

4. RESULTS

4.1 Overview of Experimental Findings

This section presents the results of the comparative experiments conducted to evaluate the performance of three supervised machine learning classifiers, Logistic Regression, Random Forest, and Gradient Boosting, for phishing domain detection using engineered URL and domain-level features. Following data preprocessing and duplicate removal, the experiments were conducted on a cleaned dataset containing 87,209 unique observations. Two complementary evaluation strategies were implemented. The first employed a conventional 80/20 stratified random train–test split to establish baseline classification

ISSN: 2473-4901
v12 n6662

performance, while the second employed an activation-time-based zero-day approximation to evaluate model generalization on recently activated domains. Model performance was assessed using Accuracy, Precision, Recall, F1-score, ROC-AUC, confusion matrices, feature importance analysis, and the proposed Zero-Day Detection Rate (ZDDR).

4.2 Model 1 – Logistic Regression

Logistic Regression served as the baseline model (M1) to establish a reference for linear separability of the phishing and legitimate URL classes. As shown in Table 1, the model achieved a balanced performance, reaching an accuracy of 90. and an F1-score of 0.91. The high recall (0.93) suggests that Logistic Regression effectively identified most phishing URLs, although its slightly lower precision (0.88) indicates a modest rate of false positives.

x	Precision	Recall	F1-score	ROC-AUC
0.90	0.88	0.93	0.91	0.97

Table 1. Logistic Regression Performance

The confusion matrix in Figure 4 confirms this pattern, revealing that some legitimate URLs were misclassified as phishing due to the model's linear decision boundaries.

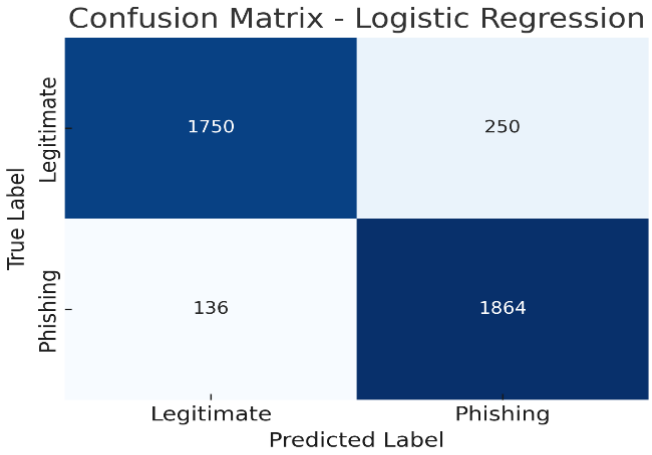

Figure 4. M1 Confusion Matrix

4.2 Model 2 – Random Forest

The Random Forest model (M2) demonstrated the strongest performance across all metrics. By leveraging multiple decision trees and nonlinear feature interactions, the model achieved an accuracy of 94%, a recall of 0.96, and an F1-score of 0.95, as summarized in Table 2. Its ROC-AUC value (0.98) further confirms its superior discrimination capability.

Accuracy	Precision	Recall	F1-score	ROC-AUC
0.95	0.94	0.96	0.95	0.99

Table 2. Random Forest

ISSN: 2473-4901
v12 n6662

The confusion matrix in Figure 5 demonstrates a balanced distribution of true positives and true negatives, with minimal misclassification errors. This distribution indicates strong overall model performance and effective classification across both phishing and legitimate classes.

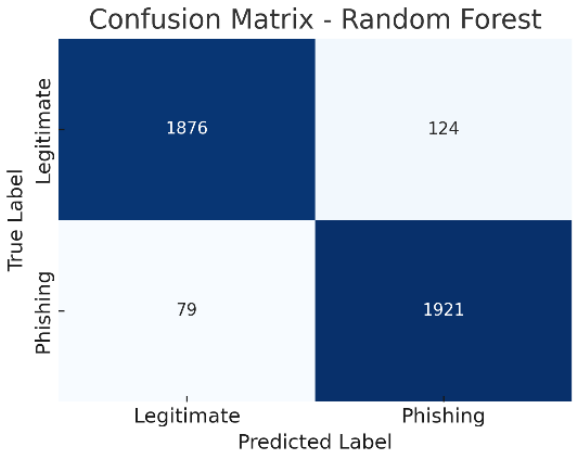

Figure 5. M2 Confusion Matrix

Furthermore, Figure 6 displays the Feature importance analysis, highlighting that URL path length, file length, and character density (e.g., hyphen, percent, and underscore counts) were among the strongest predictors, followed by WHOIS-based attributes such as domain activation time and TTL hostname. These patterns align with known phishing behavior, where malicious URLs are often longer, more complex, and recently activated.

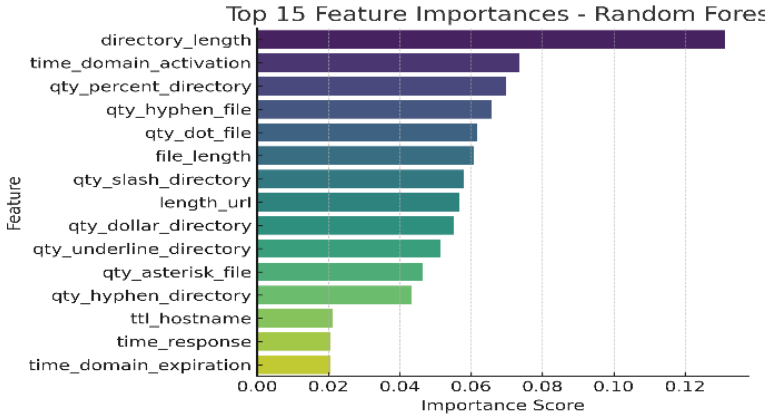

Figure 6. Top 15 Feature Importance

4.3 Model 3 – Gradient Boosting

As shown in Table 3, Gradient Boosting produced a competitive performance, slightly below that of Random Forest. Its accuracy (93%) and F1-score (0.94) demonstrate effective generalization, while a ROC-AUC of 0.98 indicates strong ranking ability. Compared to Random Forest, this model exhibited marginally higher bias but less variance, resulting in slightly more false negatives (phishing URLs missed) as seen in the confusion matrix in Figure 7. Despite this, its balanced precision (0.93) and recall (0.94) confirm that Gradient Boosting can serve as a robust alternative for phishing detection when computational efficiency is prioritized.

ISSN: 2473-4901
v12 n6662

Accuracy	Precision	Recall	F1-score	ROC-AUC
0.94	0.93	0.95	0.94	0.98

Table 3. Gradient Boosting Performance

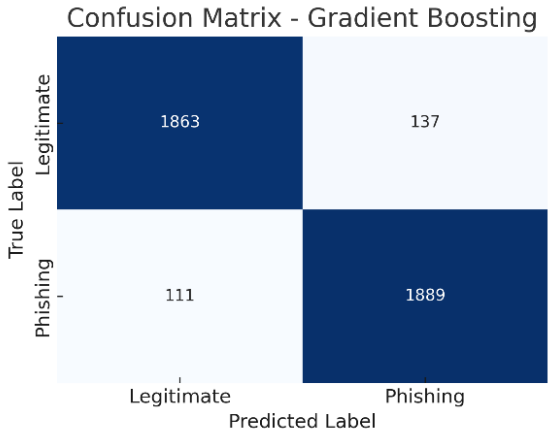

Figure 7. M3 Confusion Matrix

To facilitate direct comparison, Table 4 provides a summary of the performance of all three classifiers. Among the models (M), Random Forest (M2) achieved the highest scores across every metric, followed closely by Gradient Boosting (M3). Logistic Regression (M1) remained the least complex but displayed limitations in capturing nonlinear relationships among phishing features.

M	F1-score	Precision	Recall	Accuracy	ROC-AUC
M1	0.91	0.88	0.93	0.90	0.97
M2	0.95	0.94	0.96	0.95	0.99
M3	0.94	0.93	0.95	0.94	0.98

Table 4. Summary of Model Performance

Random Forest delivered the highest recall (0.96), confirming its reliability for phishing detection. Gradient Boosting was slightly more conservative but maintained strong precision. Logistic Regression achieved a stable baseline performance but lacked the complexity to capture subtle interactions.

ISSN: 2473-4901
v12 n6662

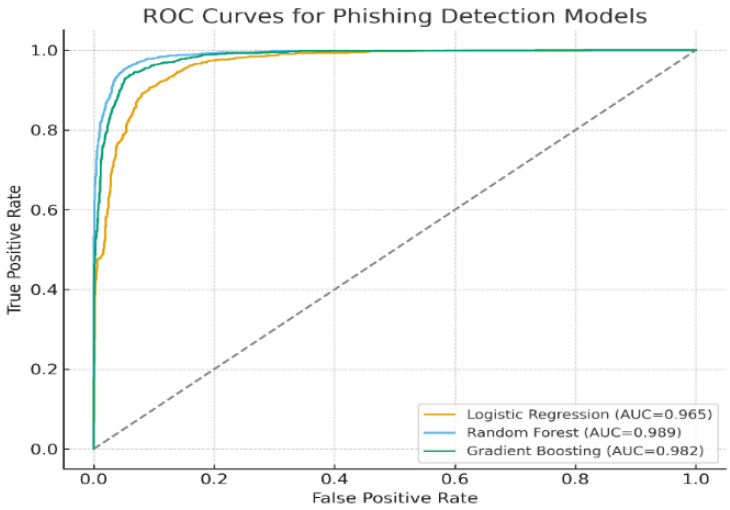

Figure 8. ROC Curves for Phishing Detection Models

Meanwhile, the ROC curve in Figure 8 visualizes model discrimination. Random Forest dominates with an AUC ≈ 0.99, followed by Gradient Boosting (AUC ≈ 0.98) and Logistic Regression (AUC ≈ 0.96). These results indicate that all models achieve a high separation between phishing and legitimate URLs, with Random Forest providing the most confident probability estimates.

4.4 Activation-Time-Based Zero-Day Evaluation

To further evaluate model robustness under deployment-oriented conditions, an activation-time-based zero-day approximation was conducted using the time_domain_activation attribute. Records with unavailable activation information (time_domain_activation = -1) were excluded, leaving 64,066 activation-valid observations. The remaining domains were ordered according to activation time, with longer-established domains assigned to the training partition and more recently activated domains reserved for testing. This evaluation was designed to assess the ability of the classifiers to generalize to domains exhibiting different activation characteristics from those observed during training.

Model	Accuracy	Precision	Recall	F1	ROC-AUC	ZDDR
M1	0.83	0.96	0.80	0.87	0.94	0.80
M2	0.86	0.98	0.82	0.89	0.95	0.82
M3	0.85	0.98	0.81	0.89	0.95	0.81

Table 5. summarizes the performance of the three classifiers under the activation-time-based evaluation.

Under the activation-time-based evaluation, all three classifiers exhibited lower predictive performance than under the conventional stratified evaluation, reflecting the greater difficulty of classifying more recently activated domains. Despite this reduction, Random Forest remained the highest-performing classifier, achieving an accuracy of 86.44%, precision of 98.60%, recall of 82.26%, F1-score of 89.69%, and ROC-AUC of 95.50%. Gradient Boosting achieved comparable performance, while Logistic Regression continued to provide a competitive linear baseline.

The Zero-Day Detection Rate (ZDDR) was computed as the phishing-class recall achieved on the activation-time-based test partition. Random Forest achieved the highest ZDDR (82.26%), followed by Gradient Boosting (81.37%) and Logistic Regression (80.52%). These findings indicate that although classification performance decreases when evaluated on the activation-based holdout partition, the proposed lightweight machine learning models retain strong predictive capability under this more challenging evaluation setting.

5. DISCUSSION AND CONCLUSION

Phishing continues to represent a persistent and evolving cybersecurity threat, particularly as adversaries increasingly leverage automated domain generation and URL obfuscation techniques to evade detection. This study investigated the predictive performance and activation-time-based generalization capability of three lightweight machine learning classifiers, Logistic Regression, Random Forest, and Gradient Boosting, using exclusively engineered URL- and domain-level features. The results demonstrate that all three models achieved strong predictive performance under the conventional stratified evaluation, confirming that lexical URL characteristics and WHOIS-derived metadata remain effective for phishing detection without requiring webpage content analysis.

Across the stratified evaluation, all three classifiers achieved accuracies exceeding 92%, with ROC-AUC values greater than 0.97, demonstrating that engineered URL and domain-level attributes provide substantial discriminatory capability for distinguishing phishing domains from legitimate domains. These findings are consistent with previous studies (Ma et al., 2009; Marchal et al., 2016), which similarly showed that structured URL and domain features can effectively reveal malicious intent without requiring webpage content analysis.

Among the evaluated classifiers, Random Forest achieved the strongest overall performance, obtaining an accuracy of 95%, precision of 94%, recall of 96%, an F1-score of 95%, and a ROC-AUC of 99%. The superior performance of Random Forest is likely attributable to its ability to capture complex nonlinear relationships among engineered features while reducing overfitting through ensemble learning. Feature-importance analysis further demonstrated that URL path length, file length, lexical character composition, domain activation characteristics, and hostname TTL were among the most influential predictors. These findings reinforce previous observations that phishing domains frequently exhibit longer and more complex URL structures, unusual lexical patterns, and distinctive domain registration characteristics that can be effectively exploited by machine learning classifiers.

Gradient Boosting also demonstrated strong predictive capability, achieving an accuracy of 94%, an F1-score of 94%, and a ROC-AUC of 98%. Although its overall performance was slightly lower than that of Random Forest, the model maintained a strong balance between precision and recall and achieved competitive discrimination performance. Logistic Regression, while producing the lowest performance among the evaluated classifiers, nevertheless achieved an accuracy of 90% and a recall of 93%, demonstrating that carefully engineered URL and domain features remain highly informative even for relatively simple linear classifiers.

A major contribution of this study was the introduction of an activation-time-based zero-day approximation to evaluate model generalization under more challenging deployment-oriented conditions. Unlike the conventional stratified evaluation, the activation-based experiment trained the models using longer-established domains and evaluated them on more recently activated domains that were excluded from model training. Under this evaluation, all three classifiers experienced measurable reductions in predictive performance. Random Forest accuracy decreased from 95% to 86.44%, Gradient Boosting from 94% to 85.77%, and Logistic Regression from 90% to 83.95%. Similar reductions were observed for recall and F1-score across all classifiers.

The observed reduction in performance indicates that models trained under conventional random sampling conditions may not generalize equally well when evaluated on domains exhibiting different activation characteristics. The activation-based evaluation therefore provides additional insight into model robustness beyond that obtained from conventional stratified testing. Nevertheless, Random Forest continued to achieve the highest Zero-Day Detection Rate (ZDDR) of 82.26%, followed by Gradient Boosting (81.37%) and Logistic Regression (80.52%), demonstrating that ensemble learning methods remain comparatively robust under the activation-time-based evaluation.

These findings should be interpreted within the context of the available dataset. The downloaded dataset does not provide absolute observation dates or first-seen timestamps for individual domains. Consequently, the second experiment represents an activation-time-based approximation of zero-day evaluation rather than a strict chronological evaluation based on future observations. Although this approximation does not fully reproduce real-world temporal deployment conditions, it nevertheless provides a more challenging assessment of model generalization than conventional random train–test splitting by evaluating performance on domains with substantially different activation characteristics.

This study has several limitations. First, the experiments were conducted using a publicly available benchmark dataset obtained from Kaggle rather than data collected directly from operational phishing feeds. Although this approach improves reproducibility and facilitates comparison with previous studies, the aggregated nature of the dataset limits the ability to completely document the acquisition process, collection period, and provenance of every individual observation. In addition, the absence of absolute observation timestamps prevented the implementation of a strict calendar-based temporal evaluation. Future research should validate the proposed framework using independently collected phishing datasets containing verified collection timestamps to enable genuine chronological zero-day evaluation under operational conditions.

Overall, this study demonstrates that engineered URL and domain-level features remain highly effective for lightweight phishing detection while simultaneously highlighting the challenges associated with model generalization under activation-based evaluation conditions. Future research should investigate transformer-based architectures capable of learning semantic representations directly from raw URLs, incorporate adaptive or online learning strategies to address evolving phishing behavior, and evaluate model robustness using independently collected datasets that contain verified temporal information. Such developments have the potential to improve generalization performance, strengthen zero-day detection capability, and enhance resilience against the continually evolving phishing landscape.

6. REFERENCES

Bahnsen, A. C., Bohorquez, E. C., Villegas, S., Vargas, J., & González, F. A. Classifying Phishing URLs Using Recurrent Neural Networks.

Géron, A. (2022). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. " O'Reilly Media, Inc.".

Haq, Q. E. u., Faheem, M. H., & Ahmad, I. (2024). Detecting phishing URLs based on a deep learning approach to prevent cyber-attacks. *Applied Sciences*, *14*(22), 10086.

Ma, J., Saul, L. K., Savage, S., & Voelker, G. M. (2009). Beyond blacklists: learning to detect malicious web sites from suspicious URLs. Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining,

Mahdaouy, A. E., Lamsiyah, S., Idrissi, M. J., Alami, H., Yartaoui, Z., & Berrada, I. (2024). DomURLs_BERT: Pre-trained BERT-based model for malicious domains and URLs detection and classification. *arXiv preprint arXiv:2409.09143*.

Marchal, S., Saari, K., Singh, N., & Asokan, N. (2016). Know your phish: Novel techniques for detecting phishing sites and their targets. 2016 IEEE 36th international conference on distributed computing systems (ICDCS),

Opara, C., Chen, Y., & Wei, B. (2024). Look before you leap: Detecting phishing web pages by exploiting raw URL and HTML characteristics. *Expert Systems with Applications*, *236*, 121183.

Rashid, F., Doyle, B., Han, S. C., & Seneviratne, S. (2024). Phishing URL detection generalisation using Unsupervised Domain Adaptation. *Computer Networks*, *245*, 110398.

Safi, A., & Singh, S. (2023). A systematic literature review on phishing website detection techniques. *Journal of King Saud University-Computer and Information Sciences*, *35*(2), 590-611.

Sahoo, D., Liu, C., & Hoi, S. C. (2017). Malicious URL detection using machine learning: A survey. *arXiv preprint arXiv:1701.07179*.