Phishing URL Detection: A semantic based model

Aiman Himi

1670788

Abstract—This study uses PhiUSIIL dataset to develop a machine learning tool focusing on URL-based features for real-time phishing detection. To bypass challenges with HTML content retrieval, we designed semantic URL features, including a novel NLP-based feature, RatioNLP, that evaluates domain coherence to detect suspicious patterns. Ensemble classifiers achieved high performance, though dataset biases due to a complete oversimplification of legitimate URLs which limit generalizability. The proposed model successfuly captures suspicion as commonly percieved.

Keywords—*Phishing detection, Phishing dataset, NLP, Machine learning*

1. Introduction

Phishing URLs pose a significant security threat, often serving as gateways for malicious activities such as identity theft [1]. To address this challenge, we aimed to develop a machine learning tool leveraging the PhiUSIIL dataset [8] (2024). This dataset includes a comprehensive collection of phishing and legitimate URLs, accompanied by features such as HTML content and URL-based attributes. Our study explores the feasibility of using this dataset to create a semantic-based classifier that operates effectively with minimal external resources.

1.1. Preliminary Analysis

PhiUSIIL dataset includes 100945 phishing URLS (labeled 0) and 134850 legitimate ones (labeled 1). To evaluate the utility of the dataset, we initially analyzed its provided HTML features to assess their discriminatory power for distinguishing phishing from legitimate URLs. A Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) revealed that the HTML features could linearly separate the classes almost perfectly as seen in Fig 1.

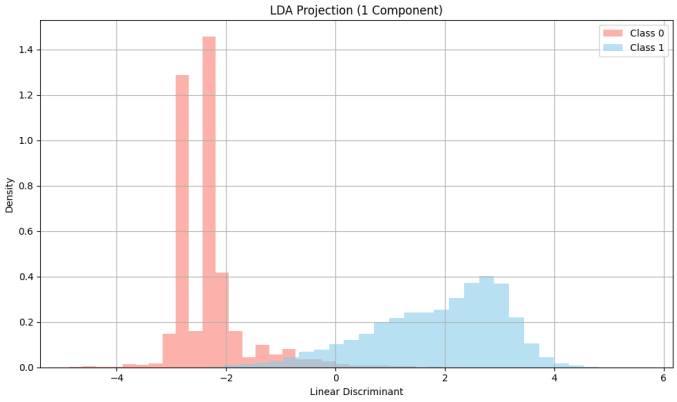

Figure 1. Linear Discriminant Analysis using HTML features.

Given these considerations, we excluded all HTML features and kept only the URL and label columns. We switched the focus to deriving from scratch features directly from the URL string. This decision not only streamlines implementation but also ensures compatibility with any dataset containing URLs.

2. Feature Engineering

We initiated with various features wildly used in literature [3], along with one novel to this work. For this work, we considered the anatomy of URLS as seen in Fig 7

2.1. URL Features

2.1.1. URL Length (U1)

The URLs of phishing frequently tend to be lengthier compared to legitimate URLs. [6]

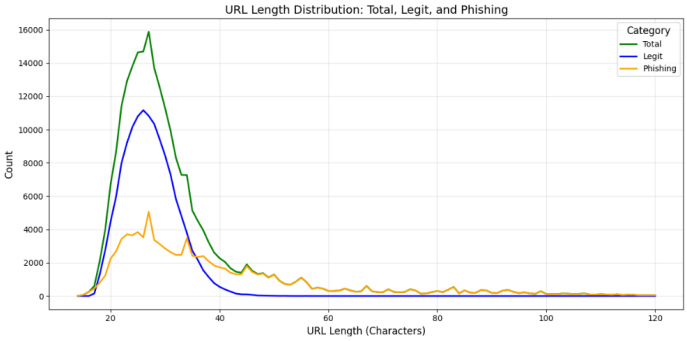

Figure 2. URL length distribution in PhiUSIIL dataset. We filtered URLs up to 120 character length for visualization purposes.

Indeed, we highlight that the mean number of characters in phishing URL's is: 46.31 and the standard deviation is 61.92. The mean number of characters in the legit URLs is: 27.21 and the standard deviation is 4.82.

While the correlation between longer URLs and phishing attempts is statistically significant, this feature must be interpreted with caution. A legitimate URL can also be long, if it includes additional like paths or query parameters. For instance, while "`https://website.com`" is a short URL, a legitimate extended version like "`https://website.com/path1?=query1...`" should remain equally valid despite its increased length.

A key issue arises from the dataset: legitimate URLs are overly simplified, often representing only the main page without additional paths or queries. This is seen in 1

Table 1. Count of URLs with presence of Path or Query

Class	Has no P/Q	Has P/Q
Phishing	40,511	60,434
Legitimate	134,851	0

All legitimate URLs lack paths or queries, which does not reflect real-world scenarios. This may mislead models to associate longer URLs with phishing, causing overfitting.

2.1.2. IsHTTPS (U2)

`IsHTTPS` is a binary attribute that checks for Secure Encrypted protocols. It plays a role in identifying potential phishing URLs, as we aim to teach the model to be cautious of web pages without encryption. The distribution of the feature highlights that all legitimate URLs in the dataset include HTTPS. On the other hand, nearly half of the phishing URLs also use HTTPS.

The dataset bias towards HTTPS may not reflect real-world scenarios, where legitimate HTTP pages exist. Despite the dataset bias, the `IsHTTPS` attribute is valuable for identifying security risks.

2.2. Host Features

Hosts are very important since registering a personalized domain can be an expensive purchase. Since most phishing campains last on average 21 h [5], it leaves most perpetrators with the only option of

recurring to free hosting providers or use a blatant URLs, (like an IP). Because of this, I hypotesize that we will extract the most meaningful information from this part of the URL.

We derived the following features:

2.2.1. CharContinuationRate (H1)

This feature is designed to quantify the obfuscation in a domain by measuring abrupt transitions between different character types (letters, numbers, symbols) within the `CoreDomain`. (Defining CoreDomain as the host without 'www. and TLD'.) We aim to capture domains such as e.g., `l1nk-f4ls-o`, `ab3d-d2a2-1sa1`, which will have a higher `CharContinuationRate` in contrast to simple legitimate domains (e.g. `google`, `4chan`...).

The `CharContinuationRate` is calculated as:

$$\texttt{CharContinuationRate} = \frac{\texttt{Num character type transitions}}{\texttt{Length of CoreDomain}}$$

This feature is particularly useful for identifying domains that utilize obfuscation techniques.

2.2.2. NoOfSubdomains (H2)

I hypothesized that the number of subdomains could positively correlate with phishing attempts, as phishing URLs often use additional subdomains to obfuscate their appearence. The dataset's distribution of subdomains for each class is summarized in Table 2.

Table 2. Distribution of Subdomains by Class

No. of Subdomains	Phishing (%)	Legitimate (%)
0	13.85	0.00
1	62.95	85.17
2	17.95	13.52
3	3.40	1.25
4+	1.85	0.04

From Table 2, we draw the following insights:

- **Legitimate URLs:** Predominantly have 1 subdomain (85.17%). The maximum observed number of subdomains is 4, but this is extremely rare (only 0.04%).
- **Phishing URLs:** Display a broader range of subdomain counts. 62.95% have 1 subdomain, 13.85% have 0 subdomains, and 5.25% have between 3-10. This variability aligns with the hypothesis that phishing URLs often use subdomains for obfuscation.

Notably, **all legitimate URLs have at least 1 subdomain**, which is a false representation of reality, as URLs like `https://example.com` (with 0 subdomains) are common in practice. This introduces a potential bias in this dataset: Absence of legitimate URLs with 0 subdomains could mislead the model to assume a false relationship.

2.2.3. LowestDomainLength (H3), LowestDomainDigits (H4)

An analysis of the ammount of characters and digits of the Lowest Level Domain would be useful to gain insight about these URL's with multiple subdomains. In this way, we are allowing a more complex discrimination model. We find that **almost all** legit URL's have the 'www' subdomain, of length 3, while the other class has a larger lowest subdomain length (8.5 mean length). Another observation was that **no legitimate** 'lowest-level' subdomain **contained numbers**, while the other class had a mean of 1.36 digits presence.

2.2.4. IsDomainIP (H5)

We created a binary feature that checks if the IP address is being used in the URL. This was done because if the attackers do not purchase a domain name, the site is only accessible by introducing the IP address in the navigation.

2.2.5. DigitsHost (H6), DigitRatioHost (H7)

An abnormal number of digits in the host could be an indicator of phishing attempt [ref]. We counted the ammount of digits and the ratio of digits over the host length. As expected, we observe that legit hosts **rarely** use numbers (mean of 0.05 digits with 0.35 std), while the usage of them is somewhat present in phishing hosts (mean of 1.99 with 3.98 std).

2.2.6. NoOfOtherSpecialInDomain (H8), SpecialCharRatioInDomain (H9)

Unlike the U3 & U4 URL features, studying the Special Characters on the Host is useful, since these parts must follow certain criteria, (Maximum length, domain avaliability, restricted characters...). I hypothesized that looking at hyphens ('-') could be specially useful, since some perpetrators might want to separate words. For example (www.realpage-freehost.com).

2.3. NLP Features

2.3.1. RatioNLP: A Novel Feature for Host Understanding (N1)

The feature`RatioNLP` is designed to measure how descriptive or understandable a URL's host is, capturing the semantic intent behind its structure. The motivation for this feature stems from the observation that some URLs may appear statistically suspicious due to their length or anatomy, such as `www.10travel-tips-in-barna.co.uk`. While this could be flagged as phishing due to its complexity, its length might instead reflect an attempt to describe specific content.

Natural Language Processing is employed to break down the domain into potential words using the `wordninja` module. This tool segments strings based on linguistic probabilities derived from large English word corpora. By comparing all possible splits, the function chooses the segmentation with the highest cumulative probability based on linguistic patterns.

Given the URL host string, it is calculated as:

$$\texttt{RatioNLP} = \frac{\sum_{w \in \texttt{actual\_words}} (\texttt{len}(w))^2}{\texttt{len(CoreDomain)} \cdot \texttt{len(split\_words)}}$$

Where:

- **actual_words:** List of actual words filtered from the output of wordninja.split that are longer than 3 characters and contain only alphabetic characters.
- **len(CoreDomain):** Length of the host without considering the Lowest Level Domain 'www' nor the Top Level Domain.
- **len(split_words):** The total number of allegedly words identified by `wordninja.split`

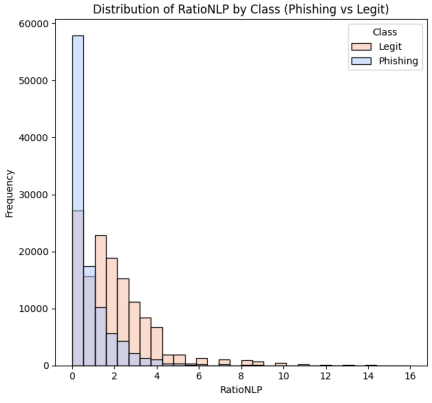

Figure 3. Distribution plot of the RatioNLP feature

The provided distribution of the `RatioNLP` feature highlights its effectiveness in distinguishing between phishing and legitimate domains:

- **Legitimate:** Domains composed entirely of meaningful words tend to have higher `RatioNLP` values, as their semantic coherence results in a larger ratio. This is evident in the range `RatioNLP > 2`, where legitimate URLs dominate.
- **Phishing:** Domains containing obfuscation techniques, such as random alphanumeric strings or short meaningless fragments, cluster around lower `RatioNLP` values (particularly in the range `RatioNLP < 1`). These patterns hinder segmentation into meaningful words, reducing the ratio.

The plot underscores the discriminative power of `RatioNLP`, making it a valuable feature for identifying phishing URLs while minimizing false negatives for phishing ones.

2.3.2. CommonWords (N2)

A dataset of 319652 frequent words in English [2] was retrieved and filtered. We counted how many of them where present in the `CoreDomain` section of the URL (defined as previously).

3. Experimental Setup

3.1. Datasets

In this work we use PhiUSIIL dataset for training and Jishnu K. S. [7] dataset for testing . These were retrieved from completely unrelated sources to ensure the performance of the model on unseen data. We ensured that the datasets did not share URLs by removing common URLs between test and train sets. Less than 450 rows were removed.

Test dataset only includes two columns; one for URL string and one for the label.

Table 3. Distribution of URL Instances by Type in Train and Test Datasets

Dataset	Legitimate	Phishing
Train	134,850 (57.2%)	100,945 (42.8%)
Test	345,738 (76.8%)	104,438 (23.2%)

3.2. Preprocessing

Both train and test datasets were preprocessed the exact same way: We first computed all the previously explained features. Then, we ensured that no errors where generated during this part; A total of 6 rows with missing values were found and deleted on the test set. Additionally, by analyzing the test data `describe()` output, we found 167 rows with a negative Number of subdomains (H1 feature). These corresponded to incorrectly stored URLs that started with double protocol: ("http://http://..."). They were removed. The train set preprocessing did not show missing or weird values. Scaling of the features was done on a copy of the dataset to train models sensitive to magnitudes (e.g. Logistic Regression). `StandardScaler` module was used.

3.3. Metric Selection

One might initially prioritize recall, ensuring that potentially malicious phishing URLs are not mistakenly classified as safe. However, in real-world scenarios, the probability of encountering a phishing URL among legitimate ones is relatively low. This shifts the focus towards balancing two critical objectives: avoiding the misclassification of legitimate URLs (a much more frequent event) while still minimizing the risk of allowing access to phishing URLs.

Given this context, optimizing the **F1-score** becomes a more suitable choice since it ensures effectiveness in identifying malicious URLs and maintaining a low false-positive rate for legit sites. This metric reflects a practical trade-off that aligns with real-world use cases.

3.4. Feature Selection

Given that we retrieved a wide array of features from the URL structure, it is likely that some features exhibit redundancies. These correlations could result in overly complex representations, potentially diminishing the model's interpretability. We assessed feature relevance through a correlation matrix analysis, which showed **all pairs** of features had correlation values **below 0.9**.

Additionally, we used a preliminary run of a tree-based model to evaluate feature importance scores.

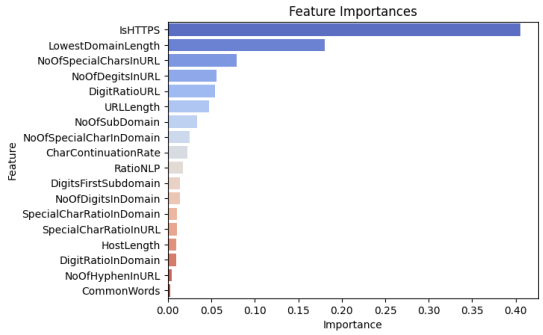

Figure 4. Feature importance of a Ra

High priority is being given to the IsHTTPS attribute and those highly related to the URL length.

Nonetheless, we retained all features. This decision was made to preserve the interpretability and exploratory nature of the study. Retaining all features allows us to hypothesize and better understand what the model is learning.

3.5. Models Training & Validation

We have selected four classifiers, widely used in the literature [4]: Logistic Regression (LR), Random Forest (RF), XGBoost (XGB), LightGMB (LGBM). We particularly focused on tree-based ensemble models such as RF, XGB, and LGBM due to their ability to handle nonlinear relationships.

The hyperparameters for XGB and LGBM were optimized using `GridSearchCV`, with stratified k-fold cross-validation and `f1` as the scoring metric. RF and LR were validated with `random_state = 42`.

We performed stratified k-fold cross-validation with `k = 5`. Stratification was used to ensure balanced representation across folds, given the slight class imbalance (see Table 3). The choice of `k = 5` asseses computational efficiency with sufficient validation splits considering the numerous instances on the dataset.

Additionally, we predicted a URL as phishing if its probability exceeded 0.9. This threshold seemed to work best in the validation process. We hypothesized that adjusting the threshold for the phishing class could improve safety for the legitimate class, which is a more common one.

For each model, the following metrics were recorded during cross-validation:

Table 4. Cross Validation Results from classification_report

Model	F1-Score (Mean)	Precision	Recall	Accuracy
LR	0.99529	0.9912	0.9994	0.9946
RF	0.99806	0.9943	0.9983	0.9986
XGB	**0.99812**	0.9966	0.9997	0.9979
LGBM	0.99811	0.9946	0.9968	0.9928

The cross-validation results demonstrate consistently excellent performance across all classifiers, with near-perfect F1-scores, precision, recall and accuracy. This indicates that that the models are learning the dataset's patterns effectively, however we must remain cautious

about potential overfitting. As highlighted in the Feature Engineering section, certain dataset biases could be artificially inflating performance.

4. Test Evaluation

When evaluating the best validated model, **XGBoost**, on the test dataset, the following metrics were observed:

Table 5. Test Dataset Classification Report for XGBoost. F1-score: 0.01537

Class	Precision	Recall	F1-Score	Support
Phishing (0)	0.23	1.00	0.38	102,897
Legitimate (1)	0.99	0.01	0.02	345,738

The overall accuracy was only 24%, with an F1-score of **0.017**. These results indicate that the model is failing to generalize, especially for legitimate URLs, despite excellent performance during validation.

4.1. Hypotheses for Model Failure

Without analyzing the test dataset, we hypothesize the cause of performance degradation based on prior observations:

- The training dataset lacks diversity in legitimate URLs, with most examples being overly simplified and devoid of paths or queries. This misrepresents real-world legitimate URL structures and teaches the model to incorrectly associate longer URLs with phishing.
- Feature importance analysis revealed high dependency on length-related attributes (e.g., `DigitsInURL`). These features, while predictive on the biased training dataset, fail to generalize to legitimate URLs with more realistic structures.

To mitigate overfitting and improve generalization we refined the feature set to reduce dependency on the train set specific biases. **All URL Length related features were discarded**. Concretely we kept 12 out of the initial features: IsHTTPS, RatioNLP, CharContinuationRate, CommonWords, NoOfSubDomains, LowestDomainLength, LowestDomainDigits, IsDomainIP, DigitsHost, DigitRatioHost, NoOfHyphenInHost, NoOfOtherSpecialInDomain, SpecialCharRatioInDomain.

4.2. Re-train & Re-validation

Considering the previous feature subset we repeated the experiments described in Models Training & Validation section. For each model we approximately obtained:

Table 6. Cross-Validation Results After Feature Refinement

Model	F1-Score (Mean)	Precision	Recall	Accuracy
LR	0.9792	0.97	0.99	0.98
RF	0.9931	0.99	0.99	0.99
XGB	0.9930	0.99	0.99	0.99
LGBM	0.**9932**	0.99	0.99	0.99

Although the performance metrics slightly dropped after refining the feature set, the models still achieve decent overall results with a F1-scores consistently high, even with reduced reliance on potentially biased features.

4.3. Re-test

We re-evaluated the models on the unseen test dataset after refining the features and adjusting the decision threshold to 0.9 for the phishing class. The results are as follows:

This time, similarly to the validation results, classifiers achieved excellent metrics, seen in maintaining high precision and recall scores. This suggests that the refined feature set effectively mitigated overfitting and improved generalization, proving our hypothesis right.

Table 7. Test Dataset Performance Metrics

Model	F1-Score	Recall	Precision	Accuracy
LR	0.9952	0.9955	0.9948	0.9925
RF	**0.9977**	0.9976	0.9978	0.9965
XGB	0.9975	0.9972	0.9979	0.9962
LGBM	0.9952	0.9924	0.9948	0.9926

5. Results

This section aims to gain insight about what the models have learned and investigate misclassified URLs to identify patterns or weaknesses in the models.

5.1. Confusion matrices

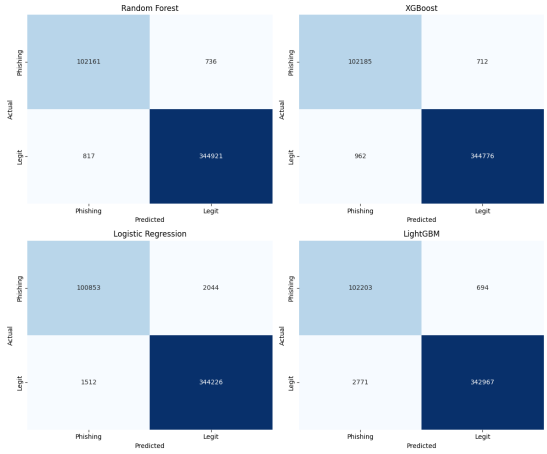

Figure 5. Confusion Matrices of the four presented models

Random Forest and XGBoost achieve high True Positive (Legit detected as Legit) and True Negative (Phishing detected as Phishing) rates, with few missclassifications. On the other hand, Logistic Regression shows a slightly higher number of False Positives and False Negatives.

5.2. AUC-ROC curves

AUC-ROC allows fair comparisons across models by summarizing their performance over all thresholds. This is important for phishing detection models, since we must often balance minimizing False Positives (avoiding unnecessary blocking of legitimate URLs) with minimizing False Negatives (avoiding undetected phishing URLs).

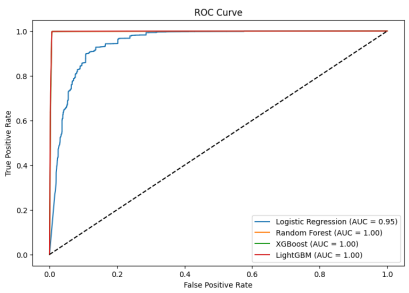

Figure 6. ROC Curves along AUC for each model

As seen in 6, ensemble models (Random Forest, XGBoost, and LightGBM) achieve near-perfect performance, as their curves are tightly hugging the top-left corner, indicating an approximate AUC value of 1.00.

For the Logistic Regression model, represented by the blue line, while it achieves an AUC of 0.95, it reflects that this model occasionally misclassifies samples more frequently.

5.3. Random Model Comparison

For numerical comparison reasons, we made predictions randomly based on a threshold equal to 0.8, meaning that a random value between 0 and 1 exceeding 0.8 would predict a phishing instance. The random model achieves on average **F1 score of 0.785**, precision of 0.7701, recall of 0.8003, and an overall accuracy of 0.6620. While reasonable due to class imbalance, they fall significantly short of the ensemble models, highlighting the ability to capture meaningful semantic patterns in the data.

5.4. Error Analysis

To understand the limitations of our best-performing model (Random Forest), we conducted a detailed analysis of the misclassified samples. This analysis revealed two significant patterns:

5.4.1. False Negatives

RF missclassifies **678** different URLs. We distinguish two types of false positives, seen in each set of Table 8

Table 8. Eight Legit Domains classified as Phishing

Set 1	Set 2
www.best-free-web-tv.911mb.com	news.bbc.co.uk
www.eicon-diva-2-01-s-t-pci.soft32.com	crain.com
www.gearlab1.eng.ohio-state.edu	lifesessions.com
www.jennashea.xanga.com	en.wikipedia.org

The first set includes URLs that are visually suspicious even to the human eye, with complex and obfuscated subdomain structures. These URLs demonstrate that the model has effectively learned to associate complex patterns and uncommon words with phishing behavior, albeit at the cost of occasionally misclassifying legitimate URLs. These misclassifications highlight the difficulty of relying solely on the semantic content for phishing detection.

The majority of false negatives (85%) fall into a second subset of URLs, shown in set 2. These include short and simple hosts. Upon closer inspection, none of these URLs begin with the "`www`" subdomain. This behavior can be attributed to initially shown dataset bias, as the training set lacked legitimate URLs without the "`www`" prefix. The absence of such examples during training led the models to overfit to this pattern.

5.4.2. False Positives

The RF model missclassifies **576** different malicious URLS as legitimate. We show 6 examples in table 9

Table 9. False Positive Hosts

www.jacquelineortega.com.br	www.canadiancmc.com
www.30netflix.com	www.paypal-servizirinnovi.it
na3.docusign.net	www.techsupportforum.com
www.functionmania.com	www.aqqloo.cloud

Most false positives appear entirely legitimate to the human eye based on inspection of the host alone. The trained models rely solely on surface-level patterns, hence making these types of classification seen in table 9 to be reasonable. Additionally, we see a smaller subset including phishing attempts by resemblance to well-known domains (e.g. 30netflix), with **minimal edits** to the original.

6. Results and discussions

This study highlights the challenges and opportunities in creating a robust classifier for phishing URL detection. By leveraging the PhiUSIIL dataset, we demonstrated the effectiveness of using semantic-based features derived from URL host sections for phishing detection. However, we also identified significant biases in the dataset that limit its generalizability to real-world scenarios.

Dataset Observations: The HTML content features within the PhiUSIIL dataset almost perfectly separate the classes, reflecting a fundamental difference in the ammount of website resources and effort between phishing and legitimate websites. Biases in the dataset—such as the absence of legitimate URLs without the www prefix or with query strings—hinder the model's ability to generalize effectively. These biases lead to overfitting when URL length or related features are heavily weighted.

Main focus: Recognizing the limitations of general URL-based features, we shifted the focus to the host section, creating semantic features CharContinuationRate, CommonWords and novel features like RatioNLP. These features enable the model to identify suspicious patterns more while avoiding reliance on misleading attributes like URL length by effectively leveraging NLP tools.

Performance: Ensemble-based classifiers, particularly Random Forest, consistently delivered robust performance. While models occasionally misclassified legitimate and phishing samples, the results reflected reasonable errors given the limitations of the dataset and lack of external context like HTML content.

Complementing the Model: This classification problem naturally suggests the use of whitelists and blacklists as complementary tools. Known safe domains can be automatically classified as legitimate, and known malicious domains can be restricted without requiring further analysis. Such lateral mechanisms could enhance the practical effectiveness of the classifier.

Finally, we built an interface that classifies the user URL input to demo its performance.

References

[1] R. M. Mohammad, F. Thabtah, and L. McCluskey, "Tutorial and critical analysis of phishing websites methods", *Computer Science Review*, vol. 17, pp. 1–24, 2015. DOI: 10.1016/j.cosrev.2015.04.001.

[2] R. Tatman, "English word frequency", 2017, Accessed: 2024-12-11.

[3] K. L. Chiew, C. L. Tan, K. Wong, K. S. Yong, and W. K. Tiong, "A new hybrid ensemble feature selection framework for machine learning-based phishing detection system", *Information Sciences*, vol. 484, pp. 153–166, 2019. DOI: 10.1016/j.ins.2019.01.064.

[4] M. Das, S. Saraswathi, R. Panda, A. K. Mishra, and A. K. Tripathy, "Exquisite analysis of popular machine learning–based phishing detection techniques for cyber systems", *Journal of Applied Security Research*, pp. 1–25, 2020. DOI: 10.1080/19361610.2020.1816440.

[5] A. Oest, P. Zhang, B. Wardman, *et al.*, "Sunrise to sunset: Analyzing the end-to-end life cycle and effectiveness of phishing attacks at scale", 2020.

[6] B. B. Gupta, K. Yadav, I. Razzak, K. Psannis, A. Castiglione, and X. Chang, "A novel approach for phishing urls detection using lexical based machine learning in a real-time environment", *Computer Communications*, vol. 175, pp. 47–57, 2021. DOI: 10.1016/j.comcom.2021.04.023.

[7] J. K. S. Kaitholikkal and A. B, "Phishing url dataset", version V1, 2024. DOI: 10.17632/vfszbj9b36.1.

[8] A. Prasad and S. Chandra, "Phiusiil phishing url (website) [dataset]", 2024. DOI: 10.1016/j.cose.2023.103545.

7. Appendix

7.1. Anatomy

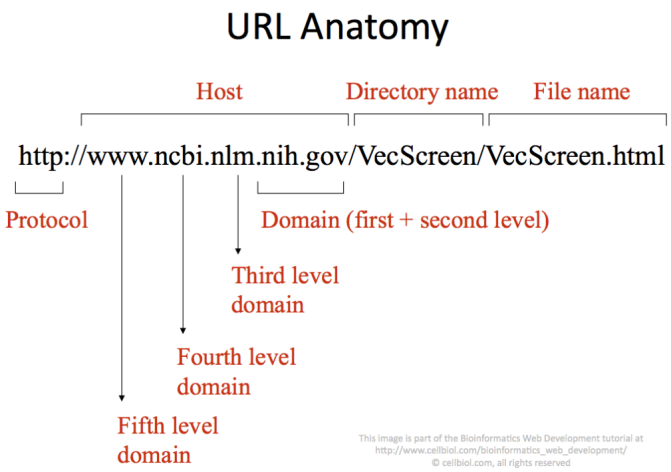

Figure 7. URL anatomy. Retrieved from cellbiol bioinformatics