International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 11 | Nov - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Explainable Artificial Intelligence for Phishing Detection: A Review of Techniques, Challenges, and Future Directions

Mohd Kaif

Department of Computer Science and Engineering, Kamla Nehru Institute of Physical and Social Sciences (KNIPSS), Faridipur, Sultanpur, Uttar Pradesh 228001, India

Email: kaifsainul27@gmail.com

Abstract— Phishing remains a persistent cybersecurity threat because attackers continually modify URLs, websites, messages, and delivery channels to imitate trusted entities and manipulate users. Machine learning (ML) and deep learning (DL) have expanded automated phishing detection beyond static blacklists and hand-written rules, but high predictive performance alone does not establish that a detector is understandable, dependable, or suitable for operational use. Explainable artificial intelligence (XAI) methods, particularly Local Interpretable Model-agnostic Explanations (LIME) and SHapley Additive exPlanations (SHAP), are increasingly used to expose the features influencing phishing predictions. This review examines phishing detection and explainability, covering URL- and website-based detection, conventional ML and DL models, post-hoc feature attribution, explanation assessment, cross-dataset generalization, and adversarial robustness. A structured narrative review approach was used to identify foundational and recent literature available up to 31 October 2026. The reviewed evidence indicates that explanation generation is not equivalent to explanation validation: an attribution plot may be readable while remaining unstable under small input changes, unfaithful to model behaviour, or misleading when data distributions shift. Recent comparative work reports differences between SHAP and LIME, while cross-dataset studies emphasize evaluation beyond a single benchmark. The review synthesizes an evaluation framework built around predictive validity, explanation fidelity, stability, generalization, adversarial robustness, human usefulness, and computational cost. It concludes that explanation quality should be treated as an evaluated security property rather than an optional visualization.

Keywords— Phishing Detection, Explainable Artificial Intelligence, Machine Learning, SHAP, LIME, Cybersecurity

I. INTRODUCTION

Phishing is a form of social engineering in which an attacker impersonates a trusted organization or individual to persuade a target to disclose credentials, financial information, personal data, or perform an unsafe action. It can be delivered through email, text messages, social networking platforms, QR codes, malicious advertisements, and deceptive websites. The attacker's advantage is the ability to change the content and infrastructure of an attack quickly. A phishing URL may use a misleading subdomain, a look-alike domain, an unusual top-level domain, URL shortening, redirects, or a compromised legitimate website. Consequently, detection based only on previously reported indicators can fail when a campaign uses new infrastructure.

Traditional defenses include blocklists, domain reputation, manually designed heuristics, browser warnings, and user education. Blocklists are effective when a malicious resource has already been discovered and shared, but they are inherently reactive. Heuristic methods identify suspicious structures without prior knowledge, yet rigid rules may produce false positives and can be evaded by attackers who understand the rules. Machine learning provides a complementary approach: a classifier can learn patterns from labelled examples and combine signals from URL lexical properties, domain information, HTML structure, page content, and hosting context [1], [2].

The adoption of ML and DL introduces another problem. A classifier may output "phishing" with high confidence without providing a useful account of why it reached that decision. In cybersecurity operations, an unexplained score can be difficult to validate, challenge, or translate into a response. Analysts may need to know whether the prediction was driven by a suspicious subdomain, an abnormal redirect chain, a form requesting credentials, or a domain-reputation signal. Users also need explanations that communicate risk without implying that a model's prediction is infallible.

Explainable artificial intelligence (XAI) seeks to make model behaviour more understandable. LIME approximates a model locally around an instance to explain an individual prediction [5]. SHAP uses Shapley-value-based feature attribution to assign contributions to features [6]. Other approaches include feature permutation, partial-dependence analysis, counterfactual explanations, interpretable-by-design models, attention visualization, and gradient-based attribution. These methods differ in assumptions, scope, computational demands, and the type of evidence they provide.

This paper reviews how XAI is used in phishing detection and how the quality of explanations should be assessed. It does not propose a new classifier, conduct new experiments, or claim original benchmark results. Its contribution is a critical synthesis of detection approaches and explainability methods, together with an evaluation framework linking predictive performance to explanation fidelity, stability, dataset shift, adversarial manipulation, human usefulness, and operational cost.

The review is guided by six questions: (1) Which ML and DL approaches are commonly used for phishing detection? (2) Which feature families are used by URL- and website-based detectors? (3) How are LIME, SHAP, and other XAI techniques integrated into phishing detection? (4) What is the difference between producing an explanation and validating its quality? (5) How do dataset variation and adversarial manipulation affect prediction and explanation reliability? (6) Which research directions can improve practical deployment?

II. BACKGROUND AND RELATED WORK

A. Phishing detection approaches

Phishing detection research spans several complementary approaches. List-based methods compare a candidate URL or domain with known malicious indicators. They are efficient when threat intelligence is current, but previously unseen domains remain a challenge. Heuristic methods use rules such as URL length, number of dots, presence of an IP address, suspicious symbols, excessive subdomains, or unusual redirection behaviour. These signals are useful for early screening but are not individually conclusive: legitimate services can use long URLs, redirects, or complex subdomains.

Machine-learning approaches transform URL, domain, page, or message characteristics into features and learn a decision boundary from labelled examples. Common algorithms include Logistic Regression, Naive Bayes, Support Vector Machines, Decision Trees, Random Forests, Gradient Boosting, XGBoost, and LightGBM. Tree ensembles are common for structured URL features because they can model nonlinear feature interactions and provide feature-importance mechanisms. Sahingoz et al. examined machine-learning-based phishing detection from URLs [2]. Safi and Singh organized phishing website detection approaches into list-based, visual-similarity, heuristic, ML, and DL categories and synthesized 80 studies [1].

Deep learning can learn representations from raw URL strings, page source, screenshots, or email text. Character-level convolutional networks and recurrent architectures can capture local and sequential patterns in URLs. Transformer-based models represent contextual relationships in text or sequences. Visual models can compare page screenshots or logos with legitimate brands. Hybrid systems combine several evidence sources. Their potential benefit is broader pattern learning, but their complexity, data requirements, inference cost, and interpretability challenges may be greater than those of conventional models.

B. Feature families

URL lexical features include total length, hostname length, path length, number of dots, hyphens and digits, use of an IP address, suspicious tokens, percent-encoding, and the number of subdomains. Host and domain features can include domain age, registration information, DNS records, certificate properties, nameserver characteristics, and reputation. Page-based features may include HTML structure, forms, external resource references, iframe use, redirects, JavaScript behaviour, and similarity to known legitimate pages. Message-based features include sender identity, subject wording, urgency, requests for credentials, and embedded links.

The choice of features affects both model performance and the nature of an explanation. A URL-only detector can provide an early decision before page rendering, but it cannot directly explain visual imitation or the intent of an email. A page-content detector may capture more context but requires fetching and parsing content, which can introduce latency and expose analysis infrastructure to malicious content. Multimodal models can combine evidence, although their explanations must distinguish which modality contributed to the decision.

C. Explainable artificial intelligence

Interpretability is not a single measurable property. An explanation may be local or global, model-specific or model-agnostic, intrinsic or post-hoc. A local explanation describes a particular prediction; a global explanation summarizes behaviour over a dataset or

model. Intrinsically interpretable models, such as shallow decision trees or sparse linear models, expose their structure directly. Post-hoc methods explain an already trained model, which is useful for complex ensembles and neural networks but introduces a separate question: does the explanation faithfully describe the model? Doshi-Velez and Kim argue for rigorous evaluation of interpretability [7]. Miller emphasizes that explanations should be considered in light of how people understand and use them [8].

LIME perturbs an input, observes the black-box model's responses, and fits a simple interpretable surrogate near the instance being explained [5]. The resulting explanation is local and depends on the perturbation strategy, sampling, feature representation, and locality weighting. SHAP assigns feature contributions using a Shapley-value framework [6]. It offers a common additive explanation form and supports local attributions that can be aggregated for global analysis. Specialized SHAP explainers trade computational efficiency against assumptions about the model and data.

Neither method should be treated as a certificate of correctness. LIME's sampled neighbourhood may not represent realistic URL changes. SHAP values depend on how absent features are defined and on the background distribution used by an explainer. Correlated URL features can share or redistribute attribution. A feature receiving a large contribution is not necessarily a causal reason for phishing, and an explanation does not prove that the underlying label is correct.

D. Existing reviews and recent studies

Safi and Singh provide a broad systematic review of phishing website detection techniques [1]. Thalpage and Jayarathne specifically review XAI approaches for detecting and mitigating phishing attacks, identifying SHAP, LIME, Integrated Gradients, attention methods, and interpretable models as important families [10]. Alsuqayh, Mirza, and Alhogail examine feature engineering and XAI for phishing website detection and identify challenges involving scalability and evolving phishing strategies [11]. These reviews establish that XAI in phishing is already an active research topic; therefore, this paper does not claim to be the first review of the subject.

Recent empirical research adds focused evidence. Yi, Omotosho, and Balogun study phishing URL detection and interpretability across multiple datasets, using Random Forest, XGBoost, LightGBM, and SHAP [13]. Their cross-dataset framing is important because performance on a single dataset may overstate deployment readiness. Panthi et al. compare SHAP and LIME for boosting models on a public phishing URL dataset containing 11,054 instances and 30 URL features; they report differences in global and local explanation behaviour [12]. A 2026 IEEE conference paper evaluates adversarial robustness and SHAP explanation stability across feature-engineered and raw URL representations [14]. These studies support evaluating explanations under changing inputs and data distributions, not merely displaying attribution plots.

III. REVIEW METHODOLOGY

A. Review design

This work uses a structured narrative review rather than claiming a fully reproducible systematic review or meta-analysis. It combines foundational explainability literature, phishing-detection surveys, and recent empirical studies. This design is appropriate for synthesizing heterogeneous work in which datasets, model architectures, feature representations, attack assumptions, and explanation metrics are not directly comparable. No new phishing dataset was collected, no classifier was trained, and no experimental metrics were generated by the author.

B. Search strategy and sources

The literature search covered material available up to 31 October 2026. Search concepts combined terms for phishing detection (phishing website, phishing URL, phishing email, malicious URL), learning approaches (machine learning, deep learning, ensemble, transformer), and explainability (explainable AI, XAI, SHAP, LIME, feature attribution, explanation stability, adversarial robustness, and cross-dataset evaluation). Relevant platforms and discovery services included IEEE Xplore, ACM Digital Library, SpringerLink, ScienceDirect, Wiley Online Library, Nature/Springer Nature, and Google Scholar. Publisher pages and DOI records were preferred for bibliographic verification.

C. Inclusion and exclusion criteria

Included material comprised peer-reviewed journal and conference papers relevant to phishing detection, explainability, model interpretability, robustness, or evaluation methodology; foundational papers defining widely used explanation methods; and review articles synthesizing phishing or cybersecurity XAI research. Papers were prioritized when they provided a clear method, dataset description, evaluation approach, or direct relevance to phishing. Duplicate records, unrelated cybersecurity applications, non-verifiable bibliographic entries, and sources lacking enough information to

support a technical claim were excluded from the core synthesis. Preprints and repository records were treated cautiously and were not used as the sole basis for strong claims.

D. Data extraction and synthesis

For each selected study, the review considered the research objective, detection modality, feature family, model family, explanation technique, evaluation setting, reported contribution, and limitations. The evidence was synthesized thematically rather than pooled statistically. Reported accuracy values are retained only when explicitly attributed to the original study and are not compared as if they came from a common benchmark. Differences in datasets, train-test splits, class balance, feature extraction, and validation procedures make direct ranking across papers inappropriate.

E. Methodological limitations

The search and screening were conducted by a single student author and were not independently checked by a second reviewer. Search results may omit relevant studies, particularly very recent papers or work indexed under different terminology. The review does not provide a PRISMA flow diagram or claim exhaustive coverage. The synthesis is limited by full-text availability and inconsistent reporting of explanation-quality metrics. These constraints are disclosed so that the paper does not overstate the completeness or certainty of its conclusions.

IV. COMPARATIVE SYNTHESIS OF TECHNIQUES

A. Detection model families

Traditional ML models remain useful for structured URL and domain features. Logistic Regression offers a relatively transparent linear baseline. Decision Trees expose threshold-based rules, while Random Forest and gradient-boosting methods capture nonlinear interactions. Support Vector Machines can be effective in high-dimensional feature spaces but may require additional explanation tools. Deep neural networks can learn from raw character sequences, HTML, images, and message text, but their internal representations are more difficult to inspect directly. A practical system may use a lightweight model for rapid URL screening and a richer model when additional page or message evidence is available.

B. LIME in phishing detection

LIME can explain one URL classification by identifying which interpretable features locally support or oppose the model's prediction. For example, an explanation may show that an unusually long URL, multiple subdomains, or suspicious lexical tokens increased a phishing score. This can help an analyst inspect an individual case. However, the explanation depends on the neighbourhood generated around the URL. Randomly changing characters or tokens may create malformed or unrealistic URLs, causing the surrogate to learn from examples that do not resemble valid attacker behaviour. Repeated runs may also yield different explanations unless sampling and random seeds are controlled.

C. SHAP in phishing detection

SHAP assigns contribution values to input features relative to a reference or background distribution. For a tree-based URL detector, SHAP can provide per-instance feature contributions and aggregate patterns across many samples. It can help identify whether a model relies heavily on domain age, URL length, special characters, or other features. Panthi et al. report a comparative analysis of SHAP and LIME for gradient-boosting models and find that the two methods can produce different explanation patterns and serve different analytical purposes [12]. The result reinforces that explanation methods should be selected according to the question being asked: model inspection at population level, or a human-readable account of one decision.

D. Other explanation approaches

Integrated Gradients attributes a neural network's prediction to input dimensions by integrating gradients along a path from a baseline to the input. Attention visualizations can show which tokens or regions receive attention weights, although attention weights should not automatically be interpreted as faithful explanations. Counterfactual explanations describe a change to input features that would alter a prediction, such as which URL properties would need to differ for the model to classify a sample as benign. Rule-based and inherently interpretable models can offer transparent decision logic, but may sacrifice some representational flexibility. A robust evaluation should compare explanation methods based on their purpose and assumptions rather than treating all visualizations as interchangeable.

E. Comparison of LIME and SHAP

TABLE I. COMPARISON OF LIME AND SHAP

Dimension | LIME | SHAP

Core principle | Local surrogate approximation | Shapley-value feature attribution

Typical output | Local feature weights | Local contributions; aggregate summaries

Model dependence | Model-agnostic querying | Model-agnostic framework; specialized explainers

Main sensitivity | Perturbation and sampling design | Background distribution and feature dependence

Common use | Explain one prediction | Explain individual predictions and summarize effects

Key caution | Surrogate may not represent nearby model behaviour | Attribution is not causal proof and may be costly

F. From explanation generation to explanation validation

A recurring weakness in applied XAI is the assumption that an explanation is trustworthy because it is visually plausible. Explanation generation answers "what did the tool highlight?" Explanation validation asks "does that highlight accurately and consistently represent the model's behaviour, and is it useful for the intended person?" These are separate questions. A detector can produce a polished SHAP plot while relying on a spurious feature, or a LIME explanation can change substantially after a small, semantically irrelevant input edit.

A useful validation framework should include at least four technical checks. First, fidelity or faithfulness measures whether the explanation approximates the model's behaviour. Second, stability measures whether similar inputs produce appropriately similar explanations. Third, sensitivity checks whether changing a genuinely influential feature changes the explanation in a meaningful way. Fourth, robustness checks whether adversarial or distribution-shifted inputs cause the explanation to become misleading. Human evaluation is also needed because a mathematically consistent explanation may still be too complex, ambiguous, or poorly framed for an analyst or end user.

G. Cross-dataset generalization

Phishing datasets differ in collection date, geographic coverage, source, labeling policy, feature extraction, and class balance. A model may learn dataset-specific artifacts rather than durable indicators of phishing. Domain age or hosting patterns may correlate with labels in one collection but not another. Yi et al. evaluate URL-based models across multiple datasets and use SHAP to interpret feature contributions [13]. Their work illustrates why external evaluation is important: an explanation that appears meaningful on a training distribution may change when the model encounters another source of URLs.

H. Adversarial robustness and explanation stability

Attackers can deliberately manipulate URL strings, redirects, domain registration, page content, or visual presentation to evade detectors. Small character changes, benign-looking tokens, URL encoding, and changes in hosting infrastructure may alter model inputs. A detector should therefore be evaluated against realistic attacker capabilities, not only random noise. The 2026 IEEE study "Cross-Domain Adversarial Robustness and Explainability Stability in Phishing Detection Systems" compares raw URL and feature-engineered representations under adversarial evaluation and uses SHAP-based cosine-similarity drift to examine explanation consistency [14]. This line of work makes explanation stability a security-relevant property: if an attacker can induce a large attribution shift with a minor change, analysts may receive inconsistent evidence even when the prediction remains unchanged.

I. Human-centred and operational considerations

The intended recipient determines what counts as a useful explanation. A security analyst may need feature names, attribution values, domain context, and links to corroborating evidence. An ordinary user may need a concise warning such as "This link uses a look-alike domain and asks for account credentials," together with a safe action. An administrator may need aggregated trends and evidence for policy decisions. A single explanation format is unlikely to serve all audiences. Systems should avoid overstating certainty, distinguish model evidence from external threat intelligence, and provide a route for human review when evidence conflicts.

J. Dataset quality, leakage, and temporal validity

Dataset construction is a foundational part of phishing-detection research because a model can only learn patterns represented in its training data. Public collections are valuable for reproducibility, but their labels, collection procedures, and time coverage are not uniform. A dataset may contain URLs reported by a community, URLs collected by a crawler, or pages obtained from a commercial feed. Each source has a different reporting delay and may over-represent particular campaigns, languages, hosting providers, or geographic regions. Consequently, a high score on one benchmark should be interpreted as evidence about that benchmark rather than a universal estimate of operational performance.

A particularly important risk is data leakage. Duplicate URLs, near-duplicate pages, shared domains, or records

from the same campaign can appear in both training and test partitions. Random splitting may then allow the model to recognize campaign-specific patterns rather than generalizable phishing characteristics. More defensible protocols group related samples before splitting and, where timestamps exist, train on earlier observations and test on later observations. A time-based split is not a complete simulation of deployment, but it is more informative about temporal drift than a purely random split. Cross-dataset testing adds another layer by evaluating a model on data collected independently under a different protocol.

These dataset issues also affect explainability. If a model has learned a collection artifact, an attribution method may faithfully identify that artifact as influential. The explanation can therefore be locally faithful and still be operationally misleading. For example, a model may rely on a hosting-provider feature that is correlated with phishing in one dataset but is not a stable indicator across regions or time. Explanation validation must therefore be considered together with dataset provenance, leakage checks, and external evaluation.

K. Explanation metrics and their interpretation

There is no single metric that captures explanation quality. Fidelity evaluates whether an explanation approximates the behaviour of the model being explained. Depending on the method, fidelity may be estimated through surrogate agreement, feature deletion or insertion tests, or changes in model output after modifying features identified as important. Such tests need careful design: deleting a URL component can create an invalid URL, and an unrealistic perturbation may measure model behaviour outside the data distribution. The evaluation should state what is perturbed, why the change is plausible, and whether the resulting input remains meaningful.

Stability concerns the consistency of explanations for inputs that are equivalent or differ only in ways that should not materially affect the decision. In phishing detection, examples include harmless URL normalization, changes in letter casing where semantics are preserved, or equivalent encodings. Stability should not mean that every small change produces an identical explanation: a meaningful change in domain, path, or credential-request behaviour may legitimately alter the evidence. The objective is to distinguish appropriate sensitivity from arbitrary explanation drift.

Completeness and sparsity are also relevant. A concise explanation may be easier for an analyst to inspect, but excessive simplification can omit interacting evidence. Conversely, a long list of attributed features may be difficult to act upon. Explanation methods should disclose their reference distribution, perturbation strategy, feature grouping, and randomness settings. For neural models, gradient-based methods require particular attention to baseline selection and gradient saturation. Attention weights may be useful for visualization, but should not be treated as proof that a model relied on the highlighted tokens in a causal or faithful manner.

Finally, explanation quality is audience-dependent. A security analyst may value detailed feature attributions and corroborating telemetry, whereas an ordinary user may need a short risk statement and a safe next step. Human evaluation should therefore measure task accuracy, comprehension, time to decision, and the ability to identify uncertainty. Trust should be calibrated rather than maximized: a useful interface helps people recognize when a model is likely to be wrong or when additional review is needed.

L. Deployment architecture and lifecycle monitoring

A practical explainable phishing detector is better viewed as a pipeline than as a standalone classifier. A possible architecture begins with safe URL normalization and feature extraction, followed by a lightweight screening model. Suspicious or uncertain cases can be escalated to deeper inspection of DNS and certificate information, page structure, redirects, screenshots, or message context. The prediction service can then attach an explanation generated from the evidence actually available to the model. This separation helps avoid presenting page-level explanations when only URL-level features were examined.

The architecture should record the model version, feature schema, timestamp, data-source provenance, prediction score, explanation method, and relevant external intelligence. Such records support incident review and make it possible to determine whether a change in output resulted from a new model, updated threat intelligence, or changed input features. Where external reputation services are used, their signals should be clearly distinguished from learned model contributions. Missing or stale features should be represented explicitly rather than silently treated as benign.

Deployment also creates resource constraints. Generating SHAP explanations for every request may be expensive, especially for large ensembles or high-volume gateways.

Systems can use tiered explanation policies: fast local explanations for high-priority alerts, sampled explanations for monitoring, and deeper analysis for incidents requiring investigation. The trade-off must be measured rather than assumed. Latency, throughput, memory use, failure behaviour, and the cost of external lookups should be reported alongside predictive metrics.

Models and explanations require lifecycle monitoring. Changes in phishing campaigns, domain registration practices, browser behaviour, and legitimate web design can cause feature distributions to drift. Monitoring should track input drift, prediction rates, false-positive reports, explanation distributions, and changes in external-source coverage. Retraining should not be triggered solely by a statistical drift alarm; new labels and security review are needed to determine whether the underlying concept has changed. After retraining, explanation stability and cross-source performance should be reassessed because a model update can alter both predictions and the evidence presented to users.

M. Consolidated research agenda

The reviewed evidence suggests that future studies should move from demonstrations of explanation plots toward repeatable evaluation protocols. A useful benchmark would define common URL and page feature schemas, publish dataset provenance, document time windows, identify duplicates and campaign overlap, and include at least one temporally separated and one independently collected test set. It should compare interpretable baselines, tree ensembles, and neural models under the same data protocol, while reporting class-sensitive metrics and uncertainty.

The same benchmark should test explanation properties independently. Researchers could evaluate local fidelity, perturbation realism, stability under semantics-preserving changes, and robustness under explicitly defined attacker capabilities. For each attack scenario, papers should report both classification degradation and explanation drift. Human studies should be designed for the intended role, distinguishing security professionals from general users. Reporting templates should disclose explanation configuration and computational overhead so that results can be reproduced and compared.

There is also a need to study explanations as part of a socio-technical security process. A correct attribution does not automatically improve incident response, and an understandable warning does not guarantee safe user behaviour. Research should investigate whether explanations help analysts prioritize alerts, discover false positives, communicate evidence, and make proportionate interventions. It should also examine how explanations can reveal detection logic to adversaries and how interfaces can provide useful evidence without exposing sensitive thresholds or operational details. These directions provide a practical path toward systems that are not only accurate in controlled tests but also dependable under changing and adversarial conditions.

V. RESULTS OF THE LITERATURE SYNTHESIS

The review yields a thematic synthesis rather than new experimental results. The principal findings are summarized below.

A. Phishing detection is increasingly hybrid

The literature shows a progression from lists and heuristics toward ML, DL, and hybrid pipelines that combine URL, domain, page, visual, and message evidence [1], [2]. No single feature family is sufficient for every phishing campaign. URL-only approaches can support fast screening, while page and visual analysis can capture attacks that depend on imitation of trusted websites. Hybrid systems can increase evidence coverage but also create greater demands for data integration, latency control, and explanation design.

B. SHAP and LIME are prominent but serve different purposes

Recent reviews identify feature-attribution techniques such as SHAP, LIME, and Integrated Gradients as common XAI approaches in phishing detection [10], [11]. SHAP is useful for inspecting feature contributions across individual predictions and aggregating them for global analysis. LIME is useful for local explanations through a simplified surrogate. Comparative work indicates that their outputs are not interchangeable [12]. Selection should follow the model type, feature representation, computational budget, and target audience.

C. Explanation presence does not establish explanation quality

A detector that includes SHAP or LIME is not automatically explainable in a rigorous sense. The explanation must be tested for fidelity, stability, sensitivity, and usefulness. The literature on interpretability stresses rigorous evaluation [7], [8], while phishing-focused reviews identify real-world validation and standard explanation metrics as continuing concerns [10], [11]. This is a central finding of the synthesis.

D. Evaluation beyond one dataset is essential

Single-dataset evaluation may hide distribution-specific patterns and overestimate real-world generalization. Cross-dataset studies are therefore important for assessing whether the model and its explanations remain useful on independently collected samples [13]. Dataset documentation should include time period, source, label provenance, class balance, feature definitions, and duplicate-handling procedures.

E. Robustness and explanation stability should be evaluated together

Adversarial robustness and interpretability are often assessed separately, but phishing detection is an adversarial setting. A perturbation can affect both the predicted class and the features cited as evidence. Recent work explicitly links adversarial evaluation with SHAP explanation drift [14]. The literature therefore supports treating explanation stability under realistic URL changes as part of a detector's security evaluation.

F. Practical deployment requires audience-specific explanations

An explanation that is technically detailed may not help an ordinary user; a simplified message may omit evidence required by an analyst. The system should adapt the explanation to the role, avoid unsupported causal language, and show uncertainty. Human-centred evaluation should examine comprehension, decision quality, response time, and over-reliance on automated recommendations.

VI. DISCUSSION

A. Implications for trustworthy phishing detection

The central implication is that explainability should be treated as an evaluated property of the complete detection pipeline. A classifier's accuracy, precision, recall, and F1-score describe predictive behaviour under a particular test design. They do not establish whether the model is robust to attacker adaptation, whether its explanations are faithful, or whether people can use the explanations safely. A trustworthy system needs both sound predictions and evidence that supports appropriate human action.

TABLE II. PROPOSED EVALUATION DIMENSIONS

Dimension | Example assessment | Main question

Predictive validity | Precision, recall, F1, ROC-AUC, PR-AUC | Does the detector identify phishing reliably?

Explanation fidelity | Perturbation/deletion tests, surrogate agreement | Does the explanation reflect model behaviour?

Stability | Attribution similarity under benign transformations | Are explanations consistent for similar inputs?

Generalization | External and temporal test sets | Does behaviour transfer to new sources and time periods?

Adversarial robustness | Evasion success and performance under realistic attacks | Can an attacker manipulate the detector or its evidence?

Human usefulness | Comprehension, task accuracy, response time, trust calibration | Can the intended user act correctly?

Operational efficiency | Latency, memory, throughput, explanation cost | Can explanations be generated at deployment scale?

B. Recommended evaluation protocol

A future empirical study should begin with a documented dataset protocol. Training and test sets should be separated by time or source where possible, and duplicate URLs or pages should not leak across partitions. At least one external dataset should be reserved for cross-dataset evaluation. Models should include interpretable baselines and stronger ensemble or deep-learning baselines. Predictive performance should be reported with class-sensitive metrics, especially precision, recall, F1-score, and precision-recall analysis when classes are imbalanced.

Explanations should then be evaluated independently from prediction metrics. For local explanations, the study should document the perturbation or background-generation strategy and test whether generated samples are realistic. Fidelity tests should examine whether removing or modifying highly attributed features changes the model output as expected. Stability tests should apply semantics-preserving transformations, such as benign normalization or harmless URL formatting changes, and compare attribution vectors. For adversarial tests, transformations should be constrained by a clearly stated attacker model; arbitrary random character noise is not a sufficient substitute for realistic evasion.

Human evaluation should involve representative users. Analysts can assess whether explanations help triage alerts and identify corroborating evidence. Non-expert users can be tested for comprehension of risk messages and ability to select safe actions. Trust should be calibrated rather than maximized: users should neither ignore useful warnings nor blindly accept model output.

Studies should report participant characteristics, task design, and limitations.

C. Ethical and security considerations

Explanations can expose sensitive information about a detector's decision boundary and may help attackers infer which features to manipulate. Detailed feature feedback should therefore be considered in the threat model. Explanations can also create automation bias: an analyst may accept a confident but incorrect explanation. Security tools should show the provenance and timestamp of external intelligence, separate model-derived evidence from reputation data, and allow human overrides. Training datasets may contain biased labels or geographic and language coverage gaps; these can produce uneven error rates across domains and communities.

D. Research gaps

The synthesis identifies several opportunities. First, standardized explanation benchmarks for phishing detection are needed so fidelity and stability can be compared across studies. Second, temporal evaluation should be more common because phishing infrastructure and attacker strategies evolve. Third, explanation stability should be measured under realistic adversarial transformations, not only random perturbations. Fourth, research should compare explanations across URL, page, visual, and message modalities. Fifth, human-grounded studies should determine whether explanations improve analyst decisions and user safety. Finally, deployment research should measure the combined cost of detection and explanation generation under realistic throughput and latency requirements.

E. Limitations of this review

This review is limited by its structured narrative design, single-author screening, and heterogeneity of the included literature. It does not claim exhaustive database coverage, a PRISMA-compliant screening process, or a quantitative meta-analysis. The reviewed studies use different datasets, model families, feature sets, and evaluation protocols; reported metrics should not be interpreted as a direct leaderboard. Some recent publications may have limited independent replication. The proposed evaluation matrix is a synthesis and recommendation, not a validated standard.

VII. CONCLUSION AND FUTURE DIRECTIONS

Machine learning and deep learning have expanded phishing detection beyond static blacklists and manually specified rules. However, predictive performance alone does not provide the transparency and assurance required for cybersecurity decisions. XAI methods such as LIME and SHAP can help analysts inspect model predictions, identify influential features, and communicate evidence. Their use must nevertheless be accompanied by validation: explanations can be unstable, sensitive to sampling or background data, and unreliable under dataset shift or adversarial manipulation.

The literature reviewed here indicates that SHAP and LIME are prominent approaches but serve different explanatory purposes. Cross-dataset evaluation is necessary to assess generalization, while adversarial testing should examine both classification degradation and explanation drift. Future phishing detectors should report predictive metrics alongside explanation fidelity, stability, robustness, human usefulness, and computational cost. They should distinguish model attribution from causal evidence and external threat intelligence.

The most promising direction is an evaluation-driven, human-centred framework that treats explanations as part of the security system rather than decorative output. Such a framework should combine temporally aware and cross-dataset testing, realistic attacker models, audience-specific explanation interfaces, and transparent uncertainty reporting. These steps can help move explainable phishing detection from visually interpretable prototypes toward dependable tools for security analysts and end users.

ACKNOWLEDGMENT

The author acknowledges the academic environment and guidance provided by the Department of Computer Science and Engineering, Kamla Nehru Institute of Physical and Social Sciences (KNIPSS), Sultanpur. No external funding is declared.

REFERENCES

[1] A. Safi and S. Singh, "A systematic literature review on phishing website detection techniques," Journal of King Saud University – Computer and Information Sciences, vol. 35, no. 2, pp. 590–611, 2023, doi: 10.1016/j.jksuci.2023.01.004.

[2] O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, "Machine learning based phishing detection from URLs," Expert Systems with Applications, vol. 117, pp. 345–357, 2019, doi: 10.1016/j.eswa.2018.09.029.

[3] I. Fette, N. Sadeh, and A. Tomasic, "Learning to detect phishing emails," in Proceedings of the 16th International Conference on World Wide Web (WWW '07), 2007, pp. 649–656, doi: 10.1145/1242572.1242660.

[4] R. Basnet, S. Mukkamala, and A. H. Sung, "Detection of phishing attacks: A machine learning

approach," in Studies in Fuzziness and Soft Computing, 2008, doi: 10.1007/978-3-540-77465-5_19.

[5] M. T. Ribeiro, S. Singh, and C. Guestrin, "'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, 2016, pp. 1135–1144, doi: 10.1145/2939672.2939778.

[6] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in Advances in Neural Information Processing Systems, vol. 30, 2017, pp. 4765–4774. Available: https://arxiv.org/abs/1705.07874.

[7] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," arXiv:1702.08608, 2017.

[8] T. Miller, "Explanation in artificial intelligence: Insights from the social sciences," Artificial Intelligence, vol. 267, pp. 1–38, 2019, doi: 10.1016/j.artint.2018.07.007.

[9] F. Charmet et al., "Explainable artificial intelligence for cybersecurity: A literature survey," Annals of Telecommunications, vol. 77, pp. 789–812, 2022, doi: 10.1007/s12243-022-00926-7.

[10] N. S. Thalpage and E. Jayarathne, "Explainable AI approaches for detecting and mitigating phishing attacks: A review," Journal of Digital Science, vol. 7, no. 2, 2025, doi: 10.33847/2686-8296.7.2_4.

[11] N. Alsuqayh, A. Mirza, and A. Alhogail, "Exploring feature engineering and explainable AI for phishing website detection: A systematic literature review," International Journal of Electrical and Computer Engineering, vol. 15, no. 6, pp. 5863–5878, 2025, doi: 10.11591/ijece.v15i6.pp5863-5878.

[12] S. Panthi, M. Patidar, K. Zalawadia, D. Vekariya, and V. Vekariya, "A performance analysis of SHAP and LIME for interpreting boosting models in phishing URL detection," Scientific Reports, 25 Aug. 2026, doi: 10.1038/s41598-026-66800-x.

[13] L. Yi, A. Omotosho, and H. Balogun, "Phishing URL detection and interpretability with machine learning: A cross-dataset approach," Security and Privacy, vol. 9, no. 1, Art. no. e70175, 2026, first published 29 Dec. 2025, doi: 10.1002/spy2.70175.

[14] "Cross-domain adversarial robustness and explainability stability in phishing detection systems," in Proc. 2026 International Conference on Power, Electronics, Communications, Computing, and Intelligent Infrastructure (PECCII), 17–18 June 2026, doi: 10.1109/PECCII70991.2026.11661863.

[15] M. Tawfik, A. A. Abu-Ein, A. H. Abdelhaliem, Y. M. Al-Sharo, and I. S. Fathi, "Explainable few-shot learning with modern BERT for detecting emerging phishing attacks using XF-PhishBERT," Scientific Reports, vol. 15, Art. no. 42821, 2025, doi: 10.1038/s41598-025-27500-0.