Machine Learning versus Simple Rule-Based Baselines for Phishing Website Detection: Effects of Domain-Grouped Evaluation and Feature Sources

Sharv Magdum
Department of Master of Computer Applications
DY Patil Institute of Master of Computer Applications and Management
Akurdi, Pune, India

Aditya Jadhav
Department of Master of Computer Applications
DY Patil Institute of Master of Computer Applications and Management
Akurdi, Pune, India
adityasjadhav1@gmail.com

Aniket Khandagale
Department of Master of Computer Applications
DY Patil Institute of Master of Computer Applications and Management
Akurdi, Pune, India

Harshal Pawar
Department of Master of Computer Applications
DY Patil Institute of Master of Computer Applications and Management
Akurdi, Pune, India

Abstract—Phishing detectors are often judged by random-split accuracy on public benchmarks, against weak baselines or none. Using the Hannousse and Yahiouche dataset (11,430 URLs, 87 features), we examine how much machine learning (ML) improves on simple rule-based baselines, how much of that accuracy survives when related URLs are kept apart, and which feature groups carry it. We compare a six-rule URL heuristic, a domain blacklist, a one-split decision stump and shallow decision trees with Logistic Regression, Random Forest and XGBoost, using repeated 5-fold cross-validation (15 folds) with random folds and with class-balanced folds grouped by registered domain. XGBoost on all features reached 97.08 ± 0.32% accuracy with random folds and 95.17 ± 1.06% with domain-grouped folds, against 64.3% for the rule heuristic and 78.1% for the blacklist (random folds only; zero recall by construction in grouped folds). The strongest simple baseline, a one-split stump on the external feature google_index, reached 86.5%. Domain grouping lowered tree-model accuracy by 1.6 to 1.9 points with all features and by 3.9 to 4.2 points with URL-only features, and reduced XGBoost's true-positive rate at 1% false positives from 92.7% to 83.1%. External-service features were the most informative group, and URL-only XGBoost (87.5% on grouped folds) was not significantly better than the stump. A dataset audit shows sampling artifacts. ML clearly helps, but part of the apparent gain reflects baseline choice, dataset sampling and external lookups.

Index Terms—phishing detection, machine learning, evaluation, domain-grouped cross-validation, feature ablation, dataset bias.

I. Introduction

Phishing websites imitate legitimate services to steal credentials and money. Classical defenses are static: blacklists of known bad URLs and fixed rules over URL patterns. Their weakness is time. In an early study of the blacklists then in use, Sheng et al. found that most caught under 20% of phishing sites at hour zero, 47% to 83% listed them after 12 hours, and 63% of campaigns lasted under two hours [1]. A model that learns patterns can in principle flag a site it has never seen; Butnaru et al., for example, report that a lightweight URL-based model beat Google Safe Browsing in all their experiments and still worked on URLs active a year after training [2].

Reported ML accuracy, however, depends heavily on how it is measured. Arp et al. list sampling bias among the pitfalls that make learning-based security systems look better than they are [3], and surveys note that many phishing studies rely on self-built datasets whose results are hard to validate [4]. Detectors also degrade over time [5], can be evaded [6], [7], and may depend on third-party services that are slow or unavailable at run time [8]. Two questions follow: how large is the ML advantage over genuinely simple baselines, and how much of it survives when related URLs are kept out of the training data?

We study these questions on one public benchmark. **RQ1:** How much does ML improve on simple rule-based baselines (hand-written rules, a domain blacklist, a one-split stump and shallow trees)? **RQ2:** How much does accuracy fall when folds are grouped by registered domain rather than split at random? **RQ3:** Which feature groups carry the accuracy, and how much depends on external reputation lookups? Our contributions are (i) an audit of the dataset's composition, (ii) a class-balanced, registered-domain-grouped evaluation with stronger simple baselines, (iii) a full set of metrics including the true-positive rate at 1% false positives and prevalence-adjusted precision, and (iv) an all-combination feature ablation with paired tests.

II. Related Work

Static defenses. Blacklists are fast and precise for known threats but reactive [1]. Rule-based systems need no retraining but cannot model feature interactions. Jasim reports a 13-rule heuristic framework with 86.53% accuracy, 98.90% precision, 0.82% false-positive rate (FPR) and 73.89% recall on a 200,000-URL corpus [9]. Our hand-written heuristic (Section III-C) is much weaker than this and should be read as a lower bound for rule-based methods.

ML on benchmark data. Hannousse and Yahiouche introduced the 87-feature scheme used here and reported a best Random Forest accuracy of 96.61% with hybrid features [10]. Their public dataset has 11,430 balanced URLs with 56 URL, 24 content and 7 external-service features [11]. Other studies report accuracies near or above 99% with tree ensembles on other feature sets, for example 99.17% for XGBoost on 25 hybrid URL and hyperlink features [12]. Figures differ across studies because datasets, features and splits differ.

Evaluation, drift and robustness. Ejaz et al. show that a vanilla neural network's performance deteriorated by over 20.65% over time, whereas continual learning limited deterioration to 2.45% [5]. Sánchez-Paniagua et al. report lower accuracy when models trained on 2016 data were tested on 2020 URLs [8], and Yuan et al. report detectors falling below

1% detection on phishing sites from a different region [14]. Apruzzese et al. find statistically significant degradation of 3 to 10% under realistic evasion [6], and Kulkarni et al. find significant drops against adversarially generated pages [13].

External-service features. Sánchez-Paniagua et al. note that WHOIS domain age and Google or Alexa rank depend on third-party services whose slowdowns or outages can make real-time use infeasible [8]. Abuadbba et al. argue the other side: page-level features are prone to adaptive attacks, whereas trusted-service records are harder for phishers to tamper with, and they report 96.1% to 98.8% accuracy with such features [7].

Gap. We did not find a verified study that evaluates this benchmark with registered-domain-grouped folds against simple rule baselines. Our literature search used a limited set of queries, so related work may have been missed.

III. Methodology

A. Dataset and Audit

We use the Web Page Phishing Detection dataset [11], [10]: 11,430 URLs (5,715 legitimate, 5,715 phishing) with 87 features, of which 56 describe the URL, 24 the page content and 7 come from external services (WHOIS registration, domain age, DNS record, web traffic, Google index, page rank, registration length). As described in [10], legitimate URLs came from the top 500 Alexa domains (12 crawled URLs each) and a Yandex-derived list, and phishing URLs from PhishTank and OpenPhish, with duplicates removed and at most 12 URLs per domain name kept; collection took place in March 2020. Phishing is the positive class.

Our audit found one exact duplicate URL, 8,241 distinct hostnames and 6,634 registered domains (public-suffix based). Legitimate URLs span 3,970 registered domains and phishing URLs 2,688; 24 domains occur in both classes. A few registered domains hold many URLs: blogspot.com (383), duilawyeryork.com (351), appspot.com (265), wikipedia.org (177) and google.com (151). Table I shows class differences that look like collection artifacts rather than phishing behavior. External features use sentinel values (for example −1 for unavailable domain age), which we left unchanged.

TABLE I
Dataset audit: share of URLs with the property (%), by class

Property	Legitimate	Phishing
Hostname begins with www.	66.8	20.5
google_index = 0	83.1	10.1
web_traffic = 0	14.6	63.2
page_rank = 0	1.8	44.9
domain_age = −1 (unavailable)	14.3	16.8
Registered domains (count)	3,970	2,688

B. Evaluation Protocol

Setting A (random folds) uses stratified 5-fold cross-validation. **Setting B (domain-grouped folds)** uses class-balanced grouped 5-fold cross-validation (scikit-learn StratifiedGroupKFold) with the registered domain as group, so no registered domain appears in both training and test data. Each setting was repeated three times (seeds 100–102), giving 15 test folds. Every fold has 2,286 test URLs in Setting A; in Setting B the training size is 9,144 ± 215 and the test set has 49.8 ± 4.4% phishing URLs (range 41.8% to 57.3%), so test class balance varies. In Setting A about 493 registered domains per fold appear in both training and test data; in Setting B none do. Grouping by registered domain does not separate near-duplicate pages hosted on different domains, and it places unrelated sites on shared hosting platforms in the same group. It is not a temporal split.

C. Baselines

The **rule heuristic** flags a URL if at least two of six rules hold: IP-address host, "@" in the URL, length above 75, no https, more than three dots in the host, or a hyphen in the host. The rules were fixed in advance and not tuned. The **domain blacklist** stores the registered domains of phishing URLs in the training folds and flags a test URL only if its domain is listed; it is simulated from the dataset, not a real service. The **decision stump** (one split) and **depth-3 decision trees** (all features, and URL-only) are trained on the training folds. They are supervised but extremely simple, and serve as stronger rule-like baselines than hand-written rules; they are not published systems.

D. Models, Features and Metrics

We train Logistic Regression (standardization fitted inside each training fold, 3,000 iterations), Random Forest (300 trees) and XGBoost (300 trees, depth 6, learning rate 0.1) with a 0.5 decision threshold and no tuning, on three feature sets: all 87 features, no external features (80) and URL-only (56). XGBoost was also run on content-only (24), external-only (7), URL+external (63) and content+external (31). Metrics are defined as follows, with TP, TN, FP and FN the confusion counts.

$$Accuracy = (TP + TN) / (TP + TN + FP + FN) \quad (1)$$

$$Precision = TP / (TP + FP), \; Recall = TP / (TP + FN) \quad (2)$$

$$FPR = FP / (FP + TN), \; F1 = 2PR / (P + R) \quad (3)$$

We also report ROC-AUC and the true-positive rate at 1% FPR (TPR@1%FPR, from the interpolated ROC curve). Here P and R are precision and recall. Precision depends on class prevalence, so we also compute implied precision at other prevalences as $\text{recall}\cdot p / (\text{recall}\cdot p + \text{FPR}\cdot(1 - p))$ from the mean recall and FPR.

E. Statistics

We report mean ± standard deviation (SD) over the 15 folds. Folds overlap, so the SD understates true uncertainty. For paired comparisons within a setting we use the corrected resampled t-test of Nadeau and Bengio [15] on fold accuracies. We report ten comparisons with unadjusted p-values; with a Bonferroni threshold of 0.005, only $p < 0.005$ should be treated as clear. Settings A and B use different test populations, so we report their differences but do not test them.

IV. Results

A. ML versus Simple Baselines (RQ1)

TABLE II
Setting A (random folds): mean over 15 folds, %

Method	Acc. ± SD	Prec.	Rec.	F1	FPR
Rule heuristic	64.28 ± 0.93	77.79	39.98	52.81	11.43
Domain blacklist	78.08 ± 0.79	91.52	61.90	73.85	5.74
Decision stump	86.47 ± 0.41	84.16	89.87	86.92	16.92
Depth-3 tree (URL)	77.08 ± 1.16	71.40	90.46	79.80	36.30
Depth-3 tree (all)	90.97 ± 0.45	91.73	90.08	90.88	8.15
LogReg (URL)	84.85 ± 0.36	86.87	82.13	84.43	12.44
RF (URL)	90.92 ± 0.54	90.59	91.34	90.96	9.50
XGB (URL)	91.74 ± 0.47	91.73	91.76	91.74	8.29
LogReg (all)	94.52 ± 0.38	94.78	94.23	94.50	5.20
RF (all)	96.63 ± 0.25	96.64	96.62	96.63	3.36
XGB (all)	97.08 ± 0.32	97.14	97.02	97.08	2.86

TABLE III
SETTING B (DOMAIN-GROUPED FOLDS): MEAN OVER 15 FOLDS, %

Method	Acc. ± SD	Prec.	Rec.	F1	FPR
Rule heuristic	64.30 ± 4.6	76.69	39.67	51.99	11.52
Domain blacklist	50.17 ± 4.42	n/a	0.00	0.00	0.00
Decision stump	86.44 ± 0.96	83.93	89.84	86.77	16.96
Depth-3 tree (URL)	75.98 ± 2.93	74.27	80.18	76.47	28.20
Depth-3 tree (all)	90.07 ± 1.68	90.71	89.36	89.95	9.06
LogReg (URL)	84.18 ± 3.78	86.64	80.27	83.21	12.28
RF (URL)	87.01 ± 2.53	87.23	86.44	86.79	12.66
XGB (URL)	87.53 ± 3.43	88.91	85.51	87.09	10.58
LogReg (all)	93.79 ± 0.99	94.32	93.21	93.74	5.66
RF (all)	95.04 ± 0.57	95.49	94.52	95.00	4.50
XGB (all)	95.17 ± 1.06	95.91	94.38	95.12	4.10

The rule heuristic reached 64.3% in both settings, with about 40% recall and 11.4% FPR, far below the published rule system [9]. The domain blacklist reached 78.1% with random folds (precision 91.5%, recall 61.9%, FPR 5.7%) and, by construction, zero recall with grouped folds, so its 50.2% accuracy there only reflects class balance. The decision stump was the strongest simple baseline at 86.5% and 86.4%. A stump fitted to the full dataset splits on google_index at threshold 0.5, and the fold-level mean accuracy (86.5%), recall (89.9%) and FPR (16.9%) coincide with what a single split on google_index implies from Table I; per-fold split features were not logged. google_index was also ranked first by all four filter methods in [10]. Depth-3 trees reached 91.0% and 90.1% on all features, but only 77.1% and 76.0% on URL-only features (FPR 36.3% and 28.2%).

XGBoost on all features reached 97.08 ± 0.32% (A) and 95.17 ± 1.06% (B). Table VII shows it beat the stump by 10.6 and 8.7 points and the depth-3 tree by 6.1 and 5.1 points (all $p < 0.001$), and URL-only XGBoost beat the hand-written heuristic by 27.5 and 23.2 points. With URL-only features, however, XGBoost beat the stump by 5.3 points on random folds ($p < 0.001$) but by only 1.1 points on grouped folds ($p = 0.62$). Random Forest on all features (96.63%) is close to the 96.61% reported in [10] with 10-fold cross-validation and no domain grouping.

B. Random versus Domain-Grouped Folds (RQ2)

TABLE IV
ROC-AUC AND TPR@1%FPR (%), MEAN OVER 15 FOLDS. A = RANDOM, B = DOMAIN-GROUPED FOLDS

Model	AUC A	AUC B	TPR A	TPR B
LogReg (URL)	93.38	91.56	52.56	47.03
RF (URL)	96.93	94.52	65.92	48.10
XGB (URL)	97.31	94.77	66.28	57.81
LogReg (all)	98.55	98.13	76.87	71.15
RF (all)	99.38	98.78	91.18	74.74
XGB (all)	99.48	99.01	92.69	83.13

TABLE V
ACCURACY DROP FROM SETTING A TO B (PERCENTAGE POINTS)

Features	LogReg	RF	XGBoost
All 87	0.73	1.59	1.91
No external	2.02	2.46	3.03
URL-only	0.67	3.91	4.21

Grouping by registered domain lowered Random Forest and XGBoost accuracy by 1.6 and 1.9 points with all features, 2.5 and 3.0 points without external features, and 3.9 and 4.2 points with URL-only features (Table V). Logistic Regression changed little (0.7 to 2.0 points). Variability rose sharply: the SD of XGBoost accuracy grew from 0.32 to 1.06 points (all features) and from 0.47 to 3.43 points (URL-only). ROC-AUC fell only slightly (XGBoost, all features: 99.48% to 99.01%), but the true-positive rate at 1% FPR fell from 92.7% to 83.1% for XGBoost and from 91.2% to 74.7% for Random Forest with all features, and from 65.9% to 48.1% for Random Forest with URL-only features (Table IV). Accuracy at the default threshold therefore understates the loss at a strict false-positive budget.

C. Feature Groups (RQ3)

TABLE VI
XGBOOST ACCURACY BY FEATURE SET (MEAN ± SD, %)

Feature set	No.	A: random	B: grouped
URL-only	56	91.74 ± 0.47	87.53 ± 3.43
Content-only	24	88.92 ± 0.71	83.97 ± 2.26
External-only	7	94.1 ± 0.51	87.23 ± 4.84
URL + external	63	96.84 ± 0.2	94.69 ± 1.42
Content + external	31	95.25 ± 0.44	91.99 ± 1.82
No external	80	94.46 ± 0.4	91.43 ± 2.85
All features	87	97.08 ± 0.32	95.17 ± 1.06

External features were the most informative group, as also found in [10] (94.09% for Random Forest on external features alone). The seven external features alone reached 94.1 ± 0.5% (A) and 87.2 ± 4.8% (B), and URL+external (63 features) was close to all 87 features (96.84% versus 97.08% in A, $p = 0.074$; 94.69% versus 95.17% in B, $p = 0.27$). Removing external features cost 2.6 points in A and 3.7 points in B ($p = 0.013$ in B), and removing content features as well (URL-only) cost a further 2.7 and 3.9 points. Content-only features were weakest (88.9% and 84.0%).

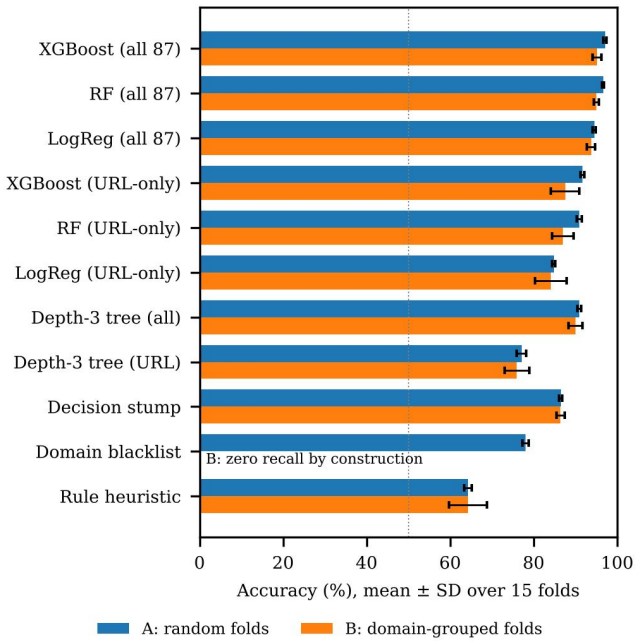


Fig. 1. Accuracy (mean ± SD over 15 folds) of simple baselines and ML models with random folds (A) and domain-grouped folds (B). The blacklist has zero recall in B by construction and is not plotted for B.

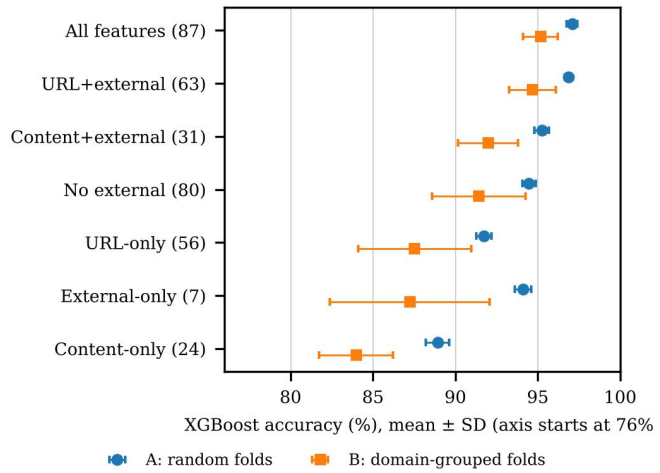


Fig. 2. XGBoost accuracy by feature set (number of features in parentheses), mean ± SD over 15 folds. The horizontal axis starts at 76%.

TABLE VII
PAIRED DIFFERENCES IN ACCURACY, POINTS (P-VALUE)

Comparison	A: random	B: grouped
XGB − RF (all)	+0.45 (0.021)	+0.13 (0.824)
XGB − LogReg (all)	+2.56 (<0.001)	+1.37 (0.006)
XGB all − stump	+10.61 (<0.001)	+8.73 (<0.001)
XGB all − tree-3 (all)	+6.12 (<0.001)	+5.10 (<0.001)
XGB URL − stump	+5.26 (<0.001)	+1.09 (0.618)
XGB URL − rule heur.	+27.46 (<0.001)	+23.23 (<0.001)
XGB URL − tree-3 (URL)	+14.65 (<0.001)	+11.55 (<0.001)
XGB all − no external	+2.62 (<0.001)	+3.74 (0.013)
XGB no ext. − URL-only	+2.72 (<0.001)	+3.90 (<0.001)
XGB all − URL+external	+0.24 (0.074)	+0.48 (0.265)

D. Precision at Low Prevalence

Using the mean recall and FPR of XGBoost on all features with grouped folds (94.4% and 4.1%), the implied precision is 54.8% at 5% phishing prevalence, 18.9% at 1% and 2.3% at 0.1%. With URL-only features (85.5% recall, 10.6% FPR) it is 29.8%, 7.5% and 0.8%. With random folds and all features it is 64.1%, 25.5% and 3.3%. These are back-calculations that assume recall and FPR stay constant as prevalence changes; they show that a few percent FPR is costly when phishing is rare.

V. DISCUSSION

ML helps, but the margin depends on the baseline (RQ1). XGBoost on all features beat every baseline in both settings, by more than 30 points against the hand-written heuristic and by 8.7 to 10.6 points against the stump on google_index. Weaker ML configurations did not always win: Logistic Regression on URL-only features (84.9% and 84.2%) fell below the stump and below the depth-3 tree on all features, and URL-only XGBoost was statistically indistinguishable from the stump on grouped folds (+1.1 points, p = 0.62), although the stump has the advantage of an external feature. The claim that ML clearly outperforms static methods therefore holds against weak rules and with external features, not unconditionally. The blacklist is precise but cannot flag unseen domains, which is its defining limitation.

Domain grouping reveals losses that random folds hide (RQ2). The 1.6 to 4.2 point accuracy drops for tree models, the larger fold-to-fold variance and especially the roughly 8 to 18 point losses in TPR@1%FPR for the tree models show that random folds overstate performance on new domains. In earlier runs on this dataset that grouped by exact hostname with a single 80/20 split and five seeds, tree-model drops were 0.9 to 1.8 points; the protocols differ, but the larger drops here are consistent with hostname grouping leaving related URLs (www. and subdomain siblings, hosting platforms) on both sides. Our grouped folds are still not temporal, and Logistic Regression's small drop is a hypothesis for future testing: it may rely on fewer idiosyncratic patterns.

Part of the signal may be sampling artifacts (RQ3). Table I shows large class differences in www. usage and in traffic, rank and index features, which fit the described collection (popularity-ranked legitimate sites versus phishing feeds [10]). A model can score well by separating popular from unpopular or absent sites, which is the sampling-bias pitfall described in [3]; a brand-new legitimate site would also lack rank and traffic, so false positives could rise in practice. We cannot separate artifact from genuine signal without another dataset. External features also require lookups whose services may be slow, changed or retired [8] (web_traffic was obtained from Alexa [10], a service that has since been discontinued), and they were collected at dataset construction time; we did not measure latency or availability.

Model choice. XGBoost beat Random Forest by 0.45 points on random folds (p = 0.021, not significant at the 0.005 threshold) and 0.13 points on grouped folds (p = 0.82), so the two tree ensembles are not clearly separable here, although XGBoost's TPR@1%FPR was higher (83.1% versus 74.7% on grouped folds; not tested).

VI. LIMITATIONS

(1) One dataset, collected in 2020, with the sampling artifacts described above; no temporal or cross-dataset test. (2) Registered-domain grouping is imperfect: it ignores same-campaign pages on different domains, and it groups unrelated sites on shared platforms; grouped test class balance varies (49.8 ± 4.4%). (3) The baselines are limited: the hand-written heuristic is not a published system, the blacklist is simulated, and we did not replay a real blocklist; the stump's split feature was confirmed on the full data but not logged per fold. (4) No hyperparameter tuning, calibration or threshold selection beyond 0.5. (5) Folds overlap, so SDs understate uncertainty; ten comparisons were tested without adjustment and most small differences are not clear. (6) Sentinel values were not treated, and no adversarial evaluation, latency or availability measurement was done. (7) Impurity-based feature importance was not used for conclusions; the ablation is the evidence.

VII. CONCLUSION

On this benchmark, XGBoost with all features outperformed every simple baseline, reaching 97.08% (random folds) and 95.17% (domain-grouped folds) accuracy. The advantage is smaller against a one-split stump (8.7 to 10.6 points) and disappears for URL-only models on grouped folds. Registered-domain grouping reduced accuracy by up to about 4 points and the true-positive rate at 1% FPR by up to about 18 points, and external features carried much of the accuracy while the audit points to dataset sampling artifacts. Future work should add time-split and second datasets, published and real-blocklist baselines, tuning and calibration, adversarial tests and measurements of lookup latency.

Code availability: the experiment script (phishing_experiment_v3.py) and per-fold results (results_v3_raw.csv) are available at https://github.com/AdityaJadhav121/Research_paper_files.

REFERENCES

[1] S. Sheng, B. Wardman, G. Warner, L. Cranor, J. Hong, and C. Zhang, "An empirical analysis of phishing blacklists," in Proc. Conf. Email and Anti-Spam (CEAS), 2009, doi: 10.1184/r1/6469805.

[2] A. M. Butnaru, A. Mylonas, and N. Pitropakis, "Towards lightweight URL-based phishing detection," Future Internet, vol. 13, no. 6, p. 154, 2021, doi: 10.3390/fi13060154.

[3] D. Arp, E. Quiring, F. Pendlebury, A. Warnecke, F. Pierazzi, C. Wressnegger, L. Cavallaro, and K. Rieck, "Dos and don'ts of machine learning in computer security," in Proc. 31st USENIX Security Symp., Boston, MA, USA, Aug. 2022.

[4] R. Zieni, L. Massari, and M. C. Calzarossa, "Phishing or not phishing? A survey on the detection of phishing websites," IEEE Access, vol. 11, pp. 18499–18519, 2023, doi: 10.1109/ACCESS.2023.3247135.

[5] A. Ejaz, A. N. Mian, and S. Manzoor, "Life-long phishing attack detection using continual learning," Sci. Rep., vol. 13, Art. no. 11488, 2023, doi: 10.1038/s41598-023-37552-9.

[6] G. Apruzzese, M. Conti, and Y. Yuan, "SpacePhish: The evasion-space of adversarial attacks against phishing website detectors using machine learning," in Proc. Annu. Comput. Secur. Appl. Conf. (ACSAC), 2022, doi: 10.1145/3564625.3567980.

[7] A. Abuadbba et al., "Towards web phishing detection limitations and mitigation," arXiv:2204.00985, 2022, doi: 10.48550/arXiv.2204.00985.

[8] M. Sánchez-Paniagua, E. Fidalgo, E. Alegre, W. Al-Nabki, and V. González-Castro, "Phishing URL detection: A real-case scenario through login URLs," IEEE Access, vol. 10, pp. 42949–42960, 2022, doi: 10.1109/ACCESS.2022.3168681.

[9] O. N. Jasim, "HEURISTIC-PHISH: A lightweight feature-based framework for malicious URL detection," Muthanna J. Eng. Technol., vol. 14, no. 3, pp. 118–138, 2026, doi: 10.52113/3/eng/mjet/2026-14-03-/117-138.

[10] A. Hannousse and S. Yahiouche, "Towards benchmark datasets for machine learning based website phishing detection: An experimental study," Eng. Appl. Artif. Intell., vol. 104, Art. no. 104347, 2021, doi: 10.1016/j.engappai.2021.104347.

[11] A. Hannousse, "Web page phishing detection," Mendeley Data, 2020, doi: 10.17632/c2gw7fy2j4.

[12] S. Das Guptta, K. T. Shahriar, H. Alqahtani, D. Alsalman, and I. H. Sarker, "Modeling hybrid feature-based phishing websites detection using machine learning techniques," Ann. Data Sci., vol. 11, pp. 217–242, 2022, doi: 10.1007/s40745-022-00379-8.

[13] A. Kulkarni, V. Balachandran, D. M. Divakaran, and T. Das, "From ML to LLM: Evaluating the robustness of phishing web page detection models against adversarial attacks," Digit. Threats: Res. Pract., vol. 6, pp. 1–25, 2025, doi: 10.1145/3737295.

[14] Y. Yuan, G. Apruzzese, and M. Conti, "Beyond the west: Revealing and bridging the gap between Western and Chinese phishing website detection," Comput. Secur., vol. 148, Art. no. 104115, 2024, doi: 10.1016/j.cose.2024.104115.

[15] C. Nadeau and Y. Bengio, "Inference for the generalization error," Mach. Learn., vol. 52, no. 3, pp. 239–281, 2003, doi: 10.1023/A:1024068626366.