

Lightweight URL-Only Phishing Detection with Domain-Grouped Validation

Abhishek Gopalakrishnan¹, Arijit Mukherjee², Aditya Soni³, Asst Prof. Pooja Mehta⁴

^{1,2,3}Students, Master of Computer Applications, D Y Patil Schools of Humanities & Sciences, Navi Mumbai, India

⁴Mentor, Master of Computer Applications, D Y Patil Schools of Humanities & Sciences, Navi Mumbai, India.

Abstract - Phishing detectors are often tested with a random train-test split. In such a split, URLs from one website can land in both the training and the test set, which makes the scores look better than they really are. In this paper we test four standard classifiers, namely Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), and XGBoost (XGB), on the public PhiUSIIL dataset, keeping all URLs of a domain on the same side of the split. We use only 18 features that can be computed from the URL text itself, and we drop one feature, URLSimilarityIndex, because it equals 100 for every legitimate URL and so almost reveals the label. Features are ranked with mutual information, and each model is tested with 5, 8, 12, and 18 features over three random splits. RF with 12 features gave the best mean F1-score (0.99736), DT with 12 features was almost as good (0.99698) with a much smaller model, and LR with 12 features scored 0.99601 using only about 2 KiB. Differences between 12 and 18 features are as small as the variation between splits. The results show that fewer features can keep strong accuracy on this dataset, but they do not prove the same performance on new or independent URLs.

Key Words: Phishing detection, URL classification, Feature selection, Mutual information, Domain-grouped validation, Machine learning, PhiUSIIL.

1. INTRODUCTION

Phishing sites copy trusted services to steal passwords, card details, and other private data. A URL classifier can quickly flag risky links, but a single accuracy figure is not enough. A useful detector should rely only on information that is available when a link is first seen, be tested fairly, and be small and fast enough to run in practice.

A common problem is domain overlap: if URLs from one domain appear in both training and test data, a model can memorise that domain instead of learning general warning signs. We avoid this by grouping rows by domain when splitting, and we use only URL-text features, not webpage content. We ask four questions: (RQ1) how does the number of features change performance; (RQ2) which classifier is best; (RQ3) how do training time, prediction time, and model size change; and (RQ4) how should results be read when

the dataset has a feature that almost gives away the label and no outside test set exists?

2. RELATED WORK

Many studies classify phishing from URL strings. Sahingoz et al. [3] showed that word-based and lexical URL features work well with tree-based models, which supports our URL-only setting. Hannousse and Yahiouche [4] compared benchmark datasets and showed that the source of the features (URL or page content) strongly affects what a detector can be used for. Prasad and Chandra [1], [2] built the PhiUSIIL dataset of 235,795 URLs, with both URL and page features, and proposed a similarity-based framework. We do not repeat that framework. We instead ask how small a URL-only feature set can become under a domain-grouped test. We use Random Forests [5], XGBoost [6], and scikit-learn [7], and rank features with a simple filter method of the kind described in [8].

3. DATASET AND FEATURE AUDIT

We use the PhiUSIIL Phishing URL (Website) dataset from the UCI repository (CC BY 4.0) [1]. It has 235,795 rows: 100,945 phishing URLs and 134,850 legitimate ones. Our CSV has 56 columns, no missing values, and no exact duplicate rows; 425 rows repeat a URL that appeared earlier but differ in other columns. We label phishing as the positive class ($y = 1$), so precision, recall, and F1 all describe phishing detection.

Features that need the webpage (line counts, HTML elements, forms, titles, and so on) were removed, since a real URL-only detector cannot use them. The raw text fields (FILENAME, URL, Domain, TLD, Title) were also removed. We then found that URLSimilarityIndex is exactly 100 for all 134,850 legitimate rows but varies for phishing rows, so it nearly separates the classes by itself. We excluded it as a cautious choice; this does not mean the feature is useless in other settings.

The 18 features kept come directly from the dataset and describe: URL length (URLLength); the domain (DomainLength, IsDomainIP, TLDLength, NoOfSubDomain); obfuscation (HasObfuscation, NoOfObfuscatedChar, ObfuscationRatio); letters and digits (NoOfLettersInURL, LetterRatioInURL,

NoOfDegitsInURL, DigitRatioInURL); special characters (NoOfEqualsInURL, NoOfQMarkInURL, NoOfAmpersandInURL, NoOfOtherSpecialCharsInURL, SpacialCharRatioInURL); and HTTPS use (IsHTTPS). Spellings follow the dataset.

4. METHOD

4.1 Splitting

We used GroupShuffleSplit (80/20) with the dataset's Domain field as the group, and seeds 42, 123, and 2026. All rows of a domain go to either training or test. The test sets had 46,721, 46,852, and 48,233 rows, with 42.34%–44.13% phishing. This is stricter than a random split, but Domain is a hostname, so different subdomains of one site can still end up on opposite sides. The split is also not a test over time or on another dataset.

4.2 Feature ranking

Mutual information (MI) was computed on a random 5,000-row sample of each training set only, and the top $k = 5, 8, 12$, or 18 features were kept. Test data were never used for ranking or training. MI looks at one feature at a time, so it cannot detect overlap between features.

4.3 Models

All models used fixed settings with no tuning. LR: standardised inputs, LBFGS, up to 500 iterations, balanced class weights. DT: balanced weights, minimum leaf size 2. RF: 50 trees, square-root feature sampling, balanced-subsample weights. XGB: 50 rounds, depth 6, learning rate 0.1, 0.8 row and column sampling, $L2 = 1.0$, and scale_pos_weight set from the training data. Random states matched the split seeds. The software used was Python 3.13, scikit-learn 1.8.0, pandas 2.2.3, NumPy 2.3.5, and XGBoost 3.1.3.

4.4 Metrics

We report accuracy, phishing precision, recall, F1, and ROC-AUC, plus training time, median prediction time per URL (three batch runs of up to 10,000 rows), and the saved joblib file size. Prediction time covers numeric features only and leaves out the time to extract features from a URL. Timings depend on the machine and are for comparison only. Scores are averaged over the three splits with the sample standard deviation (SD). The three splits come from the same dataset, so the SD only describes variation here and is not a confidence interval for real-world use.

5. RESULTS

5.1 Accuracy

RF with 12 features had the highest mean F1 (0.99736 ± 0.00026), with precision 0.99897, recall 0.99576, accuracy 0.99774, and ROC-AUC 0.99844 (Table -1). DT with 12 features was close (0.99698 ± 0.00025), and LR with 12 features reached 0.99601 ± 0.00024 . Going from 12 to 18 features changed F1 by about 0.0002 at most, which is similar to the SD, so the two sizes perform alike (Fig -1). With only 5 features, scores dropped, and LR was clearly worse and less stable (0.96848 ± 0.01653).

Table -1: Results on domain-grouped test sets (mean of three splits)

Model	k	F1 mean±SD	Precision	Recall	Accuracy	ROC-AUC
DT	5	0.99062±0.00334	0.99533	0.98596	0.99197	0.99687
LR	5	0.96848±0.01653	0.97139	0.96559	0.97292	0.99275
RF	5	0.99069±0.00319	0.99542	0.98601	0.99203	0.99721
XGB	5	0.98741±0.00487	0.99016	0.98469	0.98919	0.99714
DT	8	0.99419±0.00014	0.99783	0.99058	0.99503	0.99703
LR	8	0.99110±0.00069	0.99536	0.98688	0.99240	0.99760
RF	8	0.99427±0.00009	0.99791	0.99066	0.99510	0.99737
XGB	8	0.99354±0.00026	0.99610	0.99100	0.99447	0.99772
DT	12	0.99698±0.00025	0.99870	0.99526	0.99741	0.99748
LR	12	0.99601±0.00024	0.99955	0.99249	0.99659	0.99846
RF	12	0.99736±0.00026	0.99897	0.99576	0.99774	0.99844
XGB	12	0.99507±0.00019	0.99772	0.99244	0.99578	0.99814
DT	18	0.99683±0.00021	0.99856	0.99510	0.99728	0.99739
LR	18	0.99597±0.00028	0.99955	0.99242	0.99656	0.99847
RF	18	0.99718±0.00028	0.99889	0.99547	0.99758	0.99849
XGB	18	0.99512±0.00009	0.99777	0.99249	0.99582	0.99812

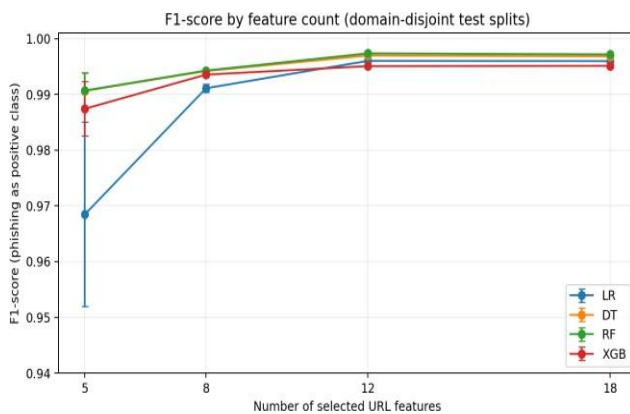


Fig -1: Mean F1-score by number of features. Error bars show SD over three splits.

5.2 Cost

LR models were the smallest (1.80–2.44 KiB) and RF models the largest (3.27–5.56 MiB). DT with 12 features trained in 0.29 s, took 0.108 μ s per URL for prediction, and used 78.0 KiB. RF with 12 features was the most accurate but used about 4.96 MiB and 2.97 μ s per URL. LR with 12 features used only about 2.14 KiB, a good choice when a very small drop in F1 is acceptable (Table -2, Fig -2).

Table -2: Training time, prediction time, and model size (mean of three splits)

Model	k	Train time (s)	Prediction (μ s/URL)	Size (KiB)
DT	5	0.135	0.097	70.02
LR	5	0.599	0.077	1.80
RF	5	2.031	2.957	3767.98
XGB	5	0.267	0.414	135.00
DT	8	0.176	0.095	54.03
LR	8	1.212	0.079	1.95
RF	8	2.176	3.302	3349.81
XGB	8	0.319	0.396	147.67
DT	12	0.291	0.108	78.00
LR	12	0.808	0.101	2.14
RF	12	2.445	2.967	5079.76
XGB	12	0.348	0.401	135.49
DT	18	0.344	0.122	85.08
LR	18	0.840	0.099	2.44
RF	18	2.632	3.315	5693.35
XGB	18	0.425	0.422	134.86

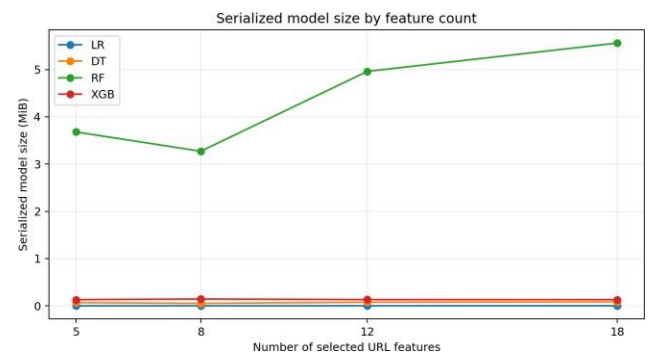


Fig -2: Saved model size by number of features (MiB).

5.3 Which features were chosen

The same features were ranked highest in all three splits. LetterRatioInURL, NoOfOtherSpecial Chars In URL, IsHTTPS, and SpacialCharRatioInURL were always in the top five. URLLength, DeditRatioInURL, NoOfDeditsInURL, and NoOfLettersInURL joined them in the top eight, and DomainLength, NoOfSubDomain, and NoOfEqualsInURL in the top twelve. Only the last place at $k = 12$ varied (TLDLength or NoOfQMarkInURL). Several of these features overlap, because letter, digit, and special-character shares of a URL add up to roughly a whole, so the top-k lists are ranked candidates, not a minimal set.

6. DISCUSSION

RQ1. Cutting from 18 to 12 features caused no measurable loss; the small gains for DT and RF are within the SD and should not be called improvements. Five features were clearly too few.

RQ2. RF-12 scored highest, but DT-12 was within 0.0004 with a far smaller model. We ran no significance test, so this gap is not shown to be real. XGB scored lower in several cases, but it used only 50 untuned rounds, so this says little about boosting in general.

RQ3. The type of model changed cost far more than the number of features: LR was tiny, DT was in the middle, and RF was largest and slowest. The best choice depends on memory limits, speed needs, and how many missed phishing links are acceptable.

RQ4. A model trained on all columns would not be URL-only, and URLSimilarityIndex nearly reveals the label here. Removing it helps but does not remove all bias. For example, IsHTTPS ranking so high may reflect how the two URL groups were collected rather than a real phishing trait. Also, about 43% of our test URLs are phishing, much more than in real traffic, so with

realistic rates precision would be lower. High scores on one dataset are not proof of live-web performance.

7. LIMITATIONS

- Only one public dataset was used; there was no test over time or on a different dataset.
- Groups use the Domain field, so hostnames of one registered domain may still be split across train and test.
- Features were taken as supplied and not re-computed from raw URLs; the dataset's collection process may make classes easier to separate.
- Only one filter method (MI on 5,000 rows) was used, with no comparison to other methods and no handling of overlapping features.
- Models were not tuned or calibrated. False-positive rate, PR-AUC, results at realistic phishing rates, significance tests, and error analysis were not done.
- Three splits from one dataset give only a rough idea of variation. Speed and size values depend on the machine and exclude feature extraction.

8. CONCLUSIONS

On the PhiUSIIL dataset with domain-grouped splits, a reduced set of URL-only features kept very high accuracy. RF with 12 features had the best F1 (0.99736), DT with 12 features was nearly as good with a model of about 78 KiB, and LR with 12 features was smallest at about 2.14 KiB. We removed URLSimilarityIndex because it equals 100 for all legitimate URLs. These findings are limited to this dataset. Future work should test on independently collected URLs, group by registered domain, split by time, report false-positive rates at realistic phishing rates, and try adversarially modified URLs.

DATA AVAILABILITY

The dataset is available from the UCI repository (<https://archive.ics.uci.edu/dataset/967/phiusiil+phish> in `g+url+website+dataset`) under CC BY 4.0, and the dataset creators should be cited [1], [2]. The code and per-run results are available from the authors on request.

ACKNOWLEDGEMENT

The authors thank their mentor, Dr. Pooja Mehta, for her guidance during this work.

REFERENCES

- [1] A. Prasad and S. Chandra, "PhiUSIIL Phishing URL (Website) Dataset," UCI Machine Learning Repository, 2024. [Online]. Available: <https://archive.ics.uci.edu/dataset/967/phiusiil+phishing+url+website+dataset>
- [2] A. Prasad and S. Chandra, "PhiUSIIL: A diverse security profile empowered phishing URL detection framework based on similarity index and incremental learning," *Computers & Security*, vol. 136, Art. no. 103545, 2024, doi: 10.1016/j.cose.2023.103545.
- [3] O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, "Machine learning based phishing detection from URLs," *Expert Systems with Applications*, vol. 117, pp. 345–357, 2019.
- [4] A. Hannousse and S. Yahiouche, "Towards benchmark datasets for machine learning based website phishing detection: An experimental study," *Engineering Applications of Artificial Intelligence*, vol. 104, Art. no. 104347, 2021.
- [5] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [6] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, 2016, pp. 785–794.
- [7] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [8] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.