

Cross-Dataset Reliability of Random Forest Phishing URL Detection under Heterogeneous Feature Representations: Implications for Cybersecurity Decision-Making

Nazrulazhar Bahaman^{1*}, Sarah Aqilah Arun¹, Erman Hamid¹, Elia Erwani Hassan², Alauddin Maulana Hirzan³

¹Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka, 76100 Durian Tunggal, Melaka, Malaysia

²Faculty of Technology and Electrical Engineering, Universiti Teknikal Malaysia Melaka, 76100 Durian Tunggal, Melaka, Malaysia

³Prodi Teknik Informatika, Universitas Semarang, Tlogosari, Semarang 50196, Indonesia

* Corresponding Author

DOI: <https://doi.org/10.47772/IJRISS.2026.100800999>

Received: 09 September 2026; Accepted: 14 September 2026; Published: 22 September 2026

ABSTRACT

Phishing URL detectors are commonly evaluated using training and test samples drawn from the same dataset, a practice that can overstate reliability when operational data differ in source, collection period, class composition or feature-generation process. This study evaluates Random Forest transferability under within-dataset and heterogeneous cross-dataset conditions and considers the implications of transfer failure for cybersecurity decision-making. A 6,000-sample source dataset was constructed from PhishTank phishing URLs and an Alexa-derived legitimate URL list, while 6,000 records from the UCI Phishing Websites dataset formed an unseen structured target set. A semantic compatibility audit distinguished eight high-equivalence URL-reproducible features from three limited-equivalence measurements and 19 non-transferable attributes. Three Random Forest configurations were evaluated using grouped 10-fold cross-validation, source hold-out testing, registered-domain-grouped validation and one-way target transfer. RF-200 achieved 0.752 accuracy and 0.799 balanced accuracy on the source hold-out set; registered-domain grouping reduced balanced accuracy to 0.779. On the target dataset, the 11-feature model collapsed to an all-phishing prediction pattern, producing 0.633 accuracy and 1.000 phishing recall but 0.000 specificity and 0.500 balanced accuracy. Removing SSLfinal_State and Abnormal_URL produced a 9-feature sensitivity representation with 0.590 balanced accuracy, although one limited-equivalence port variable remained. The results show that aggregate accuracy can conceal complete class-level failure and that feature-generation semantics materially affect transferability. For operational cybersecurity, the findings emphasize independent target evaluation, domain-aware validation, transparent feature definitions, class-specific metrics and cautious use of model outputs in automated blocking or warning decisions.

Keywords: cybersecurity decision-making; cross-dataset evaluation; phishing URL detection; Random Forest; transferability

INTRODUCTION

Phishing remains a persistent cybersecurity threat because deceptive Uniform Resource Locators (URLs) can be generated, modified and abandoned faster than blacklist-centric mechanisms can be updated. Machine-learning detectors address this limitation by learning discriminative patterns from labelled phishing and legitimate samples rather than relying only on previously reported URLs [1], [2]. URL-oriented detection is especially attractive for early-stage screening because lexical and structural indicators can be extracted before a browser renders a page or requests external resources [3]-[5].

Classical machine-learning models, including Random Forest, have repeatedly produced competitive results for phishing websites and malicious URL classification [4]-[8]. Recent work has also explored deep URL representation learning, hybrid neural ensembles, production-oriented validation and feature-selection strategies [3], [13], [19], [21], [22]. These studies demonstrate that strong predictive performance is attainable when training and evaluation data are sufficiently aligned. However, high accuracy on a random split from a single dataset does not establish that learned decision rules will remain reliable when the data source, collection period, class prior or measurement process changes.

This distinction has practical consequences for cybersecurity operations. Automated URL classifiers may influence browser warnings, access-control policies, security-operations triage, email filtering and incident-response priorities. A model that becomes biased toward the phishing class can create excessive false alarms and warning fatigue, whereas a model that becomes conservative can permit malicious URLs to pass undetected. Accordingly, model reliability must be evaluated not only as a predictive problem but also as a decision-support problem in which class-specific errors have different operational consequences.

Phishing datasets are heterogeneous by construction. PhishTank supplies community-reported phishing URLs, whereas the UCI Phishing Websites dataset represents websites through 30 engineered attributes spanning URL syntax, domain registration, Domain Name System (DNS) information, webpage behaviour, hyperlink structure and external reputation information [9], [10]. Newer resources such as LegitPhish similarly emphasize curated and reproducible URL data, yet differences in collection and representation remain unavoidable across benchmark sources [20]. A classifier transferred between such datasets can therefore encounter distribution shift, temporal shift and measurement shift simultaneously.

Cross-dataset evaluation is a substantially more demanding robustness test than within-source validation. Rashid et al. [11] showed that phishing URL detectors can lose generalization when source and target datasets differ and investigated unsupervised domain adaptation as a remedy. Recent robustness protocols also emphasize domain-disjoint partitioning because URL-level overlap can conceal memorization of repeated hosts or infrastructure [25]. Nevertheless, cross-dataset experiments remain difficult to interpret when nominally corresponding features are produced by different measurement procedures.

This study evaluates Random Forest under both within-source and heterogeneous cross-dataset conditions. Dataset A contains 6,000 raw URLs from PhishTank and an Alexa-derived legitimate list; Dataset B contains 6,000 unseen records sampled from the UCI Phishing Websites dataset. A semantic compatibility audit distinguishes URL-reproducible variables from measurements requiring additional network, certificate, WHOIS, webpage or reputation information. The study compares an 11-feature extended representation with a 9-feature reduced sensitivity representation and uses registered-domain grouping to assess whether ordinary URL-level splitting provides optimistic source estimates.

The contribution is fourfold: (1) a source-to-target Random Forest evaluation that preserves target independence; (2) an explicit compatibility audit of the UCI feature space rather than assuming that similarly named variables have equivalent measurement semantics; (3) registered-domain-grouped source validation to reduce avoidable domain overlap; and (4) an operational interpretation of class-level transfer failure using specificity and balanced accuracy in addition to conventional accuracy, precision, recall and F1-score. The objective is not to claim superior classifier accuracy, but to show how evaluation design and representation compatibility affect the trustworthiness of phishing-detection evidence.

LITERATURE REVIEW

Machine Learning and URL-Oriented Phishing Detection

Supervised phishing detection has been studied using lexical URL attributes, domain characteristics, webpage content, hyperlinks and combinations of these signals. Sahingoz et al. [4] used URL and natural-language-processing features with multiple classifiers, while Gupta et al. [5] demonstrated low-cost lexical detection in a real-time setting. Zamir et al. [6] reported strong performance from diverse classical and ensemble classifiers on

the UCI structured dataset. Random Forest remains useful because it captures nonlinear interactions without requiring feature scaling and is comparatively straightforward to inspect through feature importance [8].

Deep models reduce dependence on manual feature design by learning URL representations directly. URLNet learns character- and word-level URL representations [3], while Yang et al. [13] combined convolutional learning with Random Forest. Ghalechyan et al. [19] evaluated neural approaches across open and production-oriented URL sources and highlighted sensitivity to preprocessing and validation data. Model sophistication, however, does not remove the need for independent and distributionally distinct evaluation data.

Recent IJRISS publications also show that cybersecurity and phishing detection are active application areas for machine learning. For example, Wan Abdullah et al. [26] combined lexical and registration features for phishing URL detection, reinforcing the importance of feature-source transparency in operational screening.

Feature Representation and Dataset Dependence

Feature engineering determines the information available to a classifier. Pure URL features such as length, IP-literal usage, special characters, subdomain depth, hyphens and deceptive tokens can be computed from a URL string [4], [5]. In contrast, UCI attributes such as SSL certificate state, favicon behaviour, form actions, domain age, DNS records, web traffic, PageRank and statistical reports require external measurements [10], [16]. The UCI port feature checks whether selected services on a server have preferred open or closed states rather than merely whether an explicit port number appears in the URL. It is therefore not semantically equivalent to a URL-only explicit-port indicator.

Feature selection can improve compactness and reduce noise, but it should not be driven by an unseen target set. Mat Rani et al. [22] reported that feature ranking affected phishing-detection performance across two datasets, while Bahaghighat et al. [21] showed that feature processing and model choice can produce very high within-dataset accuracy. Such results are useful for classifier optimization but do not by themselves quantify transfer under heterogeneous measurement processes.

Cross-Dataset Generalization and Operational Reliability

Many phishing studies still report cross-validation or train-test performance within the same source distribution. Cross-dataset work is less common because datasets differ in available features, collection period, labelling conventions and measurement semantics. Rashid et al. [11] directly demonstrated this generalization problem using domain-adaptation experiments. More recent evaluation work has argued for domain-disjoint partitioning to reduce host or domain leakage and to better approximate deployment on unseen infrastructure [25].

From an operational perspective, the consequences of model transfer failure depend on the direction of the error. A phishing-biased model can increase false blocking and analyst workload, while a legitimate-biased model can increase exposure to malicious sites. This motivates the use of specificity, balanced accuracy and confusion-matrix analysis alongside aggregate accuracy. Table 1 positions the present study against representative prior work.

Table 1. Positioning of related work

Reference	Focus	Evaluation	Relevance / limitation
[4]	Lexical/NLP phishing URLs	10-fold CV	Within-source; no heterogeneous target
[6]	UCI structured phishing	10-fold CV	Single feature-generation process
[11]	Domain adaptation	Cross-domain	Direct generalization focus
[19]	Neural URL detection	Multiple / production data	Shows value of broader validation

[20]	LegitPhish dataset	Dataset resource	Improves reproducibility; not a transfer study
[25]	Robustness protocol	Domain-disjoint external +	Supports leakage-aware evaluation
This study	RF transferability	Source + unseen target	Audits semantic compatibility and class collapse

RESEARCH METHODOLOGY

Experimental Design and Datasets

The experimental design separates source modelling from target evaluation. Dataset A is used for preprocessing, feature construction, cross-validation, model fitting and source testing. Dataset B is reserved as an unseen target and is not used for target-driven feature selection, hyperparameter selection, calibration or retraining. Figure 1 summarizes this workflow.

Dataset A contains 6,000 raw URLs, comprising 3,800 phishing URLs obtained from PhishTank and 2,200 legitimate URLs from an archived Alexa-derived list. Exact duplicates are removed before feature extraction. The source corpus is treated as a fixed experimental snapshot; because the archived Alexa-derived ranking is not version-addressable in the same way as modern research-oriented rankings, this provenance limitation is reported explicitly rather than inferred from later metadata. Future replication should preferably use a versioned research-oriented ranking such as Tranco [24].

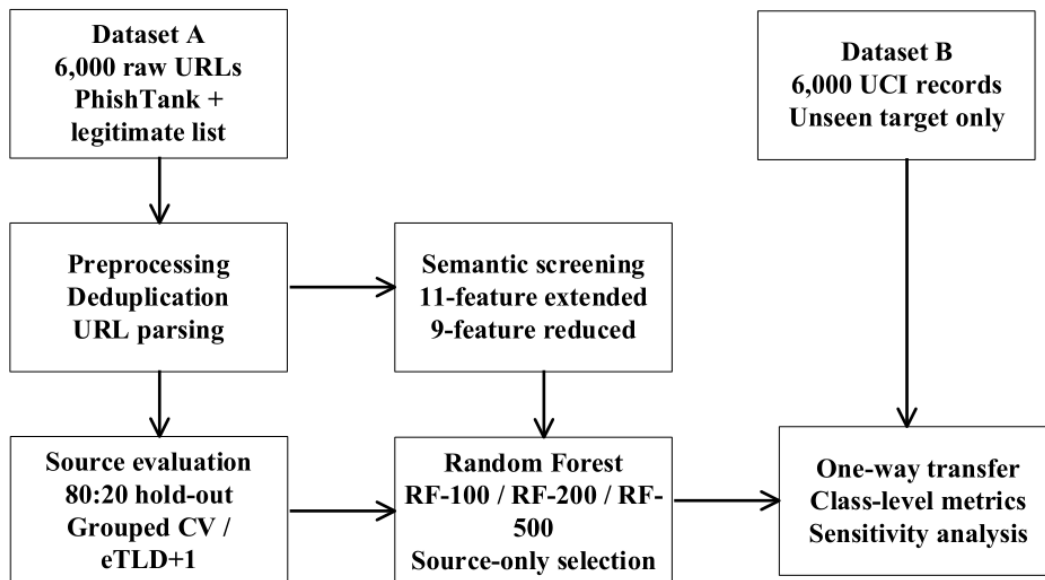


Figure 1. Experimental workflow for source evaluation, semantic screening and one-way cross-dataset testing.

Dataset B is drawn from the UCI Phishing Websites dataset, which contains 11,055 instances and 30 engineered predictors [10]. A reproducible stratified sample of 6,000 UCI records is selected using `random_state = 42` to match the source class counts. Matching prevalence is deliberate because it controls the class prior so that changes in target behaviour are not explained merely by a different phishing proportion. Consequently, target accuracy is not interpreted as an estimate of real-world prevalence-weighted performance. The original UCI result values are recoded so that phishing is the positive class (1) and legitimate is the negative class (0).

Table 2. Dataset profile

Set	Source / format	Samples	Role
A	PhishTank + Alexa-derived list / raw URLs	6,000 (3,800 phishing; 2,200 legitimate)	Training, validation, source tests
B	UCI Phishing Websites / structured features	6,000 (3,800 phishing; 2,200 legitimate)	Unseen target only

Preprocessing and Partitioning

For phishing records, only the URL is retained; submission metadata and identifiers are discarded. Legitimate-domain entries are normalized to a URL form before parsing. Scheme handling is standardized and malformed entries that cannot be parsed are removed. For UCI records, the identifier column is excluded and the class label is recoded. No statistics calculated from Dataset B are used to transform Dataset A.

Two source evaluation strategies are used. First, an 80:20 stratified URL-level split produces a 4,800-sample training subset and a 1,200-sample hold-out subset containing 760 phishing and 440 legitimate URLs. Second, a registered-domain-grouped sensitivity split uses the effective top-level domain plus one (eTLD+1) as the grouping key so that URLs sharing the same registered domain do not appear on both sides of the split. Registered-domain extraction follows Public Suffix List semantics. Group-aware 10-fold cross-validation is performed on the source training data while preserving group separation across folds. The target set remains untouched until final transfer evaluation.

Semantic Feature Compatibility Audit

The semantic compatibility audit distinguishes features that can be reproduced from a raw URL string from variables whose native UCI definitions depend on service state, certificates, webpage content, DNS, WHOIS, traffic, reputation or other external information. Eight variables are classified as high-equivalence URL-reproducible features. Three variables as port, SSLfinal_State and Abnormal_URL are treated as limited-equivalence measurements, while the remaining 19 UCI attributes are excluded from cross-dataset modelling. Figure 2 summarizes this screening process and Table 3 lists the resulting categories.

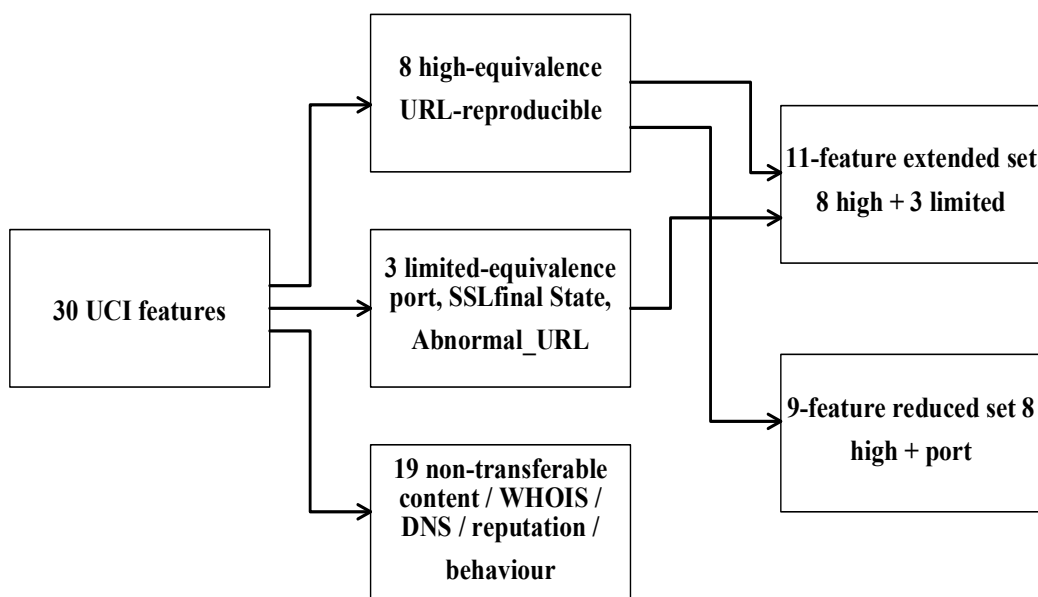


Figure 2. Semantic compatibility screen: eight high-equivalence URL features, three limited-equivalence variables and 19 excluded features.

Table 3. Feature-compatibility screen

Category	Features	Treatment
High (8)	IP address; URL length; shortening service; @; // redirecting; prefix-suffix; subdomain; HTTPS token	Retained in 11- and 9-feature representations
Limited (3)	port; SSLfinal_State; Abnormal_URL	All retained in 11-feature set; port retained in 9-feature sensitivity set
Excluded (19)	WHOIS/DNS; content; hyperlinks; behaviour; reputation; traffic	Removed from cross-dataset modelling

The 11-feature extended representation combines eight high-equivalence features with three limited-equivalence measurements. The 9-feature reduced sensitivity representation removes SSLfinal_State and Abnormal_URL but retains port. Because the native UCI port variable is not fully reproducible from a raw URL string, the 9-feature result is interpreted only as a partial sensitivity analysis of representation mismatch and not as a strict semantic-equivalence experiment. No target labels are used to define either representation.

Table 4. Source-to-UCI feature mapping

UCI feature	Source-side deterministic rule	Semantic status
having_IP_Address	Parse host; flag literal IPv4/IPv6 address.	High
URL_Length	Total URL length using UCI thresholds (<54, 54-75, >75).	High
Shortening_Service	Match host against a fixed, versioned URL-shortener list.	High
having_At_Symbol	Check presence of @ in the URL string.	High
double_slash_redirecting	Check for // beyond the scheme delimiter.	High
Prefix_Suffix	Check for hyphen in the domain/registered-domain component.	High
having_Sub_Domain	Apply UCI dot/subdomain rule to the domain component.	High
HTTPS_token	Check deceptive http/https token in the domain part.	High
port	URL-only explicit-port proxy does not reproduce UCI service-port status.	Limited
SSLfinal_State	HTTPS-scheme proxy cannot reproduce certificate issuer/age.	Limited
Abnormal_URL	Lexical host consistency cannot reproduce WHOIS-based identity check.	Limited

Random Forest Configuration and Validation

Three Random Forest configurations are evaluated using the same source samples and feature set. Random Forest is intentionally held as the classifier family in this study so that the analysis can isolate evaluation-design and feature-representation effects rather than become a broad classifier benchmark. The configurations are labelled by ensemble size rather than Light/Medium/High because the settings do not represent a strictly monotonic increase in structural complexity. All models use Gini impurity, sqrt feature sampling, bootstrap

sampling, no class weighting, and random_state = 42. The configurations are representative operating points rather than an exhaustive hyperparameter search. Model selection is performed exclusively on Dataset A; no target metric participates in configuration selection. Consequently, conclusions are restricted to Random Forest behaviour under the evaluated protocol and are not generalized to other classifier families.

Table 5. Random Forest configurations and grouped 10-fold cross-validation

Model	Trees	Depth	Min split	CV accuracy	CV balanced accuracy
RF-100	100	None	2	0.744 ± 0.018	0.791 ± 0.017
RF-200	200	20	5	0.749 ± 0.016	0.796 ± 0.015
RF-500	500	30	10	0.747 ± 0.017	0.794 ± 0.016

RF-200 is carried forward because it provides the highest mean grouped cross-validation balanced accuracy, although its advantage over the other configurations is small relative to fold variability and is not treated as statistically significant. The selected configuration is then refitted using source training data for the domain-grouped and feature-sensitivity analyses.

Evaluation Metrics, Uncertainty and Reproducibility

Phishing is the positive class. Performance is reported using accuracy, phishing precision, phishing recall, phishing F1-score, legitimate specificity, balanced accuracy and confusion-matrix counts. Specificity is essential because a phishing-biased model can obtain high recall while incorrectly flagging legitimate URLs. Balanced accuracy, defined as the mean of phishing recall and legitimate specificity, exposes one-sided behaviour when class proportions make ordinary accuracy misleading.

For class-conditional rates derived from the reported confusion matrices, 95% Wilson score intervals are additionally reported where useful. On the RF-200 source hold-out set, phishing recall is 0.621 (95% CI 0.586-0.655) and specificity is 0.977 (0.959-0.988). For the 9-feature reduced target result, recall is 0.900 (0.890-0.909) and specificity is 0.280 (0.262-0.299). These intervals quantify sampling uncertainty only and do not remove dependence among related URLs; domain-aware resampling is preferable when group identifiers are available.

All experiments use fixed random_state = 42 where applicable. Preprocessing, source splitting, model fitting, target sampling and evaluation are deterministic under the reported configuration. No target labels or target statistics are used during source preprocessing, model selection or fitting. The source and target roles are fixed before evaluation to preserve one-way transfer independence.

RESULTS AND DISCUSSION

Source Cross-Validation and Hold-Out Performance

The grouped cross-validation results in Table 5 are consistent with the source hold-out behaviour. Table 6 shows that all three configurations operate at high phishing precision and high legitimate specificity but substantially lower phishing recall. On the 1,200-sample URL-level hold-out set, RF-200 produces TN = 430, FP = 10, FN = 288 and TP = 472. This corresponds to 0.752 accuracy, 0.979 precision, 0.621 recall, 0.760 F1-score, 0.977 specificity and 0.799 balanced accuracy.

Table 6. Source evaluation results

Setting	Accuracy	Precision	Recall	F1	Specificity	Balanced accuracy
RF-100 hold-out	0.750	0.977	0.620	0.758	0.975	0.797

RF-200 hold-out	0.752	0.979	0.621	0.760	0.977	0.799
RF-500 hold-out	0.751	0.981	0.618	0.759	0.980	0.799
RF-200 domain-grouped	0.729	0.968	0.592	0.735	0.966	0.779

The source operating point is conservative. False positives are rare, but approximately 38% of phishing URLs are missed. In an operational environment, this asymmetry matters because a low false-alarm rate does not compensate for a large false-negative population. Reporting accuracy alone would obscure that limitation.

Effect of Registered-Domain Grouping

The domain-grouped sensitivity analysis produces $TN = 425$, $FP = 15$, $FN = 310$ and $TP = 450$. Accuracy decreases from 0.752 to 0.729 and balanced accuracy from 0.799 to 0.779, primarily because phishing recall falls from 0.621 to 0.592. The decrease is modest but directionally important: preventing related registered domains from crossing partitions yields a stricter estimate than URL-level random splitting. This supports domain-disjoint or temporal partitioning when robustness, rather than interpolation within a collected corpus, is the evaluation objective [25].

Cross-Dataset Transfer Failure

A much larger behavioural change occurs on the unseen UCI target set. With the 11-feature extended representation, all three Random Forest configurations classify every target record as phishing. For each model, $TN = 0$, $FP = 2,200$, $FN = 0$ and $TP = 3,800$. The resulting accuracy is 0.633 and phishing recall is 1.000, values that could appear favourable if read without class-level context. However, phishing precision is 0.633, legitimate specificity is 0.000 and balanced accuracy is 0.500. Table 7 summarizes these transfer results together with the reduced-feature sensitivity analysis.

Table 7. Cross-dataset transfer and feature sensitivity

Feature / model	Accuracy	Precision	Recall	F1	Specificity	Balanced accuracy
11-feature RF-100	0.633	0.633	1.000	0.776	0.000	0.500
11-feature RF-200	0.633	0.633	1.000	0.776	0.000	0.500
11-feature RF-500	0.633	0.633	1.000	0.776	0.000	0.500
9-feature reduced RF-200	0.673	0.683	0.900	0.777	0.280	0.590

The target result represents complete loss of legitimate-class discrimination: the reported accuracy is driven by the controlled class prior rather than by meaningful two-class separation. Specificity and balanced accuracy therefore provide the appropriate interpretation of this transfer failure. This finding motivates the operational safeguards discussed below rather than any claim that the target accuracy itself is acceptable.

Reduced-Feature Sensitivity Analysis

After `SSLfinal_State` and `Abnormal_URL` are removed and RF-200 is refitted on the source training data, the target confusion matrix becomes $TN = 616$, $FP = 1,584$, $FN = 380$ and $TP = 3,420$. Accuracy increases to 0.673, specificity to 0.280 and balanced accuracy to 0.590, while phishing recall decreases from 1.000 to 0.900. The classifier therefore no longer predicts a single class. The result is consistent with representation mismatch being one contributor to transfer failure but does not establish a causal separation from distributional or temporal differences.

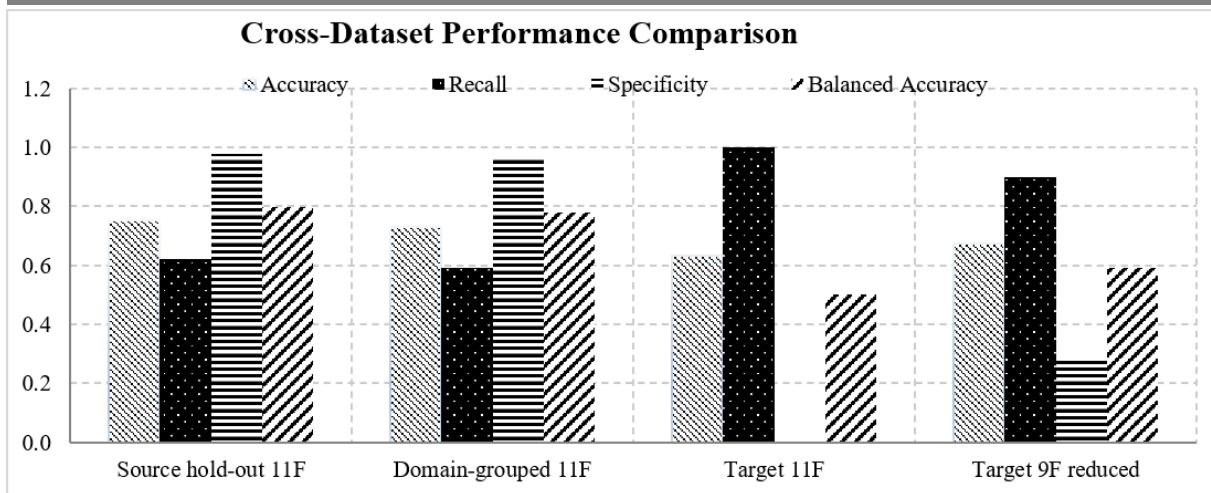


Figure 3. Accuracy, recall, specificity and balanced accuracy across source and target evaluation settings.

Because the UCI port feature depends on service-port status rather than only on an explicit port token in the URL, the 9-feature representation still contains one limited-equivalence variable. The improvement from 0.500 to 0.590 balanced accuracy is therefore interpreted as partial sensitivity evidence. Figure 3 compares the source hold-out, domain-grouped source test, 11-feature target transfer and 9-feature reduced target sensitivity analysis.

Implications for Cybersecurity Decision-Making

The results have implications beyond benchmark reporting. A detector that appears acceptable under source-domain validation can become operationally unsafe after transfer. The observed class collapse would create an excessive false-alarm burden if used for automated blocking and could erode analyst or user trust if deployed as a warning-only system. Deployment decisions should therefore be based on independent target-domain evidence rather than source accuracy alone.

Second, model-governance decisions should consider the semantics of transferred features. A feature with the same name across datasets may reflect different data sources or collection procedures. Treating such variables as interchangeable can create a false impression that model failure is caused only by the classifier rather than by the measurement pipeline. Documentation of feature provenance is therefore part of trustworthy cybersecurity evaluation.

Third, class priors and class-specific metrics should be visible in model-governance reports. Aggregate accuracy can remain superficially plausible even when one class is no longer discriminated. Model validation should therefore report confusion matrices, recall, specificity, and balanced accuracy, with the class prior stated explicitly.

Finally, model acceptance criteria should reflect intended use and the relative cost of false positives and false negatives. For high-impact automated blocking, organizations should require independent-domain validation, define minimum class-specific performance criteria before deployment, and avoid activating a model whose specificity or recall falls below the application's risk tolerance. Where probability outputs are used, calibration and threshold selection should be performed on source validation data rather than on the final target set. Initial deployment should preferably use shadow or advisory mode, with human review for uncertain cases. Post-deployment monitoring should track class-conditional error rates, input drift, alert volume, and changes in feature distributions, with predefined triggers for recalibration, retraining, or model suspension. Table 8 translates the experimental findings into practical governance controls without prescribing universal numerical thresholds, because acceptable error rates depend on deployment context and risk tolerance.

Table 8. Recommended controls for phishing-model acceptance and deployment

Stage	Recommended control	Purpose
Pre-deployment validation	Use domain-disjoint and independent-target evaluation; report class priors and class-specific metrics.	Detect transfer failure before operational use.
Acceptance criteria	Define application-specific minimum recall and specificity according to the relative cost of false negatives and false positives.	Align technical performance with operational risk.
Threshold and calibration	If probability scores are used, calibrate and select thresholds using source validation data only.	Avoid target-set tuning and improve decision consistency.
Initial rollout	Use shadow/advisory mode and human review for uncertain cases before automated blocking.	Limit harm from unobserved deployment shift.
Monitoring	Track drift, false-positive/false-negative rates, alert volume, and feature-distribution changes.	Detect deterioration after deployment.
Corrective action	Predefine triggers for recalibration, retraining, rollback, or model suspension.	Provide governance response when reliability degrades.

Limitations and Future Work

The study is restricted to one-way source-to-target transfer and one classifier family, which limits conclusions about whether the observed reliability problems generalize across alternative algorithms or transfer directions. The three Random Forest configurations are intentionally limited operating points rather than an exhaustive hyperparameter, calibration, or decision-threshold study, and their small cross-validation differences are not interpreted as statistically significant. The source legitimate list is Alexa-derived and may contain historical popularity bias; its archived provenance is less reproducible than versioned modern rankings. The UCI dataset was created in an earlier period and uses an older engineered feature schema [10], so the observed transfer gap may contain a temporal component. The 9-feature reduced set is not fully semantically strict because the port variable retains limited equivalence. In addition, the current analysis does not perform repeated domain-aware partitions or paired statistical comparisons across classifier families. Finally, the controlled target class ratio improves experimental comparability but is not intended to represent operational phishing prevalence.

Future work should extend the protocol in several directions. First, the same evaluation should be repeated with additional classifier families such as Support Vector Machine, gradient-boosted trees, XGBoost/LightGBM, and URL representation-learning models under identical source-target splits. Second, newer and temporally aligned phishing datasets such as LegitPhish [20] and reproducible benign-domain rankings such as Tranco [24] should replace historical sources where possible. Third, a strict transfer experiment should use only genuinely URL-reproducible features, excluding the limited-equivalence port variable, to isolate representation mismatch more cleanly. Bidirectional and multi-source transfer, repeated domain-aware partitions, and paired statistical comparisons should then be used to assess whether performance differences persist across modelling conditions. Finally, probability calibration, cost-sensitive threshold selection, temporal holdouts, explicit domain adaptation, representation learning, and deployment-drift monitoring should be evaluated as mechanisms for improving operational reliability.

CONCLUSION

This study evaluated Random Forest phishing URL detection beyond conventional within-dataset testing using source hold-out evaluation, registered-domain-grouped validation, and one-way cross-dataset transfer under semantically audited feature representations. RF-200 achieved 0.799 balanced accuracy on the source hold-out set and 0.779 under registered-domain grouping, but cross-dataset performance degraded sharply: the 11-feature

representation lost legitimate-class discrimination, with specificity of 0.000 and balanced accuracy of 0.500. Removing SSLfinal_State and Abnormal_URL increased target balanced accuracy to 0.590, although the retained UCI port variable means that this remains a partial sensitivity analysis rather than a strict semantic-equivalence test. The principal contribution is therefore methodological and operational: reliable phishing evaluation should preserve target independence, reduce domain-level overlap, document feature-generation semantics, state class priors, and report class-specific metrics that expose prediction collapse. For deployment, independent-domain validation, risk-based acceptance criteria, calibrated decision thresholds where applicable, and post-deployment drift monitoring are necessary before model outputs are used for automated blocking or warning decisions. Future work should test these conclusions across additional classifiers, newer and temporally aligned datasets, strict URL-reproducible feature sets, bidirectional transfer, repeated domain-aware partitions, and domain-adaptation or representation-learning approaches.

Ethical Considerations

This study used publicly available or previously published cybersecurity datasets and did not involve human participants, animals, clinical data or personally identifiable information. Formal human-subject or animal ethical approval was therefore not applicable.

Data Availability

The phishing component of the source data is derived from PhishTank and the target data are derived from the UCI Phishing Websites dataset. The archived Alexa-derived legitimate list used in the source snapshot has limited version-addressable provenance, as discussed in the limitations. Reproducibility of the reported pipeline requires the same experimental snapshot, preprocessing rules, feature mappings and random seed (42).

Conflict Of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Author Contribution

The authors confirm contribution to the paper as follows: study conception and design, data collection, analysis and interpretation of findings: Sarah Aqilah Arun, Nazrulazhar Bahaman and Erman Hamid; draft manuscript preparation: Nazrulazhar Bahaman, Elia Erwani Hassan and Alauddin Maulana Hirzan. All authors reviewed the findings and approved of the final manuscript.

ACKNOWLEDGEMENT

The authors are grateful to Universiti Teknikal Malaysia Melaka (UTeM) and the Faculty of Information and Communication Technology (FTMK) for equipment and facilities support throughout this research.

REFERENCES

1. Anti-Phishing Working Group. (2026). Phishing activity trends report, 4th quarter 2025. APWG.
2. Sahoo, D., Liu, C., & Hoi, S. C. H. (2017). Malicious URL detection using machine learning: A survey. arXiv:1701.07179. <https://doi.org/10.48550/arXiv.1701.07179>
3. Le, H., Pham, Q., Sahoo, D., & Hoi, S. C. H. (2018). URLNet: Learning a URL representation with deep learning for malicious URL detection. arXiv:1802.03162. <https://doi.org/10.48550/arXiv.1802.03162>
4. Sahingoz, O. K., Buber, E., Demir, O., & Diri, B. (2019). Machine learning based phishing detection from URLs. *Expert Systems with Applications*, 117, 345-357. <https://doi.org/10.1016/j.eswa.2018.09.029>
5. Gupta, B. B., Yadav, K., Razzak, I., Psannis, K. E., Castiglione, A., & Chang, X. (2021). A novel approach for phishing URLs detection using lexical based machine learning in a real-time environment. *Computer Communications*, 175, 47-57. <https://doi.org/10.1016/j.comcom.2021.04.023>

6. Zamir, A., et al. (2020). Phishing web site detection using diverse machine learning algorithms. *The Electronic Library*, 38(1), 65-80. <https://doi.org/10.1108/EL-05-2019-0118>
7. Khan, S. A., Khan, W., & Hussain, A. (2021). Phishing attacks and websites classification using machine learning and multiple datasets: A comparative analysis. *arXiv:2101.02552*. <https://doi.org/10.48550/arXiv.2101.02552>
8. Breiman, L. (2001). Random forests. *Machine Learning*, 45, 5-32. <https://doi.org/10.1023/A:1010933404324>
9. Yasin, A., Fatima, R., Khan, J. A., & Afzal, W. (2024). Behind the Bait: Delving into PhishTank's hidden data. *Data in Brief*, 52, 109959. <https://doi.org/10.1016/j.dib.2023.109959>
10. Mohammad, R., & McCluskey, L. (2012). Phishing Websites. *UCI Machine Learning Repository*. <https://doi.org/10.24432/C51W2X>
11. Rashid, F., Doyle, B., Han, S. C., & Seneviratne, S. (2024). Phishing URL detection generalisation using unsupervised domain adaptation. *Computer Networks*, 245, 110398. <https://doi.org/10.1016/j.comnet.2024.110398>
12. Divakaran, D. M., & Oest, A. (2022). Phishing detection leveraging machine learning and deep learning: A review. *arXiv:2205.07411*. <https://doi.org/10.48550/arXiv.2205.07411>
13. Yang, R., Zheng, K., Wu, B., Wu, C., & Wang, X. (2021). Phishing website detection based on deep convolutional neural network and random forest ensemble learning. *Sensors*, 21(24), 8281. <https://doi.org/10.3390/s21248281>
14. Alani, M. M., & Tawfik, H. (2022). PhishNot: A cloud-based machine-learning approach to phishing URL detection. *Computer Networks*, 218, 109407. <https://doi.org/10.1016/j.comnet.2022.109407>
15. Jain, A. K., & Gupta, B. B. (2019). A machine learning based approach for phishing detection using hyperlinks information. *Journal of Ambient Intelligence and Humanized Computing*, 10(5), 2015-2028. <https://doi.org/10.1007/s12652-018-0798-z>
16. Mohammad, R. M., Thabtah, F., & McCluskey, L. (2014). Predicting phishing websites based on self-structuring neural network. *Neural Computing and Applications*, 25, 443-458. <https://doi.org/10.1007/s00521-013-1490-z>
17. Ma, J., Saul, L. K., Savage, S., & Voelker, G. M. (2009). Beyond blacklists: Learning to detect malicious web sites from suspicious URLs. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1245-1254). <https://doi.org/10.1145/1557019.1557153>
18. Oest, A., et al. (2020). Sunrise to Sunset: Analyzing the end-to-end life cycle and effectiveness of phishing attacks at scale. In *Proceedings of the USENIX Security Symposium* (pp. 361-377).
19. Ghalechyan, H., Israyelyan, E., Arakelyan, A., Hovhannisyan, G., & Davtyan, A. (2024). Phishing URL detection with neural networks: An empirical study. *Scientific Reports*, 14, 25134. <https://doi.org/10.1038/s41598-024-74725-6>
20. Potpelwar, R. S., Kulkarni, U. V., & Waghmare, J. M. (2025). LegitPhish: A large-scale annotated dataset for URL-based phishing detection. *Data in Brief*, 63, 111972. <https://doi.org/10.1016/j.dib.2025.111972>
21. Bahaghighat, M., Ghasemi, M., & Ozen, F. (2023). A high-accuracy phishing website detection method based on machine learning. *Journal of Information Security and Applications*, 77, 103553. <https://doi.org/10.1016/j.jisa.2023.103553>
22. Mat Rani, L., Mohd Foozy, C. F., & Mustafa, S. N. B. (2023). Feature selection to enhance phishing website detection based on URL using machine learning techniques. *Journal of Soft Computing and Data Mining*, 4(1), 30-41.
23. Yeasmin, M. N., Refat, M. A. R., Singh, B. C., Alom, Z., Aung, Z., & Azim, M. (2026). EnLeM: Ensemble learning-based model to detect phishing websites. *EURASIP Journal on Information Security*, 2026, 1. <https://doi.org/10.1186/s13635-025-00219-1>
24. Le Pochat, V., Van Goethem, T., Tajalizadehkhoob, S., Korczyński, M., & Joosen, W. (2019). Tranco: A research-oriented top sites ranking hardened against manipulation. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*. <https://doi.org/10.14722/ndss.2019.23386>
25. Ahamed, T., Kakon, S. C., Farid, F. A., Uddin, J., & Abdul Karim, H. B. (2026). An integrated evaluation protocol for adversarial robustness, generalization and explanation stability in URL-based phishing detection. *Frontiers in Computer Science*, 8, 1834407. <https://doi.org/10.3389/fcomp.2026.1834407>
26. Wan Abdullah, W. A. A., Muslim, Z., Nahar, H., & Motsidi, R. (2026). Enhancing phishing URL detection using a two-level rule-based framework combining lexical and RDAP registration features.



International Journal of Research and Innovation in Social Science, 10(8), 330-347.
<https://doi.org/10.47772/IJRISS.2026.100800025>