

PhishLens: Calibration-Aware Selective Routing with Verified Local-LLM Evidence Selection for Phishing Website Detection

S. Nandhini^{1*}, M. Saranya², P.M.G. Jegathambal³

DOI:10.31033/IJEMR/16.5.2026.1951

^{1*} S. Nandhini, Student, Department of DS & IT, VISTAS, Pallavaram, Chennai, Tamil Nadu, India.

² M. Saranya, Student, Department of DS & IT, VISTAS, Pallavaram, Chennai, Tamil Nadu, India.

³ P.M.G. Jegathambal, Assistant Professor, Department of CSE, VISTAS, Pallavaram, Chennai, Tamil Nadu, India.

Large language models (LLMs) can explain phishing verdicts in natural language, but using them as the detector adds latency and cost, and their explanations can cite evidence that is absent or wrong. We present PhishLens, a phishing website detector in which frozen classical models set every phishing label and probability, while a small local LLM is used only to select explanatory evidence for uncertain cases. A LightGBM model on 46 URL and static-HTML features, with a 20-feature URL-only Random Forest fallback, produces the label and a calibration-checked probability. Cases whose probability falls inside a frozen uncertainty band are routed to a local Qwen3 (4B) model that may return only ranked identifiers of verified evidence and one approved safety-action code; a deterministic verifier checks every output, and any failed output is replaced by a deterministic explanation. The LLM cannot change the probability or label, and submitted URLs are never visited. Detector features, models, calibration choices, classification thresholds and routing bands were selected on registrable-domain-grouped Phish360 partitions and frozen before the final detector evaluations. The later ID selection protocol was assessed only in a 20-case development validation pilot and did not alter any detector or reported test prediction. On the untouched Phish360 test set, the URL+HTML pathway reached 94.58% accuracy, MCC 0.8869, F1 0.9316, PR-AUC 0.9798 and ECE 0.0298. In a second-look external evaluation on CompPhish v3, it reached 87.64% accuracy, MCC 0.7517, F1 0.8672, PR-AUC 0.9441 and ECE 0.0635. The routing policy would avoid LLM invocation for 80.55% of external cases, although the routed share rose from the 10% validation budget to 19.45%. In a separate 40-case validation pilot, all 40 Qwen outputs passed deterministic verification, with a Wilson 95% confidence interval of 91.2%– 100%. Exploratory feedback did not establish that Qwen explanations were more useful than deterministic explanations. Calibration and routing budgets must therefore be re-checked under dataset shift.

Keywords: Phishing Website Detection, Probability Calibration, Selective Prediction, LLM Explanation, Evidence Verification, Dataset Shift

Corresponding Author

S. Nandhini, Student, Department of DS & IT, VISTAS, Pallavaram, Chennai, Tamil Nadu, India.
Email: nshadica@gmail.com

How to Cite this Article

S. Nandhini, M. Saranya, P.M.G. Jegathambal, PhishLens: Calibration-Aware Selective Routing with Verified Local-LLM Evidence Selection for Phishing Website Detection. Int J Engg Mgmt Res. 2026;16(5):68-79.
Available From
<https://ijemr.vandanapublications.com/index.php/j/article/view/1951>

To Browse



Manuscript Received
2026-09-05

Review Round 1
2026-09-19

Review Round 2

Review Round 3

Accepted
2026-10-10

Conflict of Interest
None

Funding
Nil

Ethical Approval
Yes

Plagiarism X-checker
5.30

Note



© 2026 by S. Nandhini, M. Saranya, P.M.G. Jegathambal and Published by Vandana Publications. This is an Open Access article licensed under a Creative Commons Attribution 4.0 International License <https://creativecommons.org/licenses/by/4.0/> unported [CC BY 4.0].



1. Introduction

Phishing websites imitate trusted services to collect credentials and other sensitive data. Detection research has moved from learned URL classifiers [1], [2] and models that combine URL and page-content features [3], [4] to large language model (LLM) detectors that read a page and explain their decision [5]–[7]. Recent reviews describe the same shift [8], [9].

Using an LLM as the detector raises three practical problems. First, every page needs one or more LLM calls. PhishDebate [10], in which four specialised LLM agents debate under a moderator and a judge, reports 98.2% recall, but its full debate averaged 37.5 s per sample against 4.7 s for a single agent. Second, phishing pages are controlled by attackers and can carry instructions aimed at the model [11], [12]. Third, LLM explanations may state reasons the model did not use or cite evidence that does not exist [13]–[15].

Classical classifiers avoid the first two problems, but their outputs are hard for non-specialists to read, and their scores are often reported without any check that they behave as probabilities. Calibration matters when a score drives a later decision, such as whether an uncertain case should receive a longer explanation [16], [17].

PhishLens takes an ML-first approach. Frozen classical models set every phishing label and probability. A frozen uncertainty band on the model probability decides which cases are routed to a local LLM. The LLM may only rank identifiers (IDs) of evidence that PhishLens computed from the model and choose one approved safety action, a deterministic verifier checks every ID and the action code, and any failure falls back to a deterministic explanation. The submitted URL is never fetched and archived HTML is never executed.

PhishLens does not claim to invent ML phishing detection, LLM explanations, probability calibration or selective routing; each exists in prior work. Its contributions are:

- an implemented system that combines calibration checked ML detection, selective local-LLM invocation, evidence-ID-only selection, deterministic verification and safe fallback, in which the LLM cannot alter the probability or the label;

- a leakage-aware protocol with registrable-domain grouped splits, a global freeze before testing, a onetime internal test and a second-look external test on CompPhish v3 with a domain-disjoint subset;
- measurements of how calibration and the routing budget transfer under dataset shift; and
- small validation pilots of restricted evidence selection with a local 4B model, including a 40-case protocol for ranking verified evidence and choosing an approved safety action.

Section II reviews related work, Section III states the research gap and questions, Sections IV–VII describe the method, architecture, data and experimental design, Sections VIII–IX report results

Related Work

Traditional ML, URL and URL + HTML Detection

Ma et al. [1] showed that lexical and host-based URL features can detect malicious sites without downloading the page. Sahingoz et al. [2] compared several classifiers on URL features and found Random Forest best, and Mohammad et al. [18] built readable rule-based classifiers. CANTINA [19] used page text and search-engine queries, and CANTINA+ [3] combined URL, HTML/DOM and third-party features. Aljofey et al. [4] combined character level URL features with HTML features in a gradient boosted model, and deep models learn directly from the raw URL [20] or HTML [21], [22]. Tree ensembles such as Random Forest [23], XGBoost [24] and LightGBM [25] remain standard, and PhishGuard [26] combines several of them in a multi-layered ensemble with feature selection. Visual and reference-based detectors match screenshots or logos against known brands [27], [28] or interact with the page [29], which requires rendering or visiting it. Most of these studies report accuracy on single-dataset splits [8], [30]; a recent systematisation found that many detectors were rarely tested on diverse benign pages and risk excessive false alarms [31].

A. LLM-Based Detection

ChatPhishDetector [5] crawls a site and prompts an LLM to decide whether it is phishing. Lee et al. [6] use a multimodal LLM to identify the imitated brand and check it against the domain. KnowPhish [32] adds a multimodal brand knowledge graph,

PhishLLM [33] replaces a fixed brand list with LLM knowledge validated by search-engine queries, and PhishAgent [7] combines tools and knowledge sources. Prompted LLMs can trail fine-tuned models on URL detection [34]. PhishLang [35] runs MobileBERT on the user's device to keep browsing data local, and small language models of 1B–70B parameters are a practical local alternative on raw HTML, although they trail proprietary LLMs [36]. LLM detectors can be steered by instructions hidden in the page [11], [12]. In all of these systems the language model sets the label.

B. Multi-Agent and Debate-Based Detection

Multi-agent debate improves factuality and reasoning in general tasks [37]. PhishDebate [10] assigns URL, HTML structure, content-semantics and brand-impersonation agents, which reason over the raw URL, rendered HTML and visible text, and lets a judge decide. Debate has also been applied to phishing email [38]. MultiPhishGuard [39], a preprint, weights five agents with reinforcement learning and adds an explanation agent, and Yadav and Masum [40] decompose email evidence across role-specialised agents with a fixed output schema. These designs make explanations structured, but the LLM agents still make the decision on every input.

C. Explainable Phishing Detection

Post-hoc attribution methods such as LIME [41] and SHAP [42] explain individual predictions, and explainable ML has been used to identify the website features most associated with phishing [43]. EXPLICATE [44], a preprint on email, explains a logistic-regression classifier with LIME and SHAP and has an LLM rewrite the result in plain language. APOLLO [45] uses GPT-4o to classify phishing email and write warnings, SmishX [46] uses LLM agents with external context to detect and explain SMS phishing, and a controlled experiment found LLM-written phishing warnings about as protective as expert-written ones [47]. The closest prior work is the evidence-contract approach of Xu and Luo [48]: an LLM detector sees only admissible page evidence and emits fixed-schema findings whose provenance pointers are checked by deterministic validators. There, however, the LLM is the detector and is trained with a regulariser; in PhishLens a frozen classical model decides and an LLM that is not fine-tuned only selects evidence IDs for uncertain cases.

D. Calibration, Selective Prediction and Routing

The Brier score [49] and the binned expected calibration error (ECE) [16], [50] measure how well scores match observed frequencies. Sigmoid (Platt-style) scaling [51] and isotonic regression [52] recalibrate scores, and their effect differs across learners such as random forests and boosted trees [17]. Binned ECE is biased by the binning scheme and the sample size [53], [54]; a recent survey summarises assessment practice [55]. Chow's reject option [56] and selective classification with a validated threshold [57] trade coverage for risk, learning to defer passes hard cases to an expert [58], and confidence-based cascade deferral can fail under distribution shift [59]. LLM cascades and routers reduce cost by escalating only some queries to a larger model [60]–[62]; in these, the escalated model's answer replaces the cheaper one.

E. Faithfulness, Hallucination and Evidence Verification

Faithfulness, whether an explanation reflects the model's reasoning, differs from plausibility [63]. Erasure metrics [64] and perturbation curves test faithfulness, but removing features without retraining shifts the input distribution [65], and perturbation-based explainers can be fooled [66]. Rudin [67] argues for keeping an interpretable model as the decision maker in high-stakes settings. Chain-of-thought explanations can be unfaithful [13], and LLM self explanations need external checks [68]. Surveys of hallucination [14], [69] separate unsupported content from content that contradicts the given context. Retrieval conditions generation on evidence but does not guarantee faithful output [70]; citation benchmarks [71], sampling consistency checks [72] and self-verification [73] detect or reduce unsupported claims but do not give a deterministic guarantee. Explanations can also increase acceptance of AI advice whether or not it is correct [74].

F. Dataset Shift and Cross-Dataset Generalization

Dataset shift [75], [76] degrades both accuracy and uncertainty estimates, including post-hoc calibration fitted on validation data [77]. Security ML is prone to sampling bias, data snooping and lab-only evaluation [78], [79], and data leakage inflates results across ML-based science [80]. Phishing URL detectors generalize poorly across datasets [81],

feature importance is dataset-dependent [82], and existing phishing datasets show leakage and unrealistic base rates [83]. These studies report accuracy or F1, not calibration or routing behaviour.

Table 1: Comparison with the Closest Word

System	Label set by	LLM role	LLM output check
PhishDebate [10]	LLM judge	Debating agents	Not described
ChatPhish Detector [5]	LLM	Detect, explain	Not described
EXPLICATE [44]	Logistic regression	Narrate LIME/SHAP	Not confirmed
SmishX [46]	LLM agents	Detect, explain	External context
Evidence contracts [48]	LLM	Findings with pointers	Deterministic validators
FrugalGPT, RouteLLM [60], [62]	Escalated LLM	Answer	Scorer or router
PhishLens (this work)	Frozen ML model	Rank evidence IDs, choose action	Deterministic verifier and fallback

“Not described” means no such check was found in the verified record; it does not rule one out.

2. Research Gap and Research Questions

The literature forms a chain. Learned URL and URL+HTML classifiers removed the dependence on blocklists but give no per-decision explanation and rarely report calibration [1], [4], [31]. LLM detectors add readable reasoning but decide every label, call the LLM on every page and are exposed to prompt injection [5], [10], [12]. LLM narration of classical attributions keeps the classical decision but leaves the free text unchecked [13], [44]. Evidence contracts validate LLM evidence pointers deterministically, but the LLM remains the detector [48]. Cascades and routers save cost by escalating the answer, not the explanation [60], [62], and confidence-based routing is fragile under shift [59]. In the work reviewed here we found no website-level system in which a calibration-checked classical model fixes the label, the LLM may only select precomputed evidence IDs for uncertain cases, a deterministic verifier checks every ID with fallback on failure, and calibration and routing are reported on an external dataset. PhishLens addresses this gap with four research questions:

- RQ1: How accurate and how well calibrated is a frozen URL+HTML classifier on an untouched internal test and on a shifted external dataset?
- RQ2: How many LLM calls does a frozen uncertainty band avoid, and does its routing budget transfer under shift?
- RQ3: Can a local Qwen3 4B model restricted to verified evidence ranking and approved-action selection produce outputs that satisfy a deterministic verifier?
- RQ4: What do verification and fallback guarantee, and what do they not guarantee?

3. Proposed Methodology

A. Problem Formulation

An input is a URL u with optional archived static HTML h . If usable HTML is present, the URL+HTML pathway is used; otherwise the URL-only pathway is used. The pathway model f returns a phishing probability $p = f(x)$, and the labels

$$\hat{y} = 1[p \geq \tau], (1)$$

where $\tau = 0.44$ for the URL+HTML pathway and 0.35 for the URL-only pathway. Only the ML model sets p and $\hat{y}$; no later stage can change them.

B. Features

The URL+HTML pathway uses 46 robust URL and HTML features covering lexical, structural, link, form and resource properties; the URL-only pathway uses 20 URL features. HTML is parsed as static text. Scripts, styles, frames, forms and external resources are never executed or loaded, and no DNS, WHOIS, TLS, reputation or other network lookup is made.

C. Models and Calibration

The URL-only pathway uses a Random Forest [23] and the URL+HTML pathway uses LightGBM [25].

Hyperparameters were selected by registrable-domain grouped cross-validation inside the training partition. Calibrators were fitted only on a separate calibration partition, and the calibration method, classification threshold and model family were chosen on the validation partition. Validation selected sigmoid calibration for the Random Forest and no post-hoc calibrator for LightGBM, whose raw probabilities were kept.

D. Selective Routing

A case is routed when its probability lies inside the frozen uncertainty band of its pathway:

$$\text{route}(x) = 1[\ell \leq p \leq u], \quad (2)$$

with $[\ell, u] = [0.07403, 0.80597]$ for URL+HTML and $[0.13905, 0.56095]$ for URL-only. The bands were chosen on validation data to route about 10% and 20% of cases, respectively. Routing chooses only the explanation pathway; cases outside the band receive a deterministic explanation directly.

E. Evidence Ledger

For each eligible feature j , PhishLens replaces the observed value x_j with a baseline b_j taken from the training partition and records the change in probability

$$\Delta_j = f(x) - f(x \text{ with } x_j \leftarrow b_j), \quad (3)$$

Each ledger item has an ID, a fixed factual statement, and a direction: whether it supports or opposes the model's prediction. At most three supporting items and one opposing item are selected. This is a local sensitivity check, not a causal explanation and not a proof of the model's internal reasoning.

F. Verified Evidence Ranking and Action Selection

For a routed case, a local Qwen3 (4B) model [84], served through Ollama [85], receives the verified evidence items with their IDs, values, directions and fixed statements. It returns only a JSON object containing ranked supporting evidence IDs, ranked opposing evidence IDs, one approved safety-action code and the fixed conclusion "uncertain". Qwen cannot change the ML label or probability and does not directly write the displayed explanation: PhishLens renders the fixed statements of the selected IDs and the text of the approved action itself. Qwen never receives raw HTML or visible page text. A routed case with no supporting evidence does not reach the LLM and uses the deterministic fallback. Two earlier protocols, in which Qwen copied evidence sentences exactly (Phase 5E) or returned evidence IDs without an action code (Phase 5F), are reported in Section IX.

G. Deterministic Verification and Fallback

The verifier rejects unknown or duplicate IDs, IDs placed in the wrong direction, the wrong number of supporting or opposing items, an invalid ordering,

an action code that is not on the approved list, any change to the output schema and any conclusion other than "uncertain". If any check fails, the whole LLM output is discarded and the deterministic explanation is shown. A pass means only that the output satisfies this contract; it does not mean that the website is safe, that the prediction is correct or that the explanation is complete.

4. System Architecture

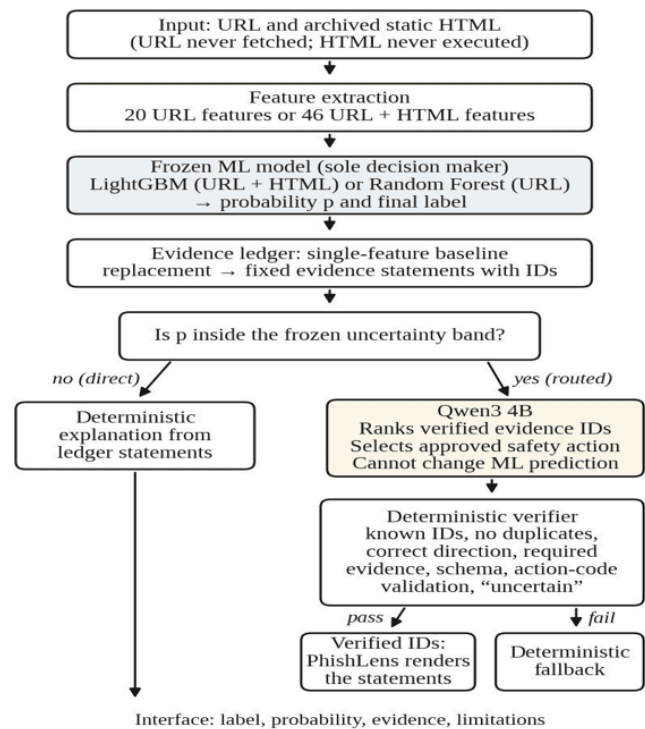


Figure 1: PhishLens processing flow. The frozen ML model is the only component that sets the label and probability; the local LLM is invoked only for routed cases, where it ranks verified evidence IDs and selects an approved safety action.

Fig. 1 shows the flow. The prototype is a FastAPI service with a React interface that runs on the local machine (127.0.0.1) and serves the frozen models. For each input, the interface shows the label, the probability on a 0–1 scale with the classification threshold and the uncertainty band, the pathway used, the evidence statements and the system's limitations. Direct cases show the deterministic explanation; routed cases show either the verified evidence selection or the deterministic fallback.

The Phase 6 freeze manifest records SHA-256 hashes of the feature definitions, models, calibration choices, classification thresholds, routing bands, and the then registered Phase 5A explanation prompt and verifier;

the manifest's own hash is 1ccaa68f5c6364963423b9494690a1725227d96b20a43c600f05b7658cef0903. Both final detector evaluations are write once artefacts tied to this freeze. The Phase 5F and Phase 5G evidence selectors were designed and frozen afterwards, were evaluated only on validation cases, and did not alter the detector or any Phase 6 result. The system never visits submitted URLs, so it cannot observe dynamic JavaScript behaviour, cloaking or short-lived changes to the live page.

5. Datasets and Leakage Prevention

Table 2: Datasets

Dataset	Role	Total	Legit.	Phish.
Phish360 [86], [87]	Development and internal test	10,748	6,416	4,332
Phish360 internal test	One-time internal test	1,587	952	635
CompPhish v3 [88]	External test (second look)	14,932	7,990	6,942
CompPhish domain-disjoint	Stricter external subset	12,573	7,654	4,919

Phish360 counts are before cleaning; CompPhish counts are eligible records after cleaning and overlap removal.

A. Phish360

Phish360 [86] contains 10,748 samples (6,416 legitimate, 4,332 phishing) collected between 2020 and 2023, each with a full URL and archived HTML; it is described in [87]. PhishLens uses only the URL, the HTML and the label. Author annotations such as brand, SSL flag and language are kept for auditing and are never used as features. Exact duplicates of a normalised URL were removed (128 records), leaving 10,620 records, and no conflicting labels occurred.

B. Grouped Splitting and Leakage Checks

Split groups are connected components over shared registrable domains and shared non-trivial identical HTML. Groups were assigned by stratified group folds to training, calibration, validation and internal-test partitions in a ratio of about 60/15/10/15. Near-duplicate pages were found with a 64-bit SimHash over visible text, and any evaluation record whose near-duplicate cluster had a training member was pruned (116 records). Training was blocked unless registrable domains, normalised URLs, non-trivial HTML hashes,

group IDs and near-duplicate clusters were all disjoint across partitions. The internal test (1,587 records) was opened once, after the freeze.

C. CompPhish v3

CompPhish v3 [88] contains 15,358 samples (8,154 legitimate, 7,204 phishing) collected between September 2024 and August 2025 and released under CC BY 4.0. Duplicate URLs and any record overlapping Phish360 by exact or normalised URL, HTML hash or near-duplicate text were excluded, and the frozen evaluation used 14,932 eligible records. A stricter domain-disjoint subset of 12,573 records excludes every registrable domain seen in Phish360. CompPhish was not used for feature design, training, calibration, model selection, threshold or band selection, prompt design or verifier rules. Because a quarantined prePhase-1 version of the project had scored CompPhish, the result is reported as a second-look external evaluation, not a pristine blind test; the frozen models had not scored it before.

D. Known Dataset Construction Cues

In both datasets, legitimate samples are mostly popular homepages with large pages; for example, the median HTML size of legitimate Phish360 pages is 63,760 bytes against 19,178 bytes for phishing pages. Such cues can produce shortcut learning [78], so the external and domain-disjoint tests are used to measure how much performance depends on them.

6. Experimental Design

All detector features, models, calibration choices, classification thresholds and routing bands were frozen before either final evaluation, and no detector component was tuned afterwards. The later Phase 5F and Phase 5G local-LLM studies used validation records only and are not final test evidence. The primary metrics are accuracy, precision, recall, specificity, F1, the Matthews correlation coefficient (MCC), the area under the precision-recall curve (PR-AUC) and ROC-AUC. Calibration is measured with the Brier score [49] and the ECE over $B = 15$ equal-width bins [16], [50]:

$$\text{ECE} = \sum_{b=1}^B (n_b / N) |o_b - \bar{p}_b|, \quad (4)$$

where n_b is the number of cases in bin b , o_b the observed phishing rate and $\bar{p}_b$ the mean predicted probability.

Because binned ECE is an estimate [53], [54], it is reported with the reliability diagram and the Brier score. Routing is measured by the routed fraction, the share of LLM calls avoided relative to calling the LLM for every case (one minus the routed fraction), and error capture, the share of all ML errors that fall inside the band. Confidence intervals use 1,000 registrable-domain cluster-bootstrap resamples (seed 42).

The explanation pilot (Phase 5F) used the same ordered 20 routed development-validation cases as an earlier exact copy protocol (Phase 5E), with qwen3:4b served locally through Ollama. Phase 5E required the model to copy evidence sentences exactly; Phase 5F returns only IDs. The Phase 5F protocol was locked after the Phase 6 detector evaluations and before any Phase 5F Qwen call; it did not modify the frozen detector or routing results. The primary outcome is the verifier pass rate with a Wilson 95% interval; secondary outcomes are failure reasons and median local inference time. This is a small validation pilot of contract compliance, not a measure of detection accuracy. A third protocol, balanced evidence and action selection (Phase 5G), was then evaluated on 40 fresh Phish360 validation cases (10 per pathway and predicted class), excluding the earlier 20 cases. Its prompt and verifier were revised on 20 separate development cases and locked before any evaluation call. In Phase 5G, Qwen and the deterministic baseline display the same number of supporting and opposing facts, so the protocols differ in evidence choice, order and recommended action, not in explanation length.

7. Results

A. Detection Performance

Table 3: Frozen Detection Results

Metric	P360 URL	P360 URL + HTML	CP URL	CP URL + HTML
Accuracy	83.68%	94.58%	82.67%	87.64%
Precision	0.8154	0.9406	0.8429	0.8667
Recall	0.7654	0.9228	0.7711	0.8676
Specificity	0.8845	0.9611	0.8751	0.8841
F1	0.7896	0.9316	0.8054	0.8672
MCC	0.6574	0.8869	0.6519	0.7517
MCC 95% CI	0.56–0.74	0.84–0.92	0.57–0.71	0.70–0.79
PR-AUC	0.8826	0.9798	0.8951	0.9441
ROC-AUC	0.8953	0.9860	0.8974	0.9513
Brier	0.1189	0.0426	0.1314	0.0935
ECE (15 bins)	0.0493	0.0298	0.0742	0.0635

P360 = Phish360 internal test (n = 1,587); CP = CompPhish v3 (n = 14,932). CIs from 1,000 cluster-bootstrap resamples.

Table III answers RQ1. On the untouched Phish360 test, the URL+HTML pathway reached 94.58% accuracy, MCC 0.8869 (95% CI 0.840–0.924), F1 0.9316, PR-AUC 0.9798 and ECE 0.0298, with 586 true positives, 37 false positives, 915 true negatives and 49 false negatives. On CompPhish v3 it reached 87.64% accuracy, MCC 0.7517 (95% CI 0.697–0.790), F1 0.8672, PR-AUC 0.9441 and ECE 0.0635 (6,023 true positives, 926 false positives, 7,064 true negatives, 919 false negatives). On the domain-disjoint subset it reached 86.17% accuracy, MCC 0.7107, F1 0.8249, PR-AUC 0.9169 and ECE 0.0707. The URL+HTML pathway outperformed the URL-only fallback on every set.

B. Calibration Under Shift

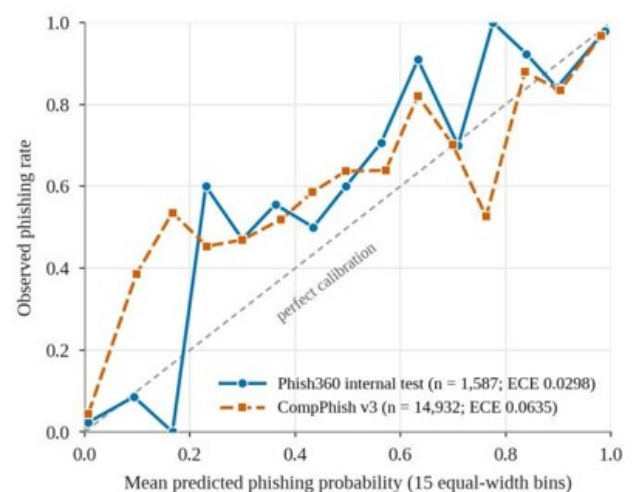


Figure 2: Reliability diagram of the URL+HTML pathway (15 equal-width bins). Most cases lie in the lowest and highest bins, so the middle bins hold few records and vary more.

Fig. 2 shows that calibration transferred less well than ranking. ECE rose from 0.0298 internally to 0.0635 on CompPhish, the Brier score rose from 0.0426 to 0.0935, and the logistic calibration slope fell from 0.83 to 0.64; a slope below one indicates overconfident probabilities. On CompPhish, cases with low predicted probability were phishing more often than predicted.

C. Selective Routing

Table 4: Routing of the Frozen Uncertainty Bands

Set and pathway	Routed	LLM calls avoided	Error capture
P360, URL+HTML	10.33%	89.67%	54.65%
CP, URL+HTML	19.45%	80.55%	64.39%
CP domain-disjoint, URL+HTML	19.72%	80.28%	62.85%
P360, URL-only	20.67%	79.33%	44.40%
CP, URL-only	20.16%	79.84%	48.47%

Error capture = ML errors inside the band / all ML errors. Routing never changes the label or probability.

Table IV answers RQ2. Internally, the URL+HTML band routed 164 of 1,587 cases (10.33%), matching its 10% validation budget, and captured 47 of 86 errors (54.65%). On CompPhish it routed 2,905 of 14,932 cases (19.45%), so 80.55% of cases needed no LLM call, and captured 1,188 of 1,845 errors (64.39%). The routed share nearly doubled under shift, and 35.61% of external errors were confidently wrong and bypassed the band.

8. Ablation and Explanation Evaluation

A. Pathway Ablation: URL-Only Versus URL + HTML

Adding static HTML raised MCC from 0.6574 to 0.8869 internally and from 0.6519 to 0.7517 on CompPhish. The URL+HTML pathway, however, lost more under shift (MCC -0.135) than the URL-only pathway (-0.005), which is consistent with HTML features partly encoding source specific page characteristics such as page size. On the domain-disjoint subset the URL+HTML pathway still led (MCC 0.7107 versus 0.6421).

B. Explanation Protocol Pilots

Table 5: Explanation Protocol Pilots (QWEN3 4B, VALIDATION CASES)

Experiment	Cases	Verifier passes	Fallbacks	Median time
Exact-copy protocol	20	12/20	8	15.04 s
ID-only protocol	20	16/20	4	8.84 s
Balanced evidence and action protocol	40	40/40	0	10.14 s

Phase 5G used fresh validation cases. Therefore, its results are not a phishing detection accuracy result and are not directly paired with the earlier 20-case pilots.

Table V answers RQ3. On the same 20 cases, the ID-only protocol passed verification in 16 cases (80%; Wilson 95% CI 58.4%–91.9%) against 12/20 for exact copying; removing the copying task removed all six quotation failures, and the four remaining failures, which placed an evidence ID in the wrong direction, fell back safely. The balanced evidence and action protocol (Phase 5G) passed verification in all 40 fresh validation cases (Wilson 95% CI 91.2%–100%) with no fallbacks, at a median of 10.14 s per case. Because Phase 5G used different cases, its result is not a paired comparison with the 20-case pilots, and no protocol changes any prediction.

C. Verifier and Perturbation Tests

Synthetic attack suites rejected all 19,527 mechanically generated invalid candidates, and mechanical perturbation tests confirmed that removing a feature's evidence removed it from the explanation, that directions stayed consistent and that explanations updated when the evidence changed. These are tests of the verifier and the ledger, not measurements of real-LLM hallucination rates.

D. Exploratory Usefulness Comparison

An exploratory usefulness comparison did not establish that Qwen explanations were more useful than deterministic explanations. Therefore, PhishLens makes no claim that Qwen improves explanation readability, usefulness or user decision-making. A larger evaluation with independent users is required.

9. Conclusion and Future Work

- A larger, pre-registered evaluation of restricted evidence selection on routed internal and external cases, reporting failures as well as passes.
- Conduct a larger pre-registered evaluation with independent raters and a user study measuring comprehension and decision quality [47], [74].
- Recalibration for new data sources with separately held calibration data, and monitoring of the routed share.

- Constrained decoding to prevent schema errors before verification [89].
- Time-aware evaluation [79] and additional external datasets [83].

PhishLens shows that phishing website detection can keep every label and probability in a frozen, calibration-checked classical model while using a local LLM for bounded evidence selection in uncertain cases. On the untouched Phish360 test, the URL+HTML pathway reached 94.58% accuracy and MCC 0.8869; on the second-look CompPhish v3 evaluation, it reached 87.64% accuracy and MCC 0.7517, with 80.55% of cases needing no LLM call. In a separate 40case validation pilot, all 40 Qwen outputs satisfied the deterministic verifier. However, exploratory feedback did not establish a usefulness advantage over deterministic explanations. PhishLens therefore demonstrates safe and selective local-LLM evidence processing without claiming that the LLM improves detection accuracy or user understanding. The same results show that calibration and routing budgets do not transfer unchanged under dataset shift, so both must be measured whenever the data source changes.

References

- [1] J. Ma, L. K. Saul, S. Savage, & G. M. Voelker. (2009). Beyond blacklists: Learning to detect malicious web sites from suspicious URLs. In *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining (KDD '09)*, pp. 1245–1254. DOI: 10.1145/1557019.1557153.
- [2] O. K. Sahingoz, E. Buber, O. Demir, & B. Diri. (2019). Machine learning based phishing detection from URLs. *Expert Syst. Appl.*, 117, 345–357. DOI: 10.1016/j.eswa.2018.09.029.
- [3] G. Xiang, J. Hong, C. P. Rose, & L. Cranor. (2011). CANTINA+: A feature-rich machine learning framework for detecting phishing web sites. *ACM Trans. Inf. Syst. Secur.*, 14(2). DOI: 10.1145/2019599.2019606.
- [4] A. Aljofey et al. (2022). An effective detection approach for phishing websites using URL and HTML features. *Sci. Rep.*, 12(1), Art. no. 8842. DOI: 10.1038/s41598-022-10841-5.
- [5] T. Koide, H. Nakano, & D. Chiba. (2024). ChatPhishDetector: Detecting phishing sites using large language models. *IEEE Access*, 12, 154381–154400. DOI: 10.1109/ACCESS.2024.3483905.
- [6] J. Lee, P. Lim, B. Hooi, & D. M. Divakaran. (2024). Multimodal large language models for phishing webpage detection and identification. In *Proc. APWG Symp. Electron. Crime Res. (eCrime)*, pp. 1–13. DOI: 10.1109/eCrime66200.2024.00007.
- [7] T. Cao et al. (2025). PhishAgent: A robust multimodal agent for phishing webpage detection. In *Proc. AAAI Conf. Artif. Intell.*, 39(27), pp. 27869–27877. DOI: 10.1609/aaai.v39i27.35003.
- [8] W. Li, S. Manickam, Y.-W. Chong, W. Leng, & P. Nanda. (2024). A state-of-the-art review on phishing website detection techniques. *IEEE Access*, 12, 187976–188012. DOI: 10.1109/ACCESS.2024.3514972.
- [9] D. Sivanewaran, C. T. E. R. Hewage, H. M. K. K. M. B. Herath, R. S. Rathore, V. K. Singh, & W. Jiang. (2026). A systematic literature review of large language models in phishing attack generation and detection. *Array*, 30, Art. no. 100775 DOI: 10.1016/j.array.2026.100775.
- [10] W. Li, S. Manickam, Y.-W. Chong, & S. Karuppayah. (2025). PhishDebate: An LLM-based multi-agent framework for phishing website detection. In *Proc. IEEE Int. Conf. Big Data (BigData)*, pp. 6606–6615. DOI: 10.1109/BigData66926.2025.11401440.
- [11] K. Greshake, S. Abdelnabi, S. Mishra, C. Endres, T. Holz, & M. Fritz. (2023). Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection. In *Proc. 16th ACM Workshop Artif. Intell. Secur. (AISec '23)*, pp. 79–90. DOI: 10.1145/3605764.3623985.
- [12] T. Koide, H. Nakano, & D. Chiba. (2026). *Clouding the mirror: Stealthy prompt injection attacks targeting LLM-based phishing detection*. arXiv:2602.05484. Available at: <https://arxiv.org/abs/2602.05484>
- [13] M. Turpin, J. Michael, E. Perez, & S. Bowman. (2023). Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. In *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 36, pp. 74952–74965. DOI: 10.52202/0752803275.
- [14] L. Huang et al. (2025). A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Trans. Inf. Syst.*, 43(2), 1–55. DOI: 10.1145/3703155.

- [15] S. Kuikel, A. Piplai, & P. Aggarwal. (2025). *Evaluating large language models for phishing detection, self-consistency, faithfulness, and explainability*. arXiv:2506.13746. Available at: <https://arxiv.org/abs/2506.13746>
- [16] C. Guo, G. Pleiss, Y. Sun, & K. Q. Weinberger. (2017). On calibration of modern neural networks. In *Proc. 34th Int. Conf. Mach. Learn. (ICML)*, ser. *PMLR*, 70, pp. 1321–1330. Available at: <https://proceedings.mlr.press/v70/guo17a.html>
- [17] A. Niculescu-Mizil & R. Caruana. (2005). Predicting good probabilities with supervised learning. In *Proc. 22nd Int. Conf. Mach. Learn. (ICML '05)*, pp. 625–632. DOI: 10.1145/1102351.1102430.
- [18] R. M. Mohammad, F. Thabtah, & L. McCluskey. (2014). Intelligent rulebased phishing websites classification. *IET Inf. Secur.*, 8(3), 153–160. DOI: 10.1049/iet-ifs.2013.0202.
- [19] Y. Zhang, J. I. Hong, & L. F. Cranor. (2007). Cantina: A content-based approach to detecting phishing web sites. In *Proc. 16th Int. Conf. World Wide Web (WWW '07)*, pp. 639–648. DOI: 10.1145/1242572.1242659.
- [20] A. S. Bozkir, F. C. Dalgic, & M. Aydos. (2023). GramBeddings: A new neural network for URL based identification of phishing web pages through n-gram embeddings. *Comput. Secur.*, 124, Art. no. 102964. DOI: 10.1016/j.cose.2022.102964.
- [21] C. Opara, B. Wei, & Y. Chen. (2020). HTMLPhish: Enabling phishing web page detection by applying deep learning techniques on HTML analysis. In *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, pp. 1–8. DOI: 10.1109/IJCNN48605.2020.9207707.
- [22] C. Opara, Y. Chen, & B. Wei. (2024). Look before you leap: Detecting phishing web pages by exploiting raw URL and HTML characteristics. *Expert Syst. Appl.*, 236, Art. no. 121183. DOI: 10.1016/j.eswa.2023.121183.
- [23] L. Breiman. (2001). Random forests. *Mach. Learn.*, 45(1), 5–32. DOI: 10.1023/A:1010933404324.
- [24] T. Chen, & C. Guestrin. (2016). XGBoost: A scalable tree boosting system. In *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining (KDD '16)*, pp. 785–794. DOI: 10.1145/2939672.2939785.
- [25] G. Ke et al. (2017). LightGBM: A highly efficient gradient boosting decision tree. In *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 30. Available at: https://proceedings.neurips.cc/paper_files/paper/2017/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html
- [26] M. S. I. Ovi, M. H. Rahman, & M. A. Hossain. (2024). PhishGuard: A multi-layered ensemble model for optimal phishing website detection. In *Proc. 6th Int. Conf. Sustain. Technol. Ind. 5.0 (STI)*. DOI: 10.1109/STI64222.2024.10951075.
- [27] S. Abdelnabi, K. Krombholz, & M. Fritz. (2020). VisualPhishNet: Zero-day phishing website detection by visual similarity. In *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS '20)*, pp. 1681–1698. DOI: 10.1145/3372297.3417233.
- [28] Y. Lin et al. (2021). Phishpedia: A hybrid deep learning based approach to visually identify phishing webpages. In *Proc. 30th USENIX Secur. Symp.*, pp. 3793–3810. Available at: <https://www.usenix.org/conference/usenixsecurity21/presentation/lin>
- [29] R. Liu, Y. Lin, X. Yang, S. H. Ng, D. M. Divakaran, & J. S. Dong. (2022). Inferring phishing intention via webpage appearance and dynamics: A deep vision based approach. In *Proc. 31st USENIX Secur. Symp.*, pp. 1633–1650. Available at: <https://www.usenix.org/conference/usenixsecurity22/presentation/liuruofan>
- [30] L. Tang, & Q. H. Mahmoud. (2021). A survey of machine learning-based solutions for phishing website detection. *Mach. Learn. Knowl. Extr.*, 3(3), 672–694. DOI: 10.3390/make3030034.
- [31] M. Alam et al., (2026). SoK: PHILTER: Uncovering security and functional gaps in AI-based phishing website detection literature via an LLM-based reasoning framework. In *Proc. 35th USENIX Secur. Symp.*, pp. 4981–5000. Available at: <https://www.usenix.org/conference/usenixsecurity26/presentation/alam>
- [32] Y. Li et al., (2024). KnowPhish: Large language models meet multimodal knowledge graphs for enhancing reference-based phishing detection. In *Proc. 33rd USENIX Secur. Symp.*, pp. 793–810. Available: <https://www.usenix.org/conference/usenixsecurity24/presentation/liyuexin>

- [33] R. Liu, Y. Lin, X. Teoh, G. Liu, Z. Huang, & J. S. Dong. (2024). Less defined knowledge and more true alarms: Reference-based phishing detection without a pre-defined reference list. In *Proc. 33rd USENIX Secur. Symp.*, pp. 523–540. Available: <https://www.usenix.org/conference/usenixsecurity24/presentation/liuruofan>
- [34] F. Trad, & A. Chehab. (2024). Prompt engineering or fine-tuning? A case study on phishing detection with large language models. *Mach. Learn. Knowl. Extr.*, 6(1), 367–384. DOI: 10.3390/make6010018.
- [35] S. Saha Roy, & S. Nilizadeh. (2026). PhishLang: A real-time, fully clientside phishing detection framework using MobileBERT. In *Proc. Netw. Distrib. Syst. Secur. (NDSS) Symp.* DOI: 10.14722/ndss.2026.231037.
- [36] G. Goldenits, P. König, S. Raubitzek, & A. Ekelhart. (2026). Small language models for phishing website detection: Cost, performance, and privacy trade-offs. *J. Cybersecur. Priv.*, 6(2), Art. no. 48. DOI: 10.3390/jcp6020048.
- [37] Y. Du, S. Li, A. Torralba, J. B. Tenenbaum, & I. Mordatch. (2024). Improving factuality and reasoning in language models through multiagent debate. In *Proc. 41st Int. Conf. Mach. Learn. (ICML)*, ser. PMLR, 235, pp. 11733–11763. Available: <https://proceedings.mlr.press/v235/du24e.html>
- [38] N. T. V. Nguyen, F. D. Childress, & Y. Yin. (2025). Debate-driven multiagent LLMs for phishing email detection. In *Proc. 13th Int. Symp. Digit. Forensics Secur. (ISDFS)*, pp. 1–5. DOI: 10.1109/ISDFS65363.2025.11012014.
- [39] Y. Xue, E. Spero, M. W. Woo, W. Gao, & G. Russello. (2025). *MultiPhishGuard: An explainable and adaptive multi-agent LLM system for phishing email detection*. arXiv:2505.23803. Available: <https://arxiv.org/abs/2505.23803>
- [40] T. Yadav, & M. Masum. (2026). Explainable multi-agent LLM framework for phishing email detection via role-specialized evidence decomposition. *Electronics*, 15(12), Art. no. 2606. DOI: 10.3390/electronics15122606.
- [41] M. T. Ribeiro, S. Singh, & C. Guestrin. (2016). ‘Why should I trust you?’: Explaining the predictions of any classifier. In *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining (KDD ‘16)*, pp. 1135–1144. DOI: 10.1145/2939672.2939778.
- [42] S. M. Lundberg, & S.-I. Lee. (2017). “A unified approach to interpreting model predictions. In *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 30, pp. 4765–4774. Available: https://proceedings.neurips.cc/paper_files/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html
- [43] M. C. Calzarossa, P. Giudici, & R. Zieni. (2024). Explainable machine learning for phishing feature detection. *Qual. Reliab. Eng. Int.*, 40(1), 362–373. DOI: 10.1002/qre.3411.
- [44] B. Lim, R. Huerta, A. Sotelo, A. Quintela, & P. Kumar. (2025). *EXPLICATE: Enhancing phishing detection through explainable AI and LLM-powered interpretability*. arXiv:2503.20796. DOI: 10.48550/arXiv.2503.20796.
- [45] G. Desolda, F. Greco, & L. Viganò. (2025). APOLLO: A GPT-based tool to detect phishing emails and generate explanations that warn users. *Proc. ACM Hum.-Comput. Interact.*, 9(4), 1–33. DOI: 10.1145/3733049.
- [46] Y. Wang et al. (2025). Can you walk me through it? Explainable SMS phishing detection using LLM-based agents. In *Proc. 21st Symp. Usable Privacy Secur. (SOUPS)*, pp. 37–56. Available: <https://www.usenix.org/conference/soups2025/presentation/wang>
- [47] F. M. Cau, G. Desolda, F. Greco, L. D. Spano, & L. Viganò. (2025). *Can large language models automate phishing warning explanations? A controlled experiment on effectiveness and user perception*. arXiv:2507.07916. DOI: 10.48550/arXiv.2507.07916.
- [48] Y. Xu, & X. Luo. (2026). Auditable and robust LLM-based phishing detection via provenance-guided evidence contracts. *Comput. Mater. Continua*. DOI: 10.32604/cmc.2026.085276.
- [49] G. W. Brier. (1950). Verification of forecasts expressed in terms of probability. *Mon. Weather Rev.*, 78(1), 1–3.

[50] M. P. Naeini, G. Cooper, & M. Hauskrecht. (2015). Obtaining well calibrated probabilities using Bayesian binning. In *Proc. AAAI Conf. Artif. Intell.*, 29(1). DOI: 10.1609/aaai.v29i1.9602.

[51] H. -T. Lin, C.-J. Lin, & R. C. Weng. (2007). A note on Platt's probabilistic outputs for support vector machines. *Mach. Learn.*, 68(3), 267–276. DOI: 10.1007/s10994-007-5018-6.

Disclaimer / Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of Journals and/or the editor(s). Journals and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.