Draft version December 3, 2025
Typeset using LaTeX **twocolumn** style in AASTeX7.0.1

Comparative Analysis of Feature Engineering Strategies for Phishing URL Detection using the PhiUSIIL Dataset

Sai Charan Petchetti[1]

[1] *University of California, Los Angeles*

ABSTRACT

Phishing attacks continue to constitute a major cybersecurity threat, with detection systems increasingly relying on machine learning techniques to identify fraudulent URLs. This study presents a comparative analysis of feature engineering strategies for phishing URL detection using the PhiUSIIL dataset, which contains 235,795 URLs with three distinct feature categories: URL-based, HTML-based, and derived features. We evaluate the discriminative power of each feature category using classical machine learning models including Logistic Regression, Random Forest, and XGBoost. Our results demonstrate that models trained on URL and HTML features achieve near-perfect classification performance (99.98% accuracy), substantially outperforming URL-only configurations (99.80% accuracy). Feature importance analysis reveals that HTML features such as *NoOfExternalRef* and *NoOfImage* provide the strongest discriminative signal, while derived features based on third-party legitimate URL databases contribute minimal additional value when HTML features are present. Additionally, TF-IDF vectorization of raw URL strings achieves competitive performance (99.70% accuracy) without manual feature engineering, though generalizability concerns remain. These results offer practical guidance for balancing detection accuracy against computational cost in phishing URL detection systems, while highlighting the critical limitation of potential model overfitting to specific datasets.

1. INTRODUCTION

Phishing attacks are deceptive attempts to steal sensitive information, such as passwords, credit card numbers, or personal data, by impersonating a trustworthy entity. One method by which phishing attacks are carried out is through fraudulent websites designed to mimic legitimate websites, thereby deceiving victims into entering their login credentials, financial information, or other sensitive data.

Phishing remains the dominant vector for cybercrime, with volume increasing annually. According to the Anti-Phishing Working Group (APWG), 2023 set the historical record for cyber threats, with nearly five million phishing attacks detected (Anti-Phishing Working Group 2024). This surge in volume highlights the inadequacy of existing detection methods.

Historically, phishing detection relied on static methods such as blacklists, which struggled to adapt to the rapid creation of new malicious domains (S. Kavya & D. Sumathi 2025). To address these limitations, machine learning techniques emerged as a critical defense mechanism, offering the ability to automate detection at scale and identify zero-day threats. I. Fette et al. (2007) pioneered this approach by training a Random Forest model to classify e-mails as either legitimate or phishing, achieving an accuracy of 96%. Following this foundational work, the field evolved rapidly. Researchers like J. Ma et al. (2009) demonstrated the effectiveness of learning from URL lexical features at scale, while later studies by O. K. Sahingoz et al. (2019) and R. S. Rao & A. R. Pais (2019) began incorporating more complex feature sets, including deep learning architectures, to capture more subtle obfuscation techniques.

Today, researchers continue to investigate diverse feature engineering strategies to improve detection performance. Unlike many machine learning domains where baseline models achieve modest performance, phishing detection models typically achieve high accuracy rates ($> 95\%$) from the outset due to the pronounced data patterns of malicious URLs. In this context, even marginal improvements of 0.1 to 0.5% in accuracy translate to thousands of additional threats detected or false alarms prevented at scale, making feature importance analysis particularly valuable.

The effectiveness of machine learning-based phishing detection systems depends on the features used to represent URLs. Three main categories of features have emerged in the literature: URL-based features extracted directly from the URL string (such as length, special

Email: saicharan@ucla.edu

character counts, and protocol indicators), HTML-based features derived from webpage content (including structural elements, metadata, and form characteristics), and derived features computed using external reference datasets of legitimate websites. However, the relative discriminative power of these feature categories remains unclear, and researchers face practical tradeoffs between detection accuracy and computational cost when deciding which features to implement.

A. Prasad & S. Chandra (2024a) compiled a comprehensive dataset of 134,850 legitimate and 100,945 phishing websites, along with all three categories of features. The authors developed an incremental ensemble learning framework that achieved 99.79% accuracy on this dataset.

This study conducts a systematic evaluation of feature engineering strategies using the PhiUSIIL dataset to address the following research questions:

1. What is the discriminative power of each feature category when used independently or in combination?
2. Which specific features contribute most to classification performance within each category?
3. Can automated feature extraction approaches such as TF-IDF vectorization match the performance of manually engineered features?
4. Do derived features based on external reference datasets provide incremental value beyond URL and HTML features?

By answering these questions, we provide practical guidance for designing phishing detection systems that balance accuracy, computational efficiency, and generalizability.

2. PHIUSIIL PHISHING URL DATASET FEATURES

The PhiUSIIL dataset (A. Prasad & S. Chandra 2024b) consists of $N = 235{,}795$ URLs, of which $N_{\text{leg}} = 134{,}850$ are legitimate URLs (approximately 57%) and $N_{\text{phish}} = 100{,}945$ are phishing URLs (approximately 43%). The dataset also contains three categories of features extracted from these URLs: URL features, HTML features, and derived features. The authors collected legitimate URLs from the OpenPage Rank Initiative (DomCop n.d.) and phishing URLs from Phish-Tank (Cisco Talos n.d.), OpenPhish (OpenPhish n.d.), and MalwareWorld (C. Polop n.d.). Derived features utilize datasets from the OpenPage Rank Initiative (DomCop n.d.), which contain URLs identified as legitimate, to calculate probability scores and similarity metrics. Some of the key features belonging to each category are listed below.

2.1. *Feature Categories*

2.1.1. *URL Features*

- **TLD**: Top-level domain (e.g., .com, .edu).
- **URLLength**: Total character count of the URL.
- **IsDomainIP**: Binary indicator of whether an IP address is used instead of a domain name.
- **NoOfSubDomain**: Count of subdomain levels.
- **NoOfObfuscatedChar**: Count of obfuscated characters.
- **IsHTTPS**: Binary indicator for the presence of the HTTPS protocol.
- **Symbol Counts**: Counts of digits, equal signs (=), question marks (?), and ampersands (&).

2.1.2. *HTML Features*

- **Structure and Metadata**: *LargestLineLength* (maximum line length in HTML), *HasTitle*, *HasFavicon*, *IsResponsive*, *HasDescription* (meta description), *HasCopyrightInfo*, and *HasSocialNet* (presence of social network links).
- **Elements**: Counts of *NoOfPopup*, *NoOfiFrame*, *NoOfImage*, and *NoOfJS* (JavaScript elements).
- **Forms and Interaction**: *HasExternalFormSubmit*, *HasPasswordField*, *HasSubmitButton*, and *HasHiddenFields*.
- **Financial Indicators**: Binary indicators for keywords such as *Bank*, *Pay*, and *Crypto*.
- **References**: Counts of hyperlink types, including *NoOfSelfRef*, *NoOfEmptyRef*, and *NoOfExternalRef*.

2.1.3. *Derived Features*

- **CharContinuationRate (CCR)**: Measures the continuity of character sequences, defined as:

$$\text{CCR} = \frac{\text{Length of longest sequences}}{\text{Total URL length}} \tag{1}$$

- **URLTitleMatchScore**: Similarity score (ranging from 0 to 100) indicating how well the URL matches the page title.

- **URLCharProb**: Probability score based on the occurrence of characters in legitimate URLs, calculated as:

$$\text{URLCharProb} = \frac{\sum_{i=0}^{n} \text{prob}(\text{URL}[\text{char}_i])}{n} \quad (2)$$

where n is the count of alphanumeric characters (a–z, 0–9).

- **TLDLegitimateProb**: Probability that the URL's TLD appears in the top 10 million legitimate websites.

- **URL Similarity Index (USI)**[2]: Similarity score quantifying how well the URL matches known legitimate URLs, calculated as:

$$\text{USI} = \sum_{i=0}^{n} \left(\frac{50}{n} + \frac{100 \times (n-i)}{n \times (n+1)} \right) \quad (3)$$

where n is the length of the longest URL being compared.

3. METHODOLOGY

3.1. *Data Pre-processing*

The original dataset labels legitimate websites as 1 and phishing websites as 0. For consistency with standard classification conventions where the positive class represents the target of detection, we inverted the labels such that phishing websites are labeled as 1 (positive class) and legitimate websites as 0 (negative class).

We defined four feature configurations for our analysis. For all configurations except the *URL String Only* configuration, non-numeric columns (e.g., the raw URL string, Domain, TLD, and Title) were removed by default to ensure compatibility with numerical classifiers:

- **URL Features Only**: Includes URL features but excludes HTML features and derived features. This configuration tests baseline performance achievable using only characteristics extracted from the URL string.

- **URL + HTML Features**: Includes URL features and HTML features, but excludes derived features. This configuration evaluates the added value of webpage content analysis.

- **URL String Only**: Includes only the raw URL strings. This is the only configuration that utilizes non-numeric data. It excludes all other URL features, HTML features, and derived features. This configuration uses TF-IDF vectorization to extract character-level patterns from the URL text without manual feature engineering.

- **All Features**: Includes URL features, HTML features, and derived features. *URLSimilarityIndex* is excluded from this configuration because preliminary experiments showed that models relied excessively on this single feature, producing artificially inflated accuracy scores that may not generalize well. The remaining derived features (*CharContinuationRate*, *URLTitleMatchScore*, *URLCharProb*, and *TLDLegitimateProb*) are retained.

For all experiments, we used an 80-20 train-test split with a fixed random state of 65 to ensure reproducibility.

3.2. *Classification Models*

All models were implemented using the scikit-learn machine learning library (F. Pedregosa et al. 2011), with the exception of XGBoost which uses the XGBoost library (T. Chen & C. Guestrin 2016).

- **Logistic Regression (LR)** is a statistical method for binary classification that models the probability of a discrete outcome using a logistic (sigmoid) function. In our implementation, *StandardScaler* is used for feature normalization and maximum iterations are set to 2000 for convergence. Scikit-learn's `LogisticRegression` uses L_2 regularization by default, which helps prevent overfitting by penalizing large coefficient values.

- **Random Forest (RF)** is an ensemble learning method that constructs multiple decision trees during training and outputs the class selected by the majority of trees. Our implementation uses 100 decision trees with bootstrap aggregation (bagging), where each tree is trained on a random subset of the data to reduce variance and improve generalization.

- **XGBoost** is an optimized gradient boosting algorithm that builds an ensemble of decision trees sequentially, with each new tree trained to correct the residual errors of the previous ensemble. Our implementation uses 100 estimators with a maximum tree depth of 6, balancing model complexity and performance.

We restricted the evaluation of XGBoost to the URL feature configuration. Preliminary experiments indi-

[2] Refer to A. Prasad & S. Chandra (2024a) for detailed algorithm.

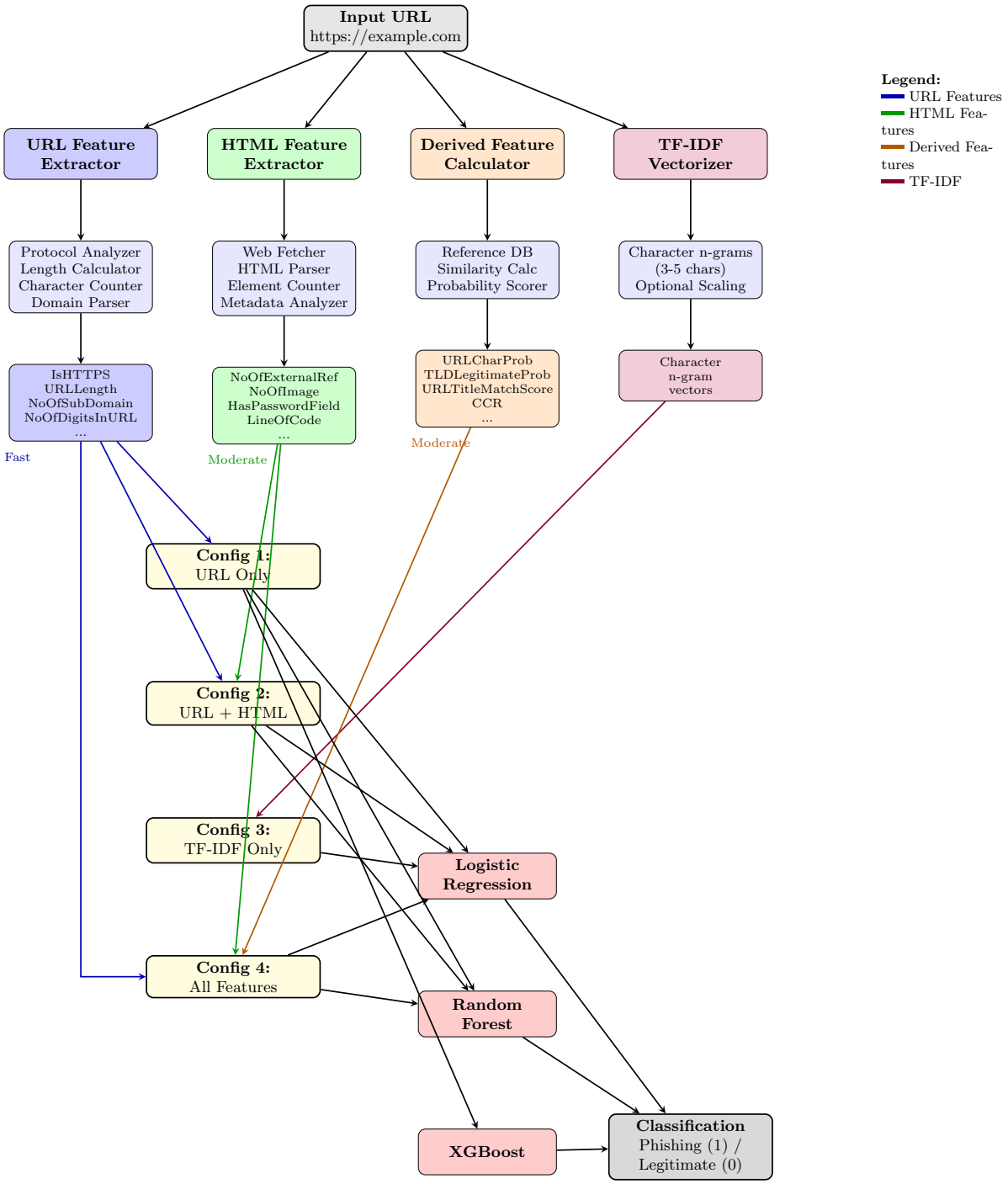


Figure 1. System architecture for phishing URL detection showing four feature extraction pathways. The diagram illustrates three manual feature engineering approaches (URL, HTML, and derived features) and one automated approach (TF-IDF vectorization). The configuration layer shows how features are combined for different experimental setups, feeding into three classical machine learning models.

cated that Random Forest and XGBoost achieved comparable performance on this dataset. Furthermore, given the inherent noise in phishing data, the gradient boosting approach may be more prone to overfitting (D. Opitz & R. Maclin 1999).

3.3. *TF-IDF URL Analysis*

Term Frequency-Inverse Document Frequency (TF-IDF) is a statistical weight used to evaluate the importance of a word to a document in a collection of docu-

Table 1. Summary of Models and Feature Configurations

Model	Feature Configuration
Logistic Regression	URL Features Only
Random Forest (n=100)	URL Features Only
XGBoost (n=100, depth=6)	URL Features Only
Logistic Regression	URL + HTML Features
Random Forest (n=100)	URL + HTML Features
Logistic Regression	URL String (No Scaling)
Logistic Regression	URL String (With Scaling)
Logistic Regression	All Features
Random Forest (n=100)	All Features

ments (G. Salton & C. Buckley 1988). It is the product of two metrics:

- **Term Frequency (TF)** measures the local prevalence of a term within a document. In our implementation, this corresponds to the frequency of a specific character n-gram within a single URL string.
- **Inverse Document Frequency (IDF)** is a global weighting factor that assigns higher value to terms appearing in fewer documents, while penalizing high-frequency terms (e.g., "http", "com", "www").

$$W_{i,j} = \mathrm{TF}_{i,j} \times \log\left(\frac{N}{DF_i}\right)$$

where,

  - $W_{i,j}$: Weight of word i in document j
  - N: Total number of documents
  - DF_i: Number of documents containing word i

Our implementation treats each URL as a document and extracts character n-grams of length 3 to 5, with a minimum document frequency threshold of 0.01%. This threshold requires that a character n-gram appear in at least 0.01% of all URLs to be included as a feature, which helps filter out infrequent n-grams that may capture noise.

4. RESULTS

To evaluate the effectiveness of the classification models, we use four standard performance metrics. These are:

- **Accuracy**: Proportion of correctly classified URLs (both legitimate and phishing) relative to the total number of samples.

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

- **Precision**: Measures the reliability of the model when it predicts a URL is phishing.

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

- **Recall**: Measures the model's ability to correctly identify all phishing URLs in the dataset.

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

- **F1-Score**: Harmonic mean of Precision and Recall.

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

In the equations above, TP, TN, FP, and FN denote True Positives, True Negatives, False Positives, and False Negatives, respectively.

Note that we do not present ROC-AUC curves, since all models achieved near-perfect class separation (F1-Scores > 0.99), with the Area Under the Curve (AUC) asymptotically approaching 1.0.

4.1. *Feature Importance Bar Chart for RF Models*

In addition to the above metrics, we computed the Gini Importance for each feature to interpret Random Forest models. This metric measures how much each feature contributes to reducing classification uncertainty across all trees in the ensemble. Features with higher importance scores are more effective at separating phishing URLs from legitimate URLs. We visualized the top 15 features using a horizontal bar chart.

4.2. *URL Features Only*

Table 2. Performance Metrics: URL Features Only

Model	Acc.	Prec.	Rec.	F1
Logistic Reg.	0.9963	0.9993	0.9921	0.9957
Random Forest	0.9976	0.9993	0.9951	0.9972
XGBoost	0.9980	0.9998	0.9955	0.9976

The results indicate that the structural and lexical characteristics of the URL string alone provide a strong

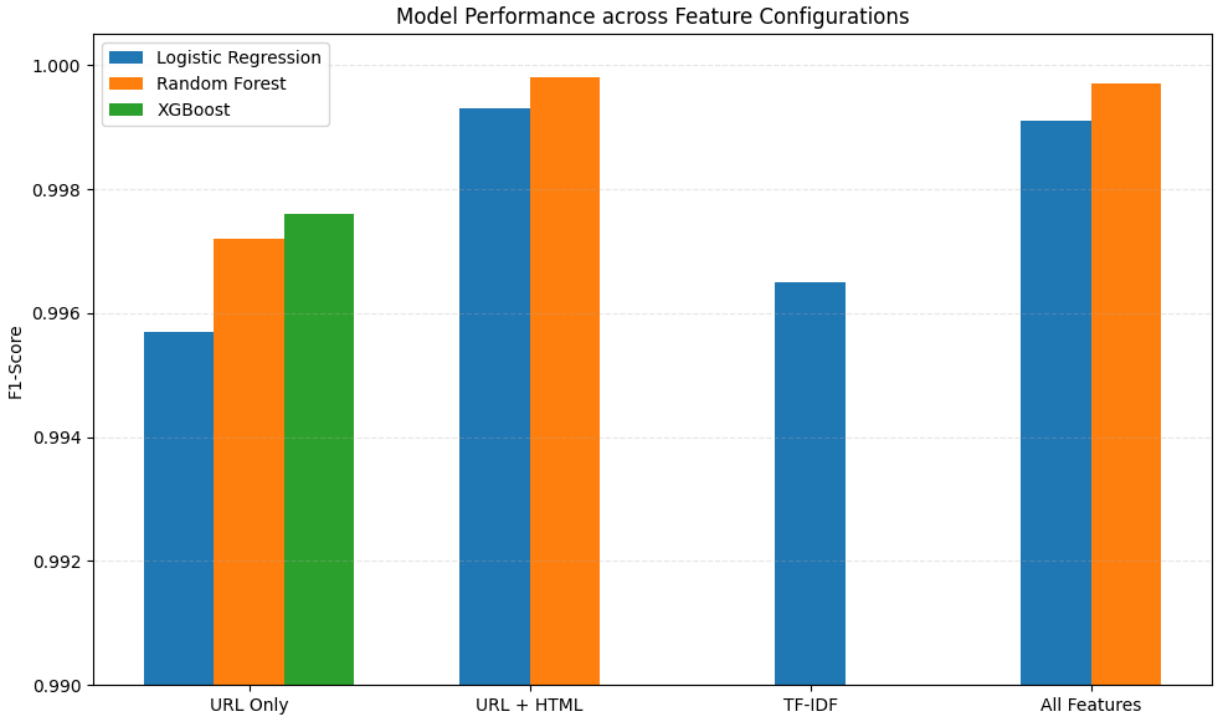


Figure 2. Model Performance across Feature Configurations. Note the y-axis range [0.99, 1.00].

signal for detecting phishing URLs. All three models achieve accuracy exceeding 99.6%, with XGBoost slightly outperforming the others at 99.80%. These findings suggest that computationally expensive HTML extraction may not be necessary to achieve high baseline performance when computational resources are limited.

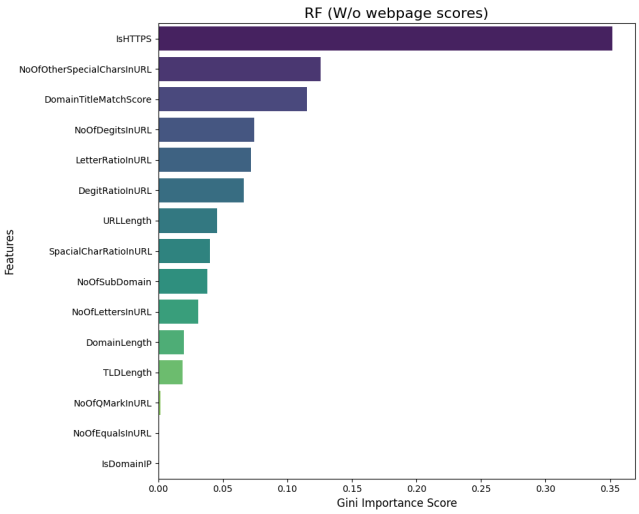


Figure 3. Top 15 features ranked by Gini Importance from the Random Forest model (URL configuration)

As shown in Figure 3, *IsHTTPS* serves as the most important feature, contributing significantly more to the model's decision-making than any other feature. This is followed by *NoOfOtherSpecialCharsInURL* and *NoOfDigitsInURL*, confirming that phishing attempts rely on insecure protocols and anomalous character patterns.

4.3. *URL + HTML Features*

Table 3. Performance Metrics: URL + HTML Features

Model	**Acc.**	**Prec.**	**Rec.**	**F1**
Logistic Reg.	0.9994	0.9996	0.9990	0.9993
Random Forest	0.9998	1.0000	0.9995	0.9998

Incorporating HTML features yields a measurable improvement in performance compared to the URL-only configuration. The Random Forest model achieves near-perfect classification with an accuracy of 99.98% and a precision of 1.0000. These results suggest that HTML attributes, such as the presence of hidden fields or external form submissions, provide additional discriminative power when the URL structure effectively mimics legitimate URLs. However, this increase in predictive performance comes with the added latency cost of fetching and parsing web content.

As shown in Figure 4, HTML features such as *NoOfExternalRef*, *NoOfImage*, and *LineOfCode* rank

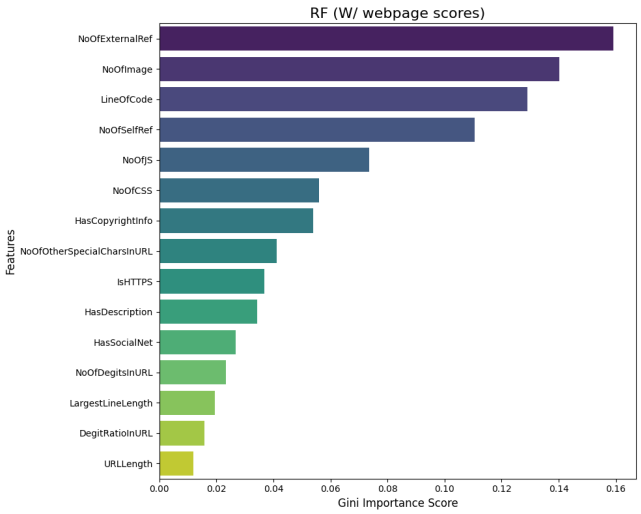


Figure 4. Top 15 features ranked by Gini Importance from the Random Forest model (URL + HTML configuration)

highest in importance, significantly outperforming URL features such as *IsHTTPS* and *URLLength*. This indicates that when webpage content is available, Random Forest models use HTML features to provide a stronger signal for detecting phishing URLs.

4.4. *URL String Only*

Table 4. Performance Metrics: URL String Only (TF-IDF)

Model Config.	Acc.	Prec.	Rec.	F1
LR (No Scaling)	0.9962	1.0000	0.9912	0.9956
LR (With Scaling)	0.9970	0.9994	0.9935	0.9965

The TF-IDF vectorization approach demonstrates that raw character n-grams can match, and in some cases surpass, the performance of manually engineered features. The scaled Logistic Regression model achieves an accuracy of 99.70%, slightly outperforming the Logistic Regression model trained on engineered URL features (99.63%). This result suggests that the model successfully learned discriminative patterns, such as common phishing keywords and suspicious character sequences, without requiring manual feature engineering.

To validate the impact of feature scaling on the TF-IDF vectorization approach, we performed 5-Fold Cross-Validation. The results demonstrate a consistent performance improvement when applying standard scaling (with mean centering disabled to preserve sparsity). As shown in Table 5, the scaled model achieved an average accuracy of 99.69%, consistently outperforming the unscaled model (99.59%) across every fold.

Table 5. 5-Fold Cross-Validation Results: Unscaled vs. Scaled TF-IDF

Fold	Unscaled	Scaled	Impr.
Fold 1	0.9958	0.9968	+0.0010
Fold 2	0.9960	0.9971	+0.0011
Fold 3	0.9957	0.9967	+0.0010
Fold 4	0.9959	0.9969	+0.0010
Fold 5	0.9961	0.9970	+0.0009
Average	**0.9959**	**0.9969**	**+0.10%**

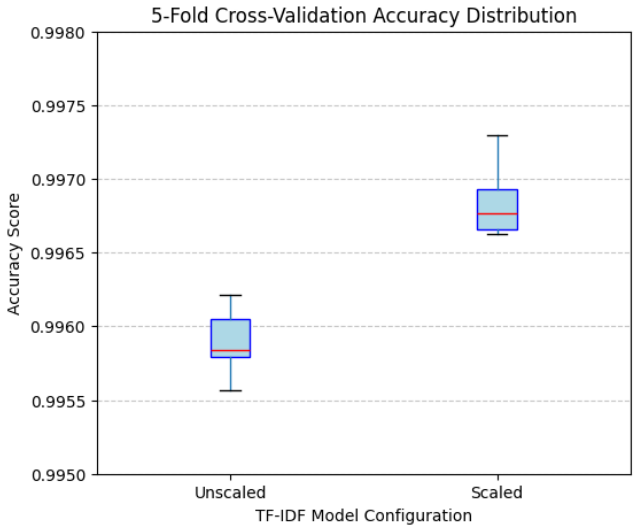


Figure 5. Box plot of 5-Fold Cross-Validation accuracy distributions: Unscaled vs. Scaled TF-IDF. Note the y-axis range [0.995, 0.998].

Figure 5 visually confirms the higher accuracy of the scaled model. Notably, the lower bound of the scaled model's performance exceeds the upper bound of the unscaled model. This improvement can be attributed to the standardization of feature variance by scaling, which prevents the L_2 regularization penalty from disproportionately suppressing informative but high-variance n-grams.

4.5. *All Features (excluding USI)*

Table 6. Performance Metrics: All Features (excluding USI)

Model	Acc.	Prec.	Rec.	F1
Logistic Reg.	0.9992	0.9994	0.9988	0.9991
Random Forest	0.9997	1.0000	0.9994	0.9997

Comparison with the URL + HTML configuration reveals that the inclusion of derived features does not result in a net performance gain. The accuracy for the Random Forest model decreases marginally from 99.98% to 99.97%, while the accuracy of the Logistic Regression model decreases from 99.94% to 99.92%. This saturation effect suggests that the URL and HTML feature sets already capture the most discriminative information in the dataset, and additional derived features provide limited incremental value.

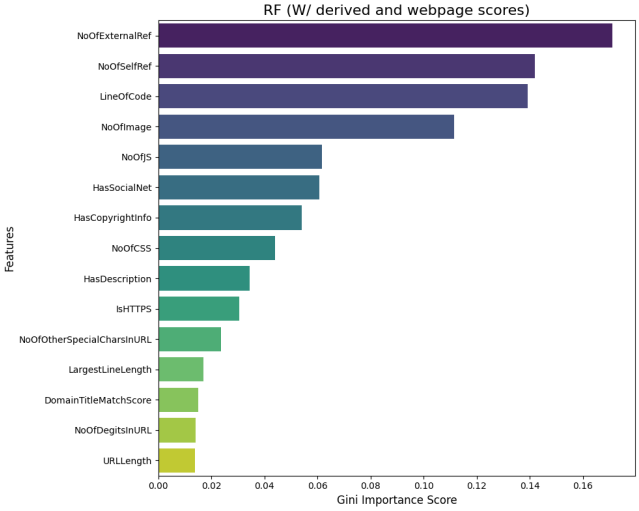


Figure 6. Top 15 features ranked by Gini Importance from the Random Forest model (All features configuration)

Figure 6 supports this observation, showing that derived features do not rank amongst the most important features in the Random Forest model. The only derived feature in the top 15 is *DomainTitleMatchScore*, which ranks near the bottom, confirming that the model prioritizes HTML features over derived features. Furthermore, although the derived features utilize the same external dataset of legitimate URLs as our training and test sets, the decrease in accuracy effectively rules out data leakage.

5. DISCUSSION

5.1. *Feature Importance Analysis*

5.1.1. *URL Features Only*

The *IsHTTPS* feature emerged as the single most discriminative URL feature. This contradicts recent research from Y. Ari Kustiawan & K. I. Ghauth (2025a) suggesting that *IsHTTPS* has limited discriminative power as phishing sites increasingly adopt HTTPS through free SSL certificate authorities such as Let's Encrypt. Y. Ari Kustiawan & K. I. Ghauth (2025a) trained their models on the PhishOFE dataset (Y. Ari Kustiawan & K. I. Ghauth 2025b) consisting of 101,063 URLs, published in September 2025, which does not explicitly state collection dates. In comparison, the PhiUSIIL dataset consisting of 235,795 URLs used in this study was collected between October 2022 and May 2023. This temporal difference may explain the discrepancy: the PhiUSIIL dataset likely captures an earlier period when HTTPS adoption among phishing sites was less prevalent. Our results indicate that *IsHTTPS* remains a valuable feature for phishing detection, particularly when combined with other URL characteristics. Rather than discarding this feature entirely, detection systems should weight it appropriately based on the distribution of HTTPS usage in their target environment.

In the absence of features such as *URLTitleMatchScore* and *LetterRatioInURL*, and the separation of *NoOfOtherSpecialCharsInURL* into three different features corresponding to the counts of digits, question marks, and ampersands, Y. Ari Kustiawan & K. I. Ghauth (2025a) demonstrated that *URLLength* and *NoOfSubDomain* are the most important features. The present study demonstrates that *NoOfOtherSpecialCharsInURL* can be an important feature if the three different types of special characters are combined to form a single feature, while *URLLength* and *NoOfSubDomain* remain important features.

While the best model trained by Y. Ari Kustiawan & K. I. Ghauth (2025a) using only URL features achieved an accuracy of 96.00%, the best model trained using only URL features in this study achieved an accuracy of 99.80%. As noted previously, some of this difference can be explained by the temporal difference between the two datasets. However, the inclusion of more sophisticated URL derived features such as *LetterRatioInURL* may enable models trained using only URL features to improve their accuracy substantially without incurring the additional latency of HTML feature extraction.

5.1.2. *URL + HTML Features*

The results indicate that classical models trained on URL and HTML features achieved the best performance of all models. The improvement in performance compared to using only URL features can be largely attributed to the inclusion of HTML-based features. The feature importance analysis (see Figure 4) supports this conclusion, showing that the Random Forest model relies predominantly on HTML-based features. These findings suggest that whenever computational resources are available, it is advantageous to include HTML-based features. However, this approach may degrade the user experience if every new webpage is first parsed by an

HTML feature extraction engine and analyzed by a model due to the additional latency.

While the 99.98% accuracy achieved by the Random Forest model trained on URL and HTML features is impressive, a qualitative analysis of the remaining 11 False Negatives (phishing URLs misclassified as legitimate) [3] reveals three distinct patterns of evasion that attackers can use to bypass feature-based detection. These are:

- **Subdomain Abuse on Legitimate Hosting Services**: A significant portion of the false negatives hosted their attacks on reputable and legitimate domains. For example, the model misclassified https://byil.blogspot.com/ and https://rariblies.blogspot.com/. These URLs possess "safe" structural features: they use HTTPS, have valid SSL certificates issued to the host (Google), and use high-reputation Top-Level Domains (TLDs). The model's URL-based features likely heavily weighted the reputation of the domains, effectively masking the malicious subdomain.
- **Typosquatting**: The model failed to flag some phishing attacks that relied on typosquatting rather than structural anomalies. An example found in the false negatives is https://www.alshayagropu.com/. This phishing attack relies on using "u" instead of "p" in "group." Since classical models do not understand words and their meaning (R. Patil et al. 2023), phishing URLs like this may be more likely to evade detection by classically trained models.
- **The "HTTPS" Blindspot**: Almost all the false negatives identified utilized the HTTPS protocol. As noted in Section 5.1.1, as attackers are increasingly able to obtain SSL certificates, the reliability of *IsHTTPS* as an indicator of a malicious website decreases. However, the model still managed to correctly identify 9,882 out of the 9,893 phishing sites that use the HTTPS protocol, demonstrating a 99.89% recall rate on the subset. This shows that *IsHTTPS* can still be an important feature in models looking to detect phishing attacks.

5.1.3. *URL String Only*

The experiment involving logistic regression using TF-IDF vectorization demonstrated that comparable results to models trained on URL-based features can be achieved without manual feature engineering. However, caution is warranted regarding the generalizability of this approach. TF-IDF models depend heavily on specific vocabulary and n-grams present in the training data. If attackers adopt new naming conventions, target different brands, or use obfuscation techniques that alter the character distribution, the model's performance is likely to degrade more rapidly than that of models based on structural features (D. Komosny 2025).

The comparison of Logistic Regression using TF-IDF vectorization when scaled and unscaled also confirmed the importance of scaling when applying logistic regression. Unscaled TF-IDF vectors exhibit high variance, which can degrade the optimization algorithm and bias the regularization term in logistic regression. This finding is consistent with established best practices indicating that scaling is beneficial for scale-sensitive models like logistic regression and support vector machines (C.-W. Hsu et al. 2003).

5.1.4. *Feature Saturation in "All Features"*

In the present study, the addition of derived features like *URLTitleMatchScore* and *TLDLegitimateProb* resulted in a saturation effect. Accuracy marginally decreased when derived features were added to URL and HTML based features for both Logistic Regression and Random Forest models. This suggests that derived features that depend on third-party sources listing legitimate websites introduce redundancy rather than unique discriminative information when HTML features are already present.

5.2. *Comparison with Original PhiUSIIL Framework*

A. Prasad & S. Chandra (2024a) reported accuracy of 99.79% using their incremental ensemble learning framework with pre-training. The results using standard batch learning approaches achieve comparable or superior performance (99.98% with Random Forest including HTML features), demonstrating that classical machine learning methods remain highly effective for this task. However, as the authors pointed out, this accuracy is highly susceptible to attackers changing their behavior. This is where a model that can be incrementally trained rather than requiring re-training every time a new training dataset is available becomes advantageous.

The purpose of this study was to identify whether the addition of each layer of features (HTML and derived) adds to the discriminative power of an existing model. The results suggest that while adding HTML-based features enhances the discriminative ability of a model significantly, derived features based on third-party datasets do not add value to classical models. As A. Prasad & S. Chandra (2024a) propose, derived features like *USI* can be advantageous as a first line of defense to quickly

[3] See Appendix B for the complete list of these 11 false negative URLs.

and efficiently classify a URL and then use more sophisticated models if the result is uncertain. However, this comes at a slight decrease in accuracy as the original PhiUSIIL framework achieved a marginally lower accuracy of 99.79% compared to the highest accuracy of 99.98% using classical models in this study. Moreover, when training models that are to be used after filtering with derived features like *USI*, the results indicate that omitting derived features might be beneficial.

5.3. *Limitations*

The most significant limitation to this study is the risk of non-generalizability of classical models trained on a specific dataset. As demonstrated by M. Mia et al. (2025), the predictive performance of models trained exclusively on URL-based features significantly deteriorates when the training and testing environments differ, even if they share the same feature set. For instance, an XGBoost model that achieved up to 99% accuracy within a single dataset dropped to accuracy levels as low as 51% and 59% when trained on one dataset and tested on another. Some of this deterioration can be mitigated by using HTML-based features in addition to URL-based features. However, it is important to recognize that some of the accuracy scores observed in this study and other studies on using ML techniques to identify phishing URLs may be highly inflated due to overfitting by models regardless of their feature configuration.

6. CONCLUSION

This study evaluated the discriminative power of different feature engineering strategies for phishing URL detection using the PhiUSIIL dataset. The results demonstrate that classical machine learning models achieve near-perfect classification performance (99.98% accuracy) when trained on URL and HTML features, matching or exceeding the performance of more complex incremental learning frameworks.

The feature importance analysis reveals that *IsHTTPS* remains the most discriminative URL feature despite recent claims of diminishing utility, though this may reflect temporal differences in dataset collection periods. HTML features such as *NoOfExternalRef*, *NoOfImage*, and *LineOfCode* provide the strongest discriminative signal when webpage content is available, substantially outperforming URL-only configurations. The TF-IDF vectorization approach demonstrates that character n-gram patterns can achieve competitive performance (99.70% accuracy) without manual feature engineering, though generalizability concerns warrant caution.

Notably, the inclusion of derived features based on third-party legitimate URL databases does not improve classification performance when HTML features are already present, suggesting a saturation effect. This finding indicates that computational resources may be better allocated to HTML extraction rather than derived feature computation for classical models.

The primary limitation of this work is the potential non-generalizability of models trained on specific datasets. Cross-dataset evaluation studies have shown that URL-based models can experience dramatic performance degradation when applied to different data sources, even with identical feature sets. Future work should focus on developing feature engineering strategies that maintain performance across diverse phishing attack patterns and temporal shifts in attacker behavior. Additionally, investigating hybrid approaches that combine the efficiency of derived features like *USI* for initial filtering with the accuracy of HTML-based models for uncertain cases presents a promising direction for practical deployment.

REFERENCES

Anti-Phishing Working Group. 2024, Phishing Activity Trends Report, 4th Quarter 2023, Tech. rep., Anti-Phishing Working Group. https://docs.apwg.org/reports/apwg_trends_report_q4_2023.pdf

Ari Kustiawan, Y., & Ghauth, K. I. 2025a, IEEE Access, 13, 126756, doi: 10.1109/ACCESS.2025.3579223

Ari Kustiawan, Y., & Ghauth, K. I. 2025b, IEEE Access, 13, 169606, doi: 10.1109/ACCESS.2025.3614126

Chen, T., & Guestrin, C. 2016, in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785–794, doi: 10.1145/2939672.2939785

Cisco Talos. n.d., PhishTank: Join the fight against phishing, https://www.phishtank.com/

DomCop. n.d., Open PageRank Initiative,, https://www.domcop.com/openpagerank/

Fette, I., Sadeh, N., & Tomasic, A. 2007, in Proceedings of the 16th International Conference on World Wide Web, WWW '07 (New York, NY, USA: Association for Computing Machinery), 649–656, doi: 10.1145/1242572.1242660

Hsu, C.-W., Chang, C.-C., & Lin, C.-J. 2003, A Practical Guide to Support Vector Classification, Tech. rep., Department of Computer Science, National Taiwan University. https://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf

Kavya, S., & Sumathi, D. 2025, Artificial Intelligence Review, 58, 50, doi: 10.1007/s10462-024-11055-z

Komosny, D. 2025, Scientific Reports, 15, 37472, doi: 10.1038/s41598-025-21384-w

Ma, J., Saul, L. K., Savage, S., & Voelker, G. M. 2009, in Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 1245–1254

Mia, M., Derakhshan, D., & Pritom, M. M. A. 2025, in Proceedings of the 11th International Conference on Networking, Systems, and Security, NSysS '24 (New York, NY, USA: Association for Computing Machinery), 137–145, doi: 10.1145/3704522.3704532

OpenPhish. n.d., OpenPhish: Global Phishing Intelligence, https://openphish.com/

Opitz, D., & Maclin, R. 1999, J. Artif. Int. Res., 11, 169–198

Patil, R., Boit, S., Gudivada, V., & Nandigam, J. 2023, IEEE Access, 11, 36120, doi: 10.1109/ACCESS.2023.3266377

Pedregosa, F., Varoquaux, G., Gramfort, A., et al. 2011, Journal of Machine Learning Research, 12, 2825

Polop, C. n.d., MalwareWorld, GitHub. https://github.com/carlospolop/MalwareWorld

Prasad, A., & Chandra, S. 2024a, Computers & Security, 136, 103545, doi: 10.1016/j.cose.2023.103545

Prasad, A., & Chandra, S. 2024b, PhiUSIIL Phishing URL (Website),, UCI Machine Learning Repository

Rao, R. S., & Pais, A. R. 2019, Neural Computing and Applications, 31, 3851, doi: 10.1007/s00521-017-3305-0

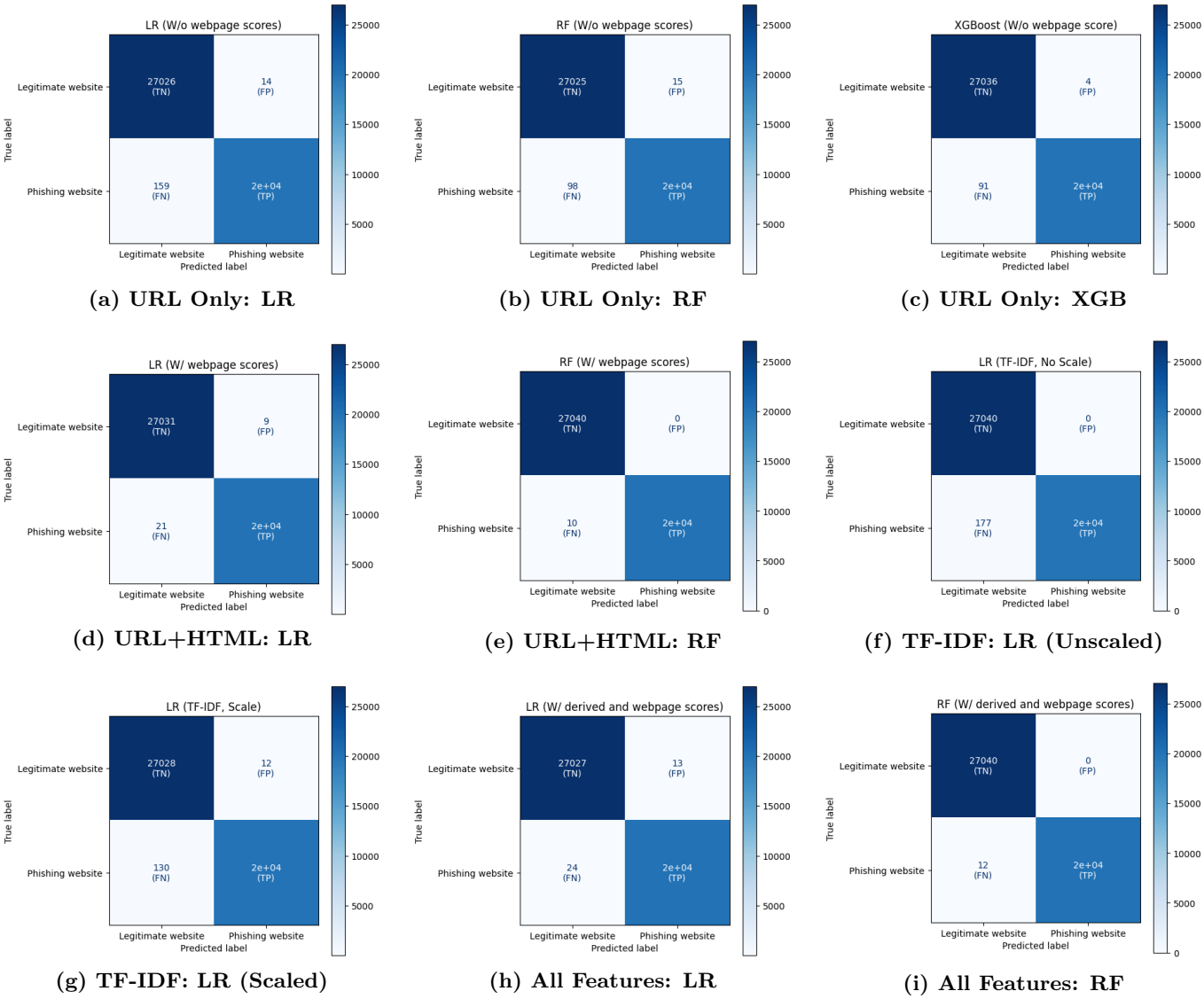
Sahingoz, O. K., Buber, E., Demir, O., & Diri, B. 2019, Expert Systems with Applications, 117, 345, doi: 10.1016/j.eswa.2018.09.029

Salton, G., & Buckley, C. 1988, Information Processing & Management, 24, 513, doi: 10.1016/0306-4573(88)90021-0

APPENDIX

A. CONFUSION MATRICES

This appendix presents confusion matrices for all models.



(a) URL Only: LR **(b) URL Only: RF** **(c) URL Only: XGB**

(d) URL+HTML: LR **(e) URL+HTML: RF** **(f) TF-IDF: LR (Unscaled)**

(g) TF-IDF: LR (Scaled) **(h) All Features: LR** **(i) All Features: RF**

Figure 7. Confusion matrices for all evaluated configurations. (a–c) URL Features Only; (d–e) URL + HTML Features; (f–g) URL String Only (TF-IDF); (h–i) All Features.

B. LIST OF FALSE NEGATIVES

Table 7 lists the 11 phishing URLs that were incorrectly classified as legitimate (False Negatives) by the Random Forest model trained on the URL + HTML feature configuration. The *Index* corresponds to the row number in the original PhiUSIIL dataset.

Table 7. False Negative URLs (Phishing sites predicted as Legitimate)

S. No	Index	URL
1	76258	https://www.vmailmessage.com
2	78962	https://www.tincanmonsters.com/
3	112781	https://byil.blogspot.com/
4	105405	https://www.interprocesoswebcancannovil.top/
5	150614	https://pha.xujmno.xyz/logi/
6	63616	https://www.alshayagropu.com/
7	30066	https://www.connextsbridge.com/
8	95625	https://www.seedjfly.top
9	58437	https://www.pencakefistaking.com/
10	28452	https://rariblies.blogspot.com/
11	63877	https://www.foroarte.es/wp/privacy-policy/logi...