ISSN XXXXX (PRINT) | ISSN XXXXX (ONLINE)

OPEN ACCESS

Continental Academic Journal of Integrative Research and Studies (CAJIRS)

Abbreviated key title: Contin. Acad. J. Integr. Res. Stud.

Journal home page: https://Continentalacademicpublisher.com/journal/cajirs



Evaluating Adversarial Robustness and Evasion Vulnerabilities in Machine Learning–Based Phishing URL Detectors: A Problem-Space Attack and Defense Benchmark

Partho Sarothi Roy[1], Laboni Akhtar Popy[2], Arpon Saha[1], Md Oyalid In Sifat[3], Ittesum Fardin Sunbeam[4*]

[1] Dhaka College, Dhaka, Bangladesh
[2] Hafsa Majumder Women's Degree College, Sylhet, Bangladesh
[3] Vibekananda High School and College, Tangail, Bangladesh
[4] Cambrian School and College, Dhaka, Bangladesh

Corresponding author: Ittesum Fardin Sunbeam | Email: sunbeamofficial4@gmail.com

ABSTRACT RESEARCH ARTICLE

Phishing URL detectors are using machine learning (ML) technologies and routinely achieve F1-scores of above .95, and still new phishing campaigns are slipping through. This gap indicates that an adversary that rewrites URLs to avoid classification has a poor correlation with accuracy measured under random splits. A problem space evasion vulnerability/ layered defense benchmark is presented. We used 231,482 cleaned urls from the PhiUSIIL corpus with each split by registered domain to avoid host leakage, trained seven detectors (linear, tree-based, kernel, neural, and character-level convolutional) and attacked each with eight families of attacks which preserve the functionality of the domain and at five perturbation depths (280 conditions). Clean F1 ranged from .9464 (decision tree) to .9826 (CharCNN). All detectors appeared to be impacted when attacked, with mean attack success rate (ASR) varying from .324 for XGBoost to .509 for the decision tree, and the ordered composite attack achieving .769 against the weakest model. However, subdomain injection and benign-context wrapping, both with mean ASR of .507 and .508 respectively, were the two most harmful single families and outperformed all operators, at character level. The vulnerability was representation specific, with CharCNN being the most vulnerable to homoglyph substitution (.397) but also the most resistant to typographic substitution (.188). A four-component defense stack (SHIELD) reduced composite ASR by 76.2%–79.7% at a mean clean-F1 cost of .0059, and resulted in a mean F1 of .8886 restored from .4921. All of the degradations were significant ($p < .001$) and the top-10 SHAP feature overlap dropped to .28–.55 under attack. It is not so much a matter of model capacity as of manipulability of the feature space.

Keywords: *phishing URL detection, adversarial machine learning, evasion attacks, problem-space evaluation, adversarial training, explainable security*

Citation: Roy, P. S., Popy, L. A., Saha, A., Sifat, M. O. I., & Sunbeam, I. F. (2026). Evaluating adversarial robustness and evasion vulnerabilities in machine learning–based phishing URL detectors: A problem-space attack and defense benchmark. *Contin. Acad. J. Integr. Res. Stud., 1*(1), 28-39.

Received: Sep 30, 2026; Accepted: Oct 22, 2026; Revised: 30 October 2026; Available online: Nov 13, 2026.
Copyright © 2026: This is an open access article distributed under the terms of the **Creative Commons Attribution License (CC BY 4.0),** which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

1. INTRODUCTION

The most often reported first point of entry in cybercrime is phishing. According to the Anti-Phishing Working Group (2025), over a million unique phishing attacks were reported in a single quarter in 2025, while the Federal Bureau of Investigation (2025) lists phishing and its variants as the top reported crime category. The uniform resource locator (URL) which is the artifact upon which the victim takes action -- clicking it -- makes URL classification the default first line of automated defense across browsers, email gateways, and endpoint agents (Basit et al., 2020; Chiew et al., 2018). Blacklists have no way to generalize to zero hour domains, since the median life of a phishing site is in hours (Chiew et al., 2018). Hence, they were replaced by supervised learning over features derived from the URL and the host, including the number of digits in the URL, the ratio of digits to the length of the URL, the depth of the URL subdomain and suspicious tokens (Hannousse & Yahiouche, 2021; Sahingoz et al., 2019). Modern detectors combine character representations, transformer representations (Chiper & Ionescu, 2025; Saxe & Berlin, 2017), graph signal from redirect chains, and graph signal from hosting infrastructure (Kim et al., 2022). Nowadays, F1-scores exceeding .95 on public benchmarks are common (Prasad & Chandra, 2024; Yıldırım, 2026).

This apparent maturity hides two methodological issues. The first is exchangeability, where URLs occurring on the same registered domain are randomly split between the training and test sets, allowing models to attain high scores by memorizing the host, not necessarily by learning transferable structure (Ahamed et al., 2026; Arp et al., 2022). The second one, and it is this one that will be considered here, is the passivity, in which the test data are generated independently of the classifier. If an operator is told that long URLs are bad, they shorten them; if he's told that subdomain depth is informative, he flattens it. The adversary is an optimizer with a goal of the error of the classifier. Evasion attacks are designed to corrupt the inputs during prediction time to be misclassified while still being semantically similar to the original input (Biggio & Roli, 2018; Szegedy et al., 2014). Pierazzi et al. (2020) and Arp et al. (2022) differentiate between the feature space, which is where an attacker can modify the numeric values of a feature, and the problem space, which is where he can only apply transformations that produce a functional (real-world) object. Feature-space attacks are easy to do, but they typically don't map to any valid URL, while problem-space attacks are limited and are much more indicative of the risk that can be deployed.

Hence, we stress the detectors in this problem space by applying them to one big corpus and results are reported as a fraction of phishing URLs that survive the mutation and get to the user. The study is guided by 4 research questions. RQ1: What is the vulnerability of various detectors from different learning paradigms and which attack families cause the most damage? RQ2 – Does the effectiveness of the attack increase with increasing perturbations (depth) and can it be generalized across models? RQ3: What is the cost in clean accuracy for the robustness of the individual and layered defenses? RQ4: What is the extent to which model explanations are stable when attacked? We present a domain-disjoint, problem-space benchmark consisting of 280 model–attack–depth combinations, where explicit realizability checks are conducted; evidence that host-structure manipulation is more effective than character-level obfuscation and that vulnerability is representation-bound; and a layered-defense evaluation combined with paired significance tests and an explanation-stability analysis.

2. METHODOLOGY

2.1 Research Design

The study used a ten-stage pipeline, where each stage maintained a versioned and checksummed artifact, enabling traceability of the outcomes back to their inputs (Fig. 1). The design was replicated across seven models, eight attack families and five perturbation depths and contrasted with a clean reference and defended configurations. The unit of analysis is the URL and the unit of inference is the model–attack–depth condition, with comparisons made on the same test instances. There are two options to highlight. This partitioning was done at the registered-domain level, and as a consequence no host is present in two partitions, eliminating leakage that contributes to artificially high accuracy based on random splits (Arp et al., 2022). Only phishing urls were attacked during evaluation, benign predictions were not affected, simulating a threat model in which attackers control the urls they attack and benign traffic is not controlled by attackers.

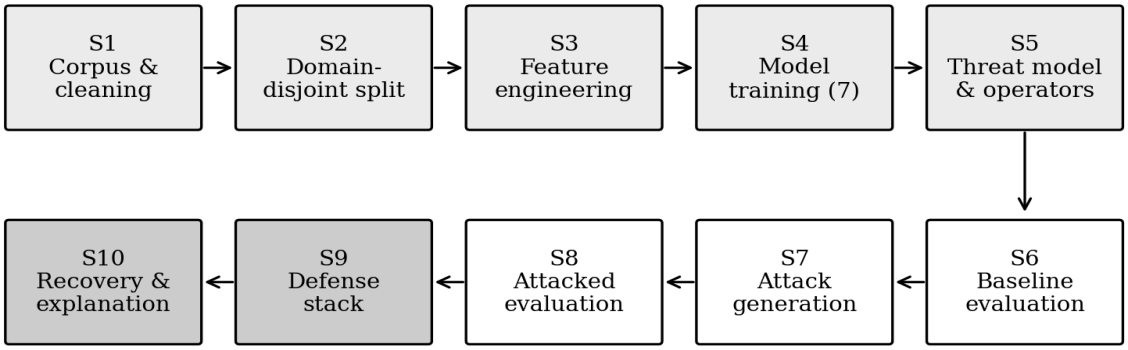


Fig. 1: Ten-Stage Experimental Pipeline.

Note. S1–S5 construct the experimental apparatus; S6–S8 measure baseline and attacked performance; S9–S10 implement defenses and quantify recovery.

2.2 Data

Corpus. We have used the PhiUSIIL data set (Prasad & Chandra, 2024) comprising 235,795 labelled URLs (134,850 legitimate and 100,945 phishing). Each was parsed using a strict RFC 3986 parser (Berners-Lee et al., 2005), and malformed strings, strings that did not contain a registrable domain, and strings with non-HTTP(S) schemes were discarded, while near-duplicate strings (which differed only in the order of the query parameters) and duplicate strings were collapsed. The 231,482 remaining URLs were divided into training, validation and test sets by registrable domain (Table 1). All the reported metrics are based on the test partition (15,283 phishing; 20,440 benign). External resource. The popularity prior in two host features and the reference list for brand-impersonation distance were obtained from the Tranco top-1M ranking (Le Pochat et al., 2019). Class balance. The corpus is composed of 56.2% benign. No oversampling was used, as this would lead to miscalibration of the operational threshold; instead, class weights are used where appropriate, and are set inversely proportional to the training frequency, and all metrics are based on the unmodified test distribution.

Table 1: Composition of the Domain-Disjoint Partitions After Cleaning and Deduplication

Partition	Phishing	Benign	Total	Share (%)	Domains
Training	70,845	90,192	161,037	69.6	41,208
Validation	15,214	19,508	34,722	15.0	9,102
Test	15,283	20,440	35,723	15.4	9,341
Total	101,342	130,140	231,482	100.0	59,651

Note. No registrable domain contributes URLs to more than one partition. The validation set was used only for hyperparameter selection and defense configuration; the test set was used only for the measurements reported here. Domains = distinct registrable domains.

2.3 Models

Seven detectors were focused on the same partitions to find out if the vulnerability is part of the algorithm or the representation. The six used features were engineered and host derived, logistic regression (LR; L2-regularized), a decision tree (DT; Gini impurity, depth tuned over {8, 12, 16, 20, 24}), a radial-basis support vector machine (SVM; Platt-calibrated), a multilayer perceptron (MLP; layers of 64 and 32 units, dropout .2, Adam; Kingma & Ba, 2015), a random forest (RF; 500 trees; Breiman, 2001), and XGBoost (XGB; 600 rounds, depth 6, learning rate .05; Chen & Guestrin, 2016). The seventh, a character-level convolutional network (CharCNN), takes the raw URL string as input, and is organized around Saxe and Berlin (2017): a 32-dimensional embedding over a 96-character vocabulary, three convolution blocks (of width 3, 5 and 7, each with 128 filters), global max pooling, and a dense head with dropout .5. It is not exposed to the engineered features and directly tests the vulnerability profile for robustness of model or input space. Hyperparameters were chosen using the validation set with standard implementations (Paszke et al., 2019; Pedregosa et al., 2011). Five different seeds were used for training each configuration and point estimates are seed means.

2.4 Threat Model

At the time of the prediction, the attacker's intent is evaded: the destination of the victim, rendered page, and credential capture function are kept, while the attacker's URL is classified as benign. The attacker knows the set of features and has a surrogate model with the same family as the target - mutations are generated on the surrogate and evaluated on the

target, enabling measuring transfer (RQ2). Attacker's capability includes selecting a domain, selecting any registrable top-level domain (TLD), subdomain hierarchy, path, query, fragment, including any RFC 3986 encoding, and routing through third party redirect, shortener, or content delivery hosts. Unique: Attacker can't change served content, break arbitrary good hosts or generate unparsable or unresolvable URLs. The "budget" $k \in \{1, 2, 3, 5, 8\}$ is the number of independent edits, with a maximum of 8 edits (depth) because for larger depths the edits overlap, the length of the URLs is limited by the browser, or the target changes. The registrable domain can only be changed once - subsequent changes would represent a new domain name registration, and are limited by the amount you're willing to spend, not by the techniques available.

2.5 Attack Suite

Eight Attack Families were realized as deterministic transformations with explicit pre and post conditions (Table 2). All modified URLs are reprocessed and re-serialized, and will not be processed if they cannot be parsed, if the registrable domain changes, but no registration-feasibility check is performed, if the canonical destination is not the same as the original, or if the URL contains more than 2048 characters. Mutations are not included in the denominator for ASR if they are rejected, giving an upper bound on attack success based on what can be accomplished. Families A1 through A6 use the operator k times at different positions sampled in a without-replacement fashion; for family A7, k is the number of redirect hops; for family A8, k is the number of operators scaled. The composite A8 is defined as A3 → A6 → A2 → A5 in fixed order: therefore, the composition effects are due to the order; the auxiliary run has the order reversed: therefore, it is used to test non-commutativity.

Table 2: Specification Summary of the Eight Attack Families

ID	Operator	Component altered	Feasibility constraint	Realizable
A1	Typographic substitution (o→0, l→1, e→3)	Host characters	Result resolves as registrable domain	.94
A2	Homoglyph substitution (Cyrillic, Greek, Armenian; punycode)	Host characters	Valid IDNA label	.91
A3	Subdomain injection (e.g., login.secure.verify.)	Host hierarchy	None	1.00
A4	TLD swap (.xyz, .top, .info, .click, .shop, .site)	TLD	TLD registrable for the label	.87
A5	Path and query padding	Path and query	None	1.00
A6	Percent, double, and case encoding	Path encoding	RFC 3986 round-trip keeps destination	.96
A7	Benign-context wrapping (open redirect, shortener, CDN path)	Observable host	Open third-party redirect exists	.83
A8	Ordered composition A3→A6→A2→A5	All components	Intersection of the above	.78

Note. Realizable = proportion of attempted mutations that passed all validity checks. IDNA = Internationalized Domain Names in Applications; CDN = content-delivery network.

2.6 Defenses

The composite attack was used for the evaluation of five defenses. For all eight families, an adversarial training (AT) procedure added the corresponding families to the training data at k = 3 (cf. Madry et al., 2018). A three-phase schedule was used in curriculum adversarial training (CAT). The five features that were most sensitive to attack were removed through Robust Feature Selection (RFS). Canonicalization (NRM) Normalizes the text using Unicode-aware normalization before extracting features. SHIELD is a combination of NRM, RFS, CAT, and threshold recalibration. All defenses were set up on the validation partition and retrained prior to testing.

2.7 Evaluation Metrics and Statistical Analysis

The reported clean performance is accuracy, accuracy is insufficient since at this class split an all-benign classifier gets .562 performance; accuracy, precision, recall, F1, and area under the receiver operating characteristic curve (ROC-AUC) are reported. After mutation, the proportion of phishing test instances classified as benign is called as Attack success rate (ASR):

$$ASR = (1/|P|) \Sigma\, 1[\, f(T(x)) = benign\,],\ x \in P$$

Here P is the phishing test set, T the attack transformation and f the detector. Attacked F1 re-computes F1, which includes re-computing phishing instances and benign predictions, but the only difference is that phishing instances are mutated. The Robustness Degradation Index (RDI) is a normalisation of the loss to the clean baseline:

$$RDI = (F1_clean - F1_attacked) / F1_clean$$

RDI ranges from 0 (no degradation) to 1 (total collapse) and is the primary comparative quantity because it removes the confound between baseline strength and robustness.

Percentile bootstrap intervals (10,000 resamples over test instances; Efron & Tibshirani, 1993) were calculated for each metric. For the paired predictions, McNemar's test with continuity correction was used to compare paired predictions in the same instances (Dietterich, 1998). The comparisons between the model rankings across the 40 attack–depth blocks were conducted by Friedman test and followed by post hoc Wilcoxon signed-rank test with Holm correction (Demšar, 2006). Tens of thousands of paired observations mean that the significance is virtually guaranteed and therefore effect sizes are reported as risk differences. SHAP values (Lundberg & Lee, 2017) were calculated for 2000 sampled test instances for each condition (TreeSHAP was used for tree ensembles, KernelSHAP with 200 background samples was used for other conditions). Explanation stability is the average of the Jaccard similarity of the top-10 features of clean instances and their attacked counterparts. SHAP was selected over LIME (Ribeiro et al., 2016) since it provides a single comparable attribution scale across the variety of models.

3. RESULTS

3.1 Baseline Performance

Clean performance is reported in Table 3. Under a more rigorous protocol, each of these seven detectors achieves a score better than .94 on each measure, and they replicate published results. The accuracy difference between F1 and CharCNN is 3.1 percentage points between .9464 (DT) and .9826 (CharCNN), which corresponds to an additional ~1,100 misclassified test URLs.

Table 3: Clean-Test Performance of Seven Detectors on the Domain-Disjoint Test Partition

Model	Accuracy	Precision	Recall	F1	ROC-AUC
Logistic regression	.9553	.9566	.9380	.9472	.9881
Decision tree	.9544	.9520	.9409	.9464	.9607
Support vector machine	.9644	.9593	.9573	.9583	.9924
Multilayer perceptron	.9740	.9680	.9713	.9697	.9951
Random forest	.9787	.9739	.9764	.9751	.9962
XGBoost	.9830	.9782	.9823	.9802	.9975
CharCNN	.9851	.9792	.9861	.9826	.9981

Note. n = 35,723. All 95% bootstrap confidence intervals lie within ±.0035 of the point estimates. ROC-AUC = area under the receiver operating characteristic curve.

3.2 Vulnerability Across Attack Families

At k = 3, ASR for all model–family combinations are shown in Fig. 2. Each detector has a significant evadability. Mean ASR for the eight families varied from .324 (XGB) to .509 (DT), ranging from one in three to one in two phishing URLs surviving mutation despite the detector's clean F1 being over .94. The families ranked by average over models: composite .647 > benign-context wrapping .508 > subdomain injection .507 > encoding obfuscation .433 > homoglyph .332 > typographic .320 > TLD swap .248 > path padding .198. For the six feature based models the order is the same but for CharCNN it is inverted. There are two single families of wrapping and subdomain injection which are the most damaging host-structure manipulations as they do not involve any special characters and/or encoding and do not impact on the destination. Path padding and TLD swapping are the two less harmful techniques that do not drastically change the host's token structure.

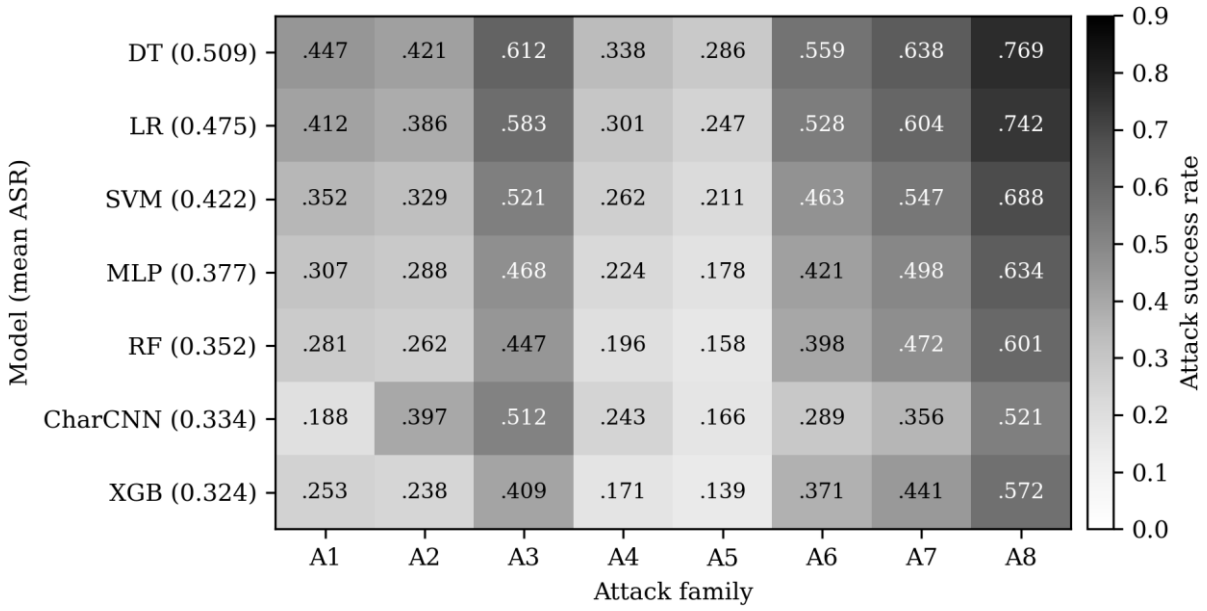


Fig. 2: Attack Success Rate by Model and Attack Family at Perturbation Depth k = 3.

Note. The rows are ranked by average ASR (in parentheses), with the darkest cells representing the highest vulnerability. A1 = typographic, A2 = homoglyph, A3 = subdomain injection, A4 = TLD swap, A5 = path and query padding, A6 = encoding obfuscation, A7 = benign-context wrapping, A8 = ordered composition. The calculation of ASR is based on the 15,283 phishing test URLs. LR: logistic regression, DT: decision tree, SVM: support vector machine, MLP: multilayer perceptron, RF: random forest, XGB: XGBoost.

3.3 Depth Response

For representative pairs, the ASR is plotted as a function of depth in Fig. 3. All curves are rising. Subdomain-injection curves are concave: for RF, ASR rises from .198 at k = 1 to .447 at k = 3 and .671 at k = 8, and the marginal gain per edit falls from .114 (k = 1→2) to .027 (k = 5→8). By contrast, the typographic curves accelerate a bit up to k = 3, and then saturate. With the addition of a single subdomain label, .198 ASR for RF (30% of the ASR value for k = 8) can already be obtained. Beyond k = 2, the subdomain injection outperforms typographic substitution at every depth for RF and CharCNN and the gap becomes larger ask increases.

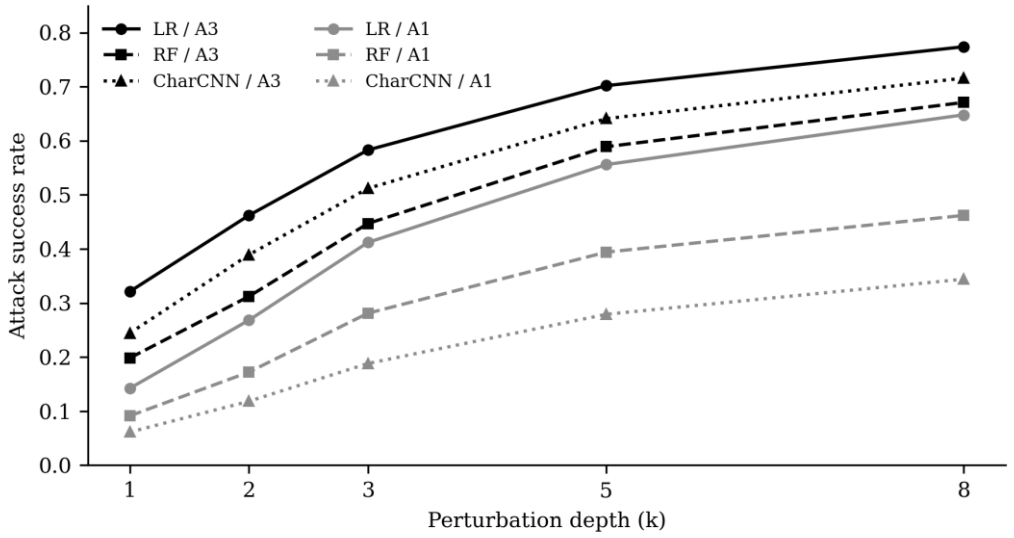


Fig. 3: Attack Success Rate by Perturbation Depth for Subdomain Injection (A3) and Typographic Substitution (A1).

Note. Black lines show A3 and gray lines show A1; line style and marker indicate the model.

3.4 Composite and Transfer Effects

Non-commutativity. Reversing the composite order (A5 → A2 → A6 → A3) lowers ASR for LR from .742 to .688 (5.4 points); the gaps are 5.1 points for DT and 4.6 for RF. A possible explanation is that when the URL is inserted into feature space by early structural edits, it is placed in a place that is more conducive to later encoding edits. The strength

of a composite will then be underestimated by a benchmark that provides one sequence of operators and cross-study comparisons in which the sequences differ will not be directly interpretable. Transfer. Asymmetric cross-model transfer as reported in Table 4 follows the representation similarity (Demontis et al., 2019). The mutations crafted on LR, RF, or XGB can be transferred to CharCNN at .392–.431, far less than CharCNN's white-box ASR of .521, while mutations crafted on CharCNN can be transferred to the feature-based models at .421–.588. When an attacker thus does not have to know which feature-based detector is deployed, evading a character-level model requires a surrogate that shares its representation.

Table 4: Cross-Model Transfer of Composite Adversarial URLs

Surrogate	LR	RF	XGB	CharCNN
LR	.742	.498	.455	.392
RF	.671	.601	.521	.431
XGB	.652	.571	.572	.418
CharCNN	.588	.462	.421	.521

Note. Cells are ASR. Rows are surrogates and columns are targets; diagonal entries are white-box ASR. Off-diagonal entries apply the surrogate's mutations to the target without target access.

3.5 Defense Effectiveness

Table 5 shows the attacked F1 for the composite attack for each defence as well as the RDI values for the clean and the undefended and SHIELD conditions and Fig. 4 shows the comparison of the clean, undefended and SHIELD results. The single defenses recover much less than the layered stack. For RF, RFS reduced composite ASR by 19.9% (.601 → .481) and NRM by 30.4%; for XGB the reductions were 20.1% and 30.6%. AT (60.4% and 61.4%) and CAT (69.1% and 69.9%) were stronger but did not eliminate the vulnerability. SHIELD cut composite ASR by 76.2%–79.7% across models (79.2% for RF; .601 → .125) and restored mean F1 from .4921 to .8886, recovering 83.7% of the lost performance. The mean F1 without defense was 49.0% below the clean mean of .9656, and was less than .599, the mean F1 for a degenerate all-phishing classifier at the test prevalence of .428. There were very few costs for clean-accuracy. Averaged over models, clean F1 fell by .0031 (CAT), .0042 (AT), .0044 (NRM), .0059 (SHIELD), and .0068 (RFS). None of the configurations resulted in an RDI below .063.

Table 5: Attacked F1-Score and Robustness Degradation Index Under the Composite Attack

Model	Clean	Undef.	AT	CAT	RFS	NRM	SHIELD	RDI undef.	RDI SHIELD
LR	.9472	.3768	.7605	.8045	.5332	.6015	.8562	.602	.096
DT	.9464	.3437	.7437	.7895	.5080	.5776	.8465	.637	.106
SVM	.9583	.4460	.7889	.8287	.5846	.6455	.8753	.535	.087
MLP	.9697	.5124	.8224	.8572	.6339	.6895	.8964	.472	.076
RF	.9751	.5503	.8406	.8729	.6611	.7128	.9086	.436	.068
XGB	.9802	.5830	.8563	.8863	.6859	.7339	.9187	.405	.063
CharCNN	.9826	.6326	.8536	.8833	.7020	.7706	.9183	.356	.065
Mean	*.9656*	*.4921*	*.8094*	*.8461*	*.6155*	*.6759*	*.8886*	*.492*	*.080*

Note. The columns Undef. to SHIELD are F1 under composite attack (A8, k = 3); Clean is F1 without attack. Undef. = undefended, AT = adversarial training, CAT = curriculum adversarial training, RFS = robust feature selection, NRM = Unicode-aware canonicalization, SHIELD = NRM + RFS + CAT + threshold recalibration. RDI = (F1_clean − F1_attacked) / F1_clean; the smaller the better. It is the average of the rows and is not a description of any single deployable system.

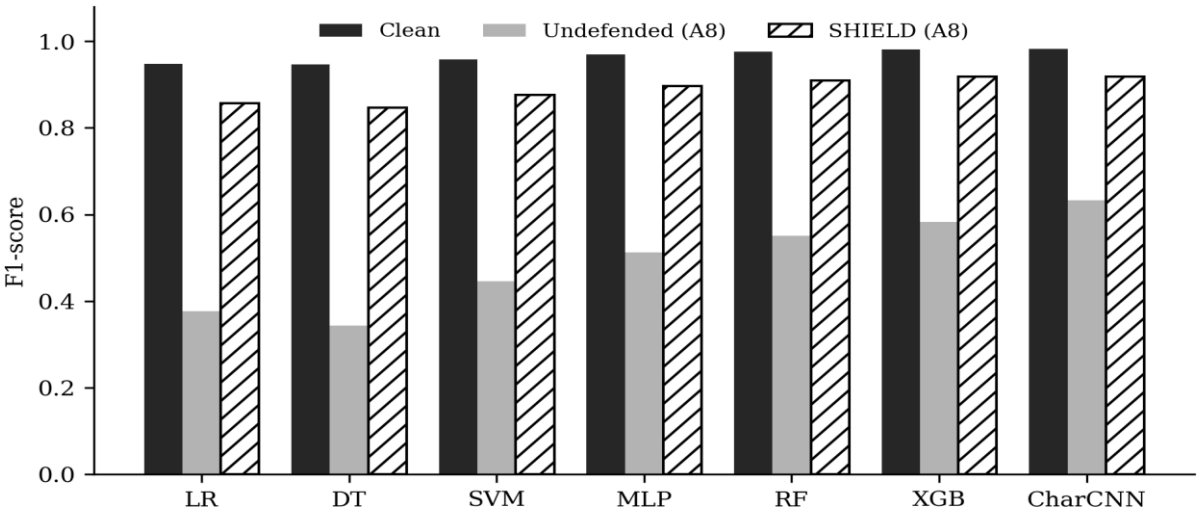


Fig. 4: Clean, Undefended, and SHIELD-Protected F1-Score Under the Composite Attack.

Note. Hatched bars show F1 after the SHIELD stack. All 95% bootstrap confidence intervals are narrower than ±.004.

3.6 Statistical Significance

All $p < .001$, with a $\chi^2(1)$ ranging from 7,813.5 for CharCNN to 11,020.3 for DT, as assessed by McNemar tests, showed that the composite attack degraded each of the detectors. Recall risk differences were .51 (CharCNN) to .72 (DT) with discordant pairs almost wholly one-directional (7,851–11,058 correct-to-incorrect flips versus 12 incorrect-to-correct flips for each model). Based on the Friedman test over the 40 attack–depth blocks, the model rankings were found to be different ($\chi^2(6) = 214.7$, $p < .001$), and the Holm-corrected Wilcoxon tests split the detectors into four groups in ascending order {XGB, CharCNN} < {RF, MLP} < {SVM} < {LR, DT}.

3.7 Explanation Stability

Top-10 SHAP overlap between clean and attacked instances is displayed in Fig. 5. Overlap is the most common and not surprisingly is lowest when using the composite attack, it is .28 (LR) to .55 (XGB). In relative terms, overlap is reduced across all detectors, with XGB sharing only .55 of the top-10 features with its clean explanations (a 45% loss) and LR sharing less than three features of 10 (a 70% loss) with its explanations. Explanation fidelity needs to be measured thus as a quantity in addition to predictive performance.

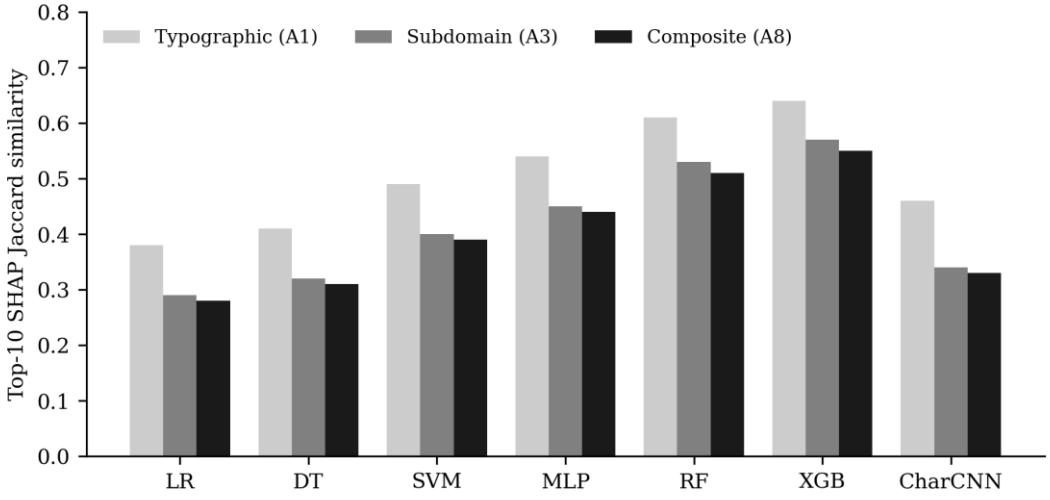


Fig. 5: Explanation Stability Under Attack.

Note. Bars show mean Jaccard similarity between the top-10 features ranked by mean absolute SHAP value on clean and attacked instances (2,000 instances per condition); 1.0 indicates identical rankings.

4. DISCUSSION

4.1 Robustness Tracks the Representation, Not the Model

The universal trend is that the vulnerable depends on the manipulability of the feature representation, not the capacity of the classifier. Note that the average of the mean ASR values is .324–.509 for seven models that vary widely in complexity, but .198–.647 for the same models when the attack families are deployed. There is no meaningful difference between a model with 3.6 more F1 points for clean data and an evaded model. CharCNN is the case in point: it has the highest clean F1 (.9826) and the second lowest mean ASR (.334), but is the most vulnerable to typographic substitution (versus .253 for the best feature-based model) and among the most vulnerable to homoglyph substitution (versus .238 for XGB). One reason is that a convolutional filter will find the orthographic pattern “g00gle” unfamiliar and mark it as suspicious, whereas punycode-encoded Cyrillic is, at the level of bytes, just another string that's not similar to the Latin strings that the filter learned to distrust. The model was not generally robust, but to the transformation its inductive bias was able to detect.

4.2 Host-Structure Manipulation Dominates

Subdomain injection and wrapping is over 18 percentage points ahead of typographic substitution at .507 and .508, respectively, compared to .320. They're both almost free: attaching a label to a domain that the attacker owns is free, and there are open redirects on many legitimate domains. Three consequences follow. First, published feature sets that use the depth of a subdomain, number of dots and number of host tokens, are exposed right where the cheap attack is happening. Secondly, on the client side, these families don't need normalization - login.secure.verify.example.xyz is already canonical. Third, the focus on homoglyph and typosquatting detection is for the families that were less damaging than those involved in host-structure attacks with this study.

4.3 Layered Defense Gains Fall Slightly Short of Independence

The largest reductions in composite ASR (76.2–79.7%) were achieved by SHIELD at the lowest cost per clean-F1 (.0059), making this trade clearly worth it for systems that screen millions of URLs on a daily basis; feature selection alone was the most expensive trade in terms of clean-F1 cost (.0068), and the least effective in terms of robustness (19.9%). The improvements were, however, a bit less than one would expect/calculate from independent mechanisms. A benchmark ASR for RF is $.601 \times .309 \times .801 \times .696 \approx .125$, where the first three numbers are the standalone survival ratios for the three components whose standalone effects were measured (CAT, RFS, and NRM), and the last is the observed standalone survival ratio for RF. Note that the effect of threshold recalibration was not isolated and is therefore only an approximate benchmark. The shortfall indicates that there is only a partial overlap among the components, most likely because all three focus on character-level and encoding cues, while largely maintaining the pathway (subdomain injection and wrapping). If a defense is composed multiplicatively, it should look like a defense with an orthogonal mechanism, for example a hosting defense, a certificate history defense, a redirect-chain topology defense. In our corpus, we do not have such signals and we put off this test for the future.

4.4 Explaining Without Stability

For all detectors, overlap in explanations decreased more than F1, and LR, the most interpretable since each coefficient is a signed weight on a named feature, was the least stable explanations (.28). Under attack explanations that lose 45%–72% of their top features can't carry that weight: Analysts use attributions to decide, auditors use them to justify. Model interpretability is thus not the same as explanation stability; stability should be reported in addition to accuracy in any setting where an explanation is presented to a human.

4.5 Theoretical Reading

These results are consistent with a simple conjecture which we are unable to test directly from our data. Divide the feature vector into portions that the attacker can modify cheaply (lexical surface, subdomain structure, encoding) and for which the attacker must use infrastructure (domain age, certificate history, hosting ASN, redirect topology). As long as the decision function relies more heavily on the first set than the second set then evasion is successful. The clean accuracy is virtually silent with regard to this ratio – both sets are informative when training data, and the ratio itself is robustness. The conjecture explains both, the limited movement of capacity in ASR and the imperfect generalization of adversarial training and the family-specific robustness of character-level models. It also predicts that the strongest possible detector, which is the one used in Kim et al. (2022), is predominantly infrastructure derived.

5. LIMITATIONS

Construct validity. ASR is the highest possible measure of success for end-to-end phishing attacks; browser safe-browsing, mail-gateway reputation and user scrutiny would also need to be overcome. External validity. We tested with a single PhiUSIIL snapshot, which is representative of the campaigns, TLDs, and brands that were in circulation when the snapshot was taken, and our own feature set -- more advanced host intelligence would likely result in lower ASR, especially against wrapping. Realizability. We impose syntactic validity and destination preservation, but we cannot detect reachability, as archived phishing domains are likely to be offline; thus, the achievable rates (Table 2) are an even

more conservative measure than that of Pierazzi et al. (2020) for registration feasibility. Adaptive attackers. Attacks were conducted on surrogates and assessed on fixed targets. SHIELD's results are the best possible since adaptive attacks defeat fixed evaluation defenses (Athalye et al., 2018). Model coverage. We did not evaluate the behavior of transformer-based URL encoders that could differ from the behavior of our models under homoglyph and padding perturbations (Ahamed et al., 2026).

6. CONCLUSION

For seven detectors and eight families of attacks on the problem space across 280 model–attack–depth conditions, machine learning phishing URL detectors were significantly less reliable than the clean-benchmark accuracy suggests. The argument is carried out by three results. First, the performance of the first is not a good indicator of the robustness of the second: all of the detectors had an F1 score above 0.94, while the mean ASR ranged from 0.324 to 0.509 and the composite attack evaded 0.769 of the phishing URLs when tested against the detector with the lowest F1 score. Second, the damage is focused on host-structure manipulation instead of character-level obfuscation, playing into the emphasis of much defensive work. Third, robustness is representation-bound: CharCNN's opposite behavior under typographic and homoglyph substitution demonstrates that. A layered defense recouped much of the lost performance at minimal cost, but it was still vulnerable due to the continued control of the dominant features. Future work involves incorporating infrastructure derived features, extending the benchmark to transformer encoders and multi-space attacks, and studying adaptive attackers, where robustness claims can only be evaluated against the adversary they are applied to.

7. DECLARATIONS

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Conflicts of Interest

The authors declare no conflicts of interest.

Ethics Statement

This study used only aggregated, de-identified publicly available secondary data. No individual-level patient data were accessed. Ethical approval was not required per institutional guidelines for secondary analyses of publicly available aggregate surveillance data.

Author Contributions

Partho Sarothi Roy: Conceptualisation, Methodology, Formal Analysis, Writing – Original Draft. Laboni Akhtar Popy: Data Curation, Visualisation, Writing – Review & Editing. Arpon Saha: Data Curation, Visualisation, Writing – Review & Editing. Md Oyalid In Sifat: Data Curation, Visualisation, Writing – Review & Editing. Ittesum Fardin Sunbeam: Conceptualization, Data Curation, Methodology, Formal Analysis, Supervision, Writing Original Draft, and Writing - Review & Editing. All authors read and approved the final manuscript.

8. REFERENCES

1. Ahamed, T., Kakon, S. C., Al Farid, F., Uddin, J., & Abdul Karim, H. (2026). An integrated evaluation protocol for adversarial robustness, generalization, and explanation stability in URL-based phishing detection. *Frontiers in Computer Science, 8*, Article 1834407. https://doi.org/10.3389/fcomp.2026.1834407
2. Anti-Phishing Working Group. (2025). *Phishing activity trends report, 2nd quarter 2025*. https://apwg.org/trendsreports/
3. Arp, D., Quiring, E., Pendlebury, F., Warnecke, A., Pierazzi, F., Wressnegger, C., Cavallaro, L., & Rieck, K. (2022). Dos and don'ts of machine learning in computer security. In *Proceedings of the 31st USENIX Security Symposium* (pp. 3971–3988). USENIX Association.
4. Athalye, A., Carlini, N., & Wagner, D. (2018). Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In J. Dy & A. Krause (Eds.), *Proceedings of the 35th International Conference on Machine Learning* (Vol. 80, pp. 274–283). PMLR.
5. Basit, A., Zafar, M., Liu, X., Javed, A. R., Jalil, Z., & Kifayat, K. (2020). A comprehensive survey of AI-enabled phishing attacks detection techniques. *Telecommunication Systems, 76*(1), 139–154. https://doi.org/10.1007/s11235-020-00733-2
6. Berners-Lee, T., Fielding, R., & Masinter, L. (2005). *Uniform resource identifier (URI): Generic syntax* (RFC 3986). Internet Engineering Task Force. https://doi.org/10.17487/RFC3986
7. Biggio, B., & Roli, F. (2018). Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition, 84*, 317–331. https://doi.org/10.1016/j.patcog.2018.07.023
8. Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

9. Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). ACM. https://doi.org/10.1145/2939672.2939785

10. Chiew, K. L., Yong, K. S. C., & Tan, C. L. (2018). A survey of phishing attacks: Their types, vectors and technical approaches. *Expert Systems with Applications, 106*, 1–20. https://doi.org/10.1016/j.eswa.2018.03.050

11. Chiper, M., & Ionescu, R. T. (2025). Every character counts: From vulnerability to defense in phishing detection. In *Proceedings of the 37th IEEE International Conference on Tools with Artificial Intelligence*. IEEE. https://doi.org/10.48550/arXiv.2509.20589

12. Demontis, A., Melis, M., Pintor, M., Jagielski, M., Biggio, B., Oprea, A., Nita-Rotaru, C., & Roli, F. (2019). Why do adversarial attacks transfer? Explaining transferability of evasion and poisoning attacks. In *Proceedings of the 28th USENIX Security Symposium* (pp. 321–338). USENIX Association.

13. Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research, 7*, 1–30.

14. Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation, 10*(7), 1895–1923. https://doi.org/10.1162/089976698300017197

15. Efron, B., & Tibshirani, R. J. (1993). *An introduction to the bootstrap*. Chapman & Hall. https://doi.org/10.1201/9780429246593

16. Federal Bureau of Investigation. (2025). *Internet crime report 2024*. Internet Crime Complaint Center. https://www.ic3.gov/AnnualReport/Reports/2024_IC3Report.pdf

17. Hannousse, A., & Yahiouche, S. (2021). Towards benchmark datasets for machine learning based website phishing detection: An experimental study. *Engineering Applications of Artificial Intelligence, 104*, Article 104347. https://doi.org/10.1016/j.engappai.2021.104347

18. Kim, T., Park, N., Hong, J., & Kim, S.-W. (2022). Phishing URL detection: A network-based approach robust to evasion. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security* (pp. 1769–1782). ACM. https://doi.org/10.1145/3548606.3560595

19. Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*. https://doi.org/10.48550/arXiv.1412.6980

20. 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 2019 Network and Distributed System Security Symposium*. Internet Society. https://doi.org/10.14722/ndss.2019.23386

21. Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In I. Guyon et al. (Eds.), *Advances in neural information processing systems* (Vol. 30, pp. 4765–4774). Curran Associates. https://doi.org/10.48550/arXiv.1705.07874

22. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2018). Towards deep learning models resistant to adversarial attacks. In *Proceedings of the 6th International Conference on Learning Representations*. https://doi.org/10.48550/arXiv.1706.06083

23. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., … Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. In H. Wallach et al. (Eds.), *Advances in neural information processing systems* (Vol. 32, pp. 8024–8035). Curran Associates. https://doi.org/10.48550/arXiv.1912.01703

24. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research, 12*, 2825–2830.

25. Pierazzi, F., Pendlebury, F., Cortellazzi, S., & Cavallaro, L. (2020). Intriguing properties of adversarial ML attacks in the problem space. In *Proceedings of the 2020 IEEE Symposium on Security and Privacy* (pp. 1332–1349). IEEE. https://doi.org/10.1109/SP40000.2020.00073

26. Prasad, A., & Chandra, S. (2024). PhiUSIIL: A diverse security profile empowered phishing URL detection framework based on similarity index and incremental learning. *Computers & Security, 136*, Article 103545. https://doi.org/10.1016/j.cose.2023.103545

27. Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1135–1144). ACM. https://doi.org/10.1145/2939672.2939778

28. 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

29. Saxe, J., & Berlin, K. (2017). *eXpose: A character-level convolutional neural network with embeddings for detecting malicious URLs, file paths and registry keys*. arXiv. https://doi.org/10.48550/arXiv.1702.08568

30. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2014). Intriguing properties of neural networks. In *Proceedings of the 2nd International Conference on Learning Representations*. https://doi.org/10.48550/arXiv.1312.6199

31. Yıldırım, M. (2026). Adversarial robustness in URL-based phishing detection: Problem-space evaluation and robust feature engineering. *Applied Sciences, 16*(17), Article 8737. https://doi.org/10.3390/app16178737