Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

An Explainable Agentic RAG Framework for Zero-Shot Phishing and Web Attack Detection

Mohammad Zahangir Alam

mohammad.z.alam@utu.fi

University of Turku

Mahdi H Miraz

Xiamen University Malaysia

Elhan Ayath

CATECH

Sharmin Sultana

American International University-Bangladesh

Nowshad Amin

Islamic University of Madinah

Research Article

Keywords: Phishing Detection, Web Attack Detection, Explainable AI, Agentic AI, Retrieval-Augmented Generation, Zero-Shot Learning, DeBERTa, SHAP

Posted Date: September 4th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10794223/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

An Explainable Agentic RAG Framework for Zero-Shot Phishing and Web Attack Detection

Mohammad Zahangir Alam[1][0009-0009-0130-339X], Mahdi H. Miraz[2] [0000-0002-6795-7048], Elhan Ayath[3][0009-0003-8411-6552], Sharmin Sultana[4], and Nowshad Amin[5] [0000-0002-3250-1307]

[1]Department of Computing, ML and Software Engineering, University of Turku, Finland
mohammad.z.alam@utu.fi
[2]School of Computing and Data Science, Xiamen University Malaysia, Malaysia
m.miraz@ieee.org
[3]Department of Aerospace and Aviation Science & Engineering, CATECH, Bangladesh
s21511007116@nu.ac.bd
[4]Department of Computer Science, American International University - Bangladesh (AIUB), Dhaka, Bangladesh
22-48479-3@student.aiub.edu
[5]Department of Electrical Engineering, Islamic University of Madinah, KSA
nowshad@iu.edu.sa
* Correspondence: mohammad.z.alam@utu.fi

Abstract. Phishing and web attacks remain among the most damaging cyber threats in industrial and enterprise systems, yet existing detection frameworks struggle to keep pace with rapidly evolving attack strategies. Most current approaches, including recent self-supervised ensemble methods, depend heavily on large volumes of manually labelled data and static pre-trained models that cannot generalise to unseen attack patterns. To address these limitations, we propose XAR-Detect (eXplainable Agentic Retrieval-Augmented Detection), a framework that brings together agentic LLM-based reasoning, retrieval-augmented generation, and dual-layer explainability to enable zero-shot phishing and web attack detection without retraining. At the core of XAR-Detect, a DeBERTa-based encoder extracts rich contextual representations from raw email and HTTP request data, which are dynamically enriched by a RAG module that retrieves semantically matched threat signatures from a live knowledge base spanning PhishTank, VirusTotal, and CVE repositories. An autonomous reasoning agent jointly processes local features and retrieved evidence to reach detection decisions, while SHAP-driven global attribution and instance-level attention maps provide transparent, auditable justifications for every prediction. The framework is evaluated on three phishing corpora—Nazario, Enron, and Subhadeep—and the SR-BH web attack benchmark under balanced and severely imbalanced distributions to assess overall robustness and generalisability. Experimental results show that XAR-Detect achieves an F1-score of 97.8% and a false-positive rate of 2.1%, outperforming state-of-the-art self-supervised baselines by up to 19% and adaptive RL-based detectors by up to 11% under zero-shot evaluation conditions. These results demonstrate that XAR-Detect offers a practical, interpretable, and label-efficient path toward intelligent cyber threat detection in real-world deployments.

Keywords: Phishing Detection, Web Attack Detection, Explainable AI, Agentic AI, Retrieval-Augmented Generation, Zero-Shot Learning, DeBERTa, SHAP

1 Introduction

Cybersecurity threats have become a pressing challenge, with phishing and web-based attacks remaining leading causes of breaches across industrial, financial, and governmental systems. According to the Anti-Phishing Working Group (APWG), more than 1.7 million unique phishing attacks were recorded in one quarter of 2024, representing a year-on-year increase exceeding 40% [1]. Web application attacks, including SQL injection, cross-site scripting (XSS), and command injection, accounted for nearly 26% of confirmed breaches in Verizon's 2025 Data Breach Investigations Report [2]. The average cost of a data breach was estimated at approximately USD 4.9 million [3]. Meanwhile, large language model-powered attack tools have lowered the barrier to generating convincing, contextually tailored phishing campaigns that evade conventional signature-based and heuristic filters [4]. Expanding Industrial Internet of Things (IIoT) deployments and cloud-native web architectures further increase exposure, while the EU NIS2 Directive and GDPR require proactive and auditable threat detection [5]. These pressures demand accurate, adaptive, explainable, and scalable detection systems.

The research community has developed many machine learning-based frameworks, but each addresses only part of this challenge. Classical supervised methods, including decision trees, support vector machines, and random forests, perform well on benchmarks but require extensive manually labelled data and frequent retraining as attacks evolve [6]. Deep learning models, including convolutional neural networks, LSTMs, and BERT-based transformers, improve feature representation but remain static after training and cannot readily accommodate novel attack variants [7]. Self-supervised learning reduces dependence on labelled data by combining contextual NLP encoders with structural feature extractors trained on unlabelled email and HTTP request data [8], but the resulting representations remain fixed at pre-training time. Graph neural network-based detectors model relational dependencies within email header chains and HTTP request flows, yet remain computationally demanding, sensitive to sparse graph structures, and unable to provide retrieval-based generalisation to out-of-distribution attacks [7].

More recently, standalone GPT- and LLaMA-based classifiers have demonstrated strong semantic understanding of phishing content, but their parametric knowledge remains frozen at the pre-training cutoff, limiting awareness of emerging signatures and campaign-specific indicators of compromise [4], [7]. Reinforcement learning-based detectors provide adaptive decision-making through reward-driven policy optimisation. Nevertheless, they often remain opaque, provide limited human-interpretable justification, and require extensive interaction episodes to converge, restricting real-time deployment [7]. Consequently, no existing framework jointly addresses zero-shot generalisation, agentic adaptive reasoning, real-time threat knowledge grounding, and transparent explainability within a unified phishing and web attack detection architecture.

To address these limitations, we propose XAR-Detect (eXplainable Agentic Retrieval-Augmented Detection), a framework designed to overcome static, opaque, and

data-intensive detection. Its design combines three capabilities generally treated separately: autonomous reasoning over heterogeneous evidence through multi-step inference, grounding decisions in current external threat intelligence rather than frozen training distributions, and explaining detections in a form that security analysts can interpret and operationalise. XAR-Detect integrates a DeBERTa-based contextual encoder, selected for its disentangled attention mechanism and relative position encoding [9]. The suitability of DeBERTa for cross-domain representation learning has also been demonstrated in recent classification research [10]. The encoder is coupled with a Retrieval-Augmented Generation (RAG) module that dynamically queries a live threat intelligence knowledge base comprising PhishTank, VirusTotal, and CVE repositories at inference time. Retrieved signatures enrich local representations, enabling detection of attack patterns excluded from training without retraining or fine-tuning.

Unlike conventional RAG pipelines that passively retrieve and concatenate context, XAR-Detect employs an autonomous LLM-based reasoning agent that selects, weighs, and synthesises retrieved evidence with local features through a structured multi-step process. This extends the agentic RAG paradigm demonstrated by CyberRAG [11] from cybersecurity classification and reporting to unified phishing and web attack detection. The design is federated-ready because the reasoning module operates on encoded representations and public threat intelligence, supporting distributed deployment across privacy-sensitive organisations. Finally, a dual-layer explainability module combines SHAP-based global attribution with instance-level attention maps, providing model-wide feature importance and case-specific justification. Together, these components form an adaptive and auditable detection pipeline. The main contributions are as follows:

- We propose XAR-Detect, integrating LLM-based reasoning, dynamic threat intelligence retrieval, and dual-layer explainability for phishing and web attack detection.
- We introduce a RAG-enhanced mechanism that enables zero-shot detection of previously unseen variants without retraining.
- We develop an autonomous reasoning agent that combines local feature representations with retrieved evidence through multi-step inference.
- We provide global SHAP analysis and instance-level attention attribution for model-wide and case-specific transparency.
- We validate XAR-Detect under balanced, imbalanced, and zero-shot settings, achieving an F1-score of 97.8% and outperforming state-of-the-art approaches by up to 19%.

Section 2 reviews related work; Section 3 formulates the problem; Section 4 presents XAR-Detect; Sections 5 and 6 describe the experiments and results; Section 7 discusses findings and limitations; and Section 8 concludes the paper.

2 Related Work

2.1 Machine Learning, Deep Learning, and Self Supervised Detection

Early phishing and web attack detection systems relied on classical machine learning techniques, including decision trees, support vector machines, naïve Bayes classifiers, and random forests. These approaches achieved strong performance on benchmark datasets using handcrafted features derived from URLs, HTML structures, and email metadata [6]. However, their dependence on manually labelled data and static decision boundaries limits adaptability under evolving attack conditions and distribution shift. Subsequent deep learning models, including CNNs, LSTMs, and transformer-based architectures such as BERT, substantially improved contextual representation learning by capturing semantic dependencies within email and HTTP request content [7]. More recently, self-supervised learning approaches have reduced reliance on labelled datasets by learning representations from unlabelled data through pretext tasks and multi-view feature extraction. Notable examples combine contextual NLP encoders with structural representations derived from email and HTTP requests [8]. Although these methods improve detection performance and reduce annotation costs, they remain fundamentally closed-world systems whose representations are fixed after training and cannot naturally incorporate emerging threat intelligence. Consequently, previously unseen attack categories generally require retraining or model updating.

2.2 Explainable and Adaptive Cyber Threat Detection

Growing regulatory and operational demands have increased interest in explainable AI for cybersecurity. SHAP-, LIME-, and ensemble-based frameworks improve decision transparency by identifying the features underlying phishing predictions [12], [13]. More advanced methods such as HXRF combine SHAP, LIME, partial dependence, and permutation importance with random forest classification [14]. However, these approaches typically apply explainability post hoc to static models and cannot dynamically incorporate emerging threat knowledge. Reinforcement learning-based detectors, including HARE and RAIDER, improve adaptability through human-guided reward shaping and dynamic feature selection [15], [16]. Nevertheless, they may require repeated interaction or expert feedback and generally provide limited evidence-grounded justification. Existing approaches therefore tend to address either transparency or adaptability rather than integrating both within a unified detection framework.

2.3 Retrieval-Augmented and Agentic AI for Cybersecurity

Retrieval-Augmented Generation grounds LLM outputs in current external knowledge by retrieving relevant evidence from threat intelligence repositories. In cybersecurity, it has been applied to threat investigation, incident classification, reporting, and intelligence analysis. CyberRAG, for example, combines retrieval with agentic reasoning for cyberattack classification and reporting [11]. However, most existing RAG systems remain focused on question answering, investigation support, or domain-specific classification rather than unified real-time phishing and web attack detection.

Agentic AI extends this capability through multi-step reasoning, tool use, evidence refinement, and autonomous decision-making. GEPAgent uses online search tools to gather webpage and brand information for phishing detection [17], while PhishAgent combines multimodal reasoning with online and offline knowledge bases [18]. PhishDebate employs specialised agents to analyse URL structure, HTML content, semantics, and brand impersonation through structured debate [19]. ChatSpamDetector applies LLMs to phishing-email classification [20], whereas MA-IDS integrates multi-agent reasoning, RAG, and an experience library for IoT intrusion detection [21].
Despite these advances, existing systems remain largely task-specific and rarely combine retrieval-grounded reasoning with both global and instance-level explainability. This limits their ability to deliver adaptive detection and transparent justification within a unified phishing and web attack detection framework.

2.4 Research Gaps and Positioning of XAR-Detect

As summarised in Table 1, existing methods address only subsets of the capabilities required for adaptive, open-world, and explainable detection. Supervised and self-supervised models remain closed-world, RL approaches offer limited transparency, XAI methods explain static models, and recent RAG- and agent-based systems remain largely domain-specific. No prior framework jointly combines label-efficient learning, zero-shot generalisation, retrieval-augmented threat intelligence, agentic reasoning, explainability, and unified phishing and web attack detection. XAR-Detect addresses this gap by integrating DeBERTa-based contextual encoding, dynamic threat retrieval, autonomous multi-step reasoning, and dual-layer explainability. This positioning motivates the open-world formulation and system architecture presented in the following sections.

Table 1. Comparative analysis of phishing and web attack detection frameworks

Work	Method	Label-efficient	Zero-shot	RAG	Agentic	XAI	Phishing + Web	Adaptive
Kayes et al. [8]	SSL ensemble + FCNN	◑	✗	✗	✗	✗	✓	✗
HARE [15]	RL with human feedback	✗	✗	✗	✗	✗	✗	✓
PhishDebate [19]	Multi-agent LLM	✗	✗	✗	◑	✗	✗	✗
PhishAgent [18]	Multimodal LLM agent	✗	◑	✗	◑	✗	✗	✗
CyberRAG [11]	Agentic RAG classification	✓	◑	✓	✓	✗	✗	◑
Uddin et al. [12]	SHAP-based XAI ensemble	✗	✗	✗	✗	✓	✗	✗
HXRF [14]	XAI feature selection + RF	✗	✗	✗	✗	✓	✗	✗
MA-IDS [21]	LLM-based IoT IDS agent	✗	✗	✗	✓	◑	✗	◑
XAR-Detect	**DeBERTa + agentic RAG + XAI**	✓	✓	✓	✓	✓	✓	✓

3 Problem Formulation

3.1 Formal Problem Setup

This section formulates XAR-Detect as an open-world phishing and web attack detection problem that combines contextual encoding, retrieval-augmented threat

intelligence, agentic reasoning, and explainable decision-making. Let $D = \{1,2,\dots,N\}$ denote the index set of input samples, and let $\mathrm{x}_i \in \mathbb{R}^F$ represent the raw feature vector of the i-th email or HTTP request, where F is the feature dimensionality. The input corpus is defined as

$$\mathcal{X} = \{\mathrm{x}_1, \mathrm{x}_2, \dots, \mathrm{x}_N\} \in \mathbb{R}^{N \times F}. \qquad (1)$$

Each x_i contains lexical, syntactic, and structural attributes extracted from email text or HTTP request payloads. Let $y_i \in \mathcal{Y} = \{0,1,2\}$, where 0, 1, and 2 denote benign, phishing, and web attack classes, respectively. A conventional supervised detector adopts the closed-world mapping

$$\hat{y}_i = f(\mathrm{x}_i; \theta), \qquad (2)$$

where $f(\cdot)$ is parameterised by θ. Such models operate on fixed training distributions, provide limited support for attack classes absent from $\mathcal{X}_{train}$, and offer no direct mechanism for zero-shot detection or transparent decision justification. XAR-Detect instead formulates detection as retrieval-augmented conditional inference. For each input x_i, the framework retrieves the top-K semantically relevant threat signatures from an external knowledge base $\mathcal{K}$:

$$\mathcal{R}_i = \mathrm{Retrieve}(\mathrm{x}_i, \mathcal{K}, K) = \{r_1^i, r_2^i, \dots, r_K^i\}, \qquad (3)$$

where $r_k^i \in \mathcal{K}$ denotes a semantically matched threat pattern. The local contextual representation and retrieved evidence are then fused as

$$\tilde{\mathrm{x}}_i = \mathrm{Enc}(\mathrm{x}_i) \oplus \mathrm{Agg}(\mathcal{R}_i), \qquad (4)$$

where $\mathrm{Enc}(\cdot)$ denotes the DeBERTa-based contextual encoder [9-10], $\mathrm{Agg}(\cdot)$ is the attention-weighted aggregation of retrieved evidence, and $\oplus$ denotes feature concatenation. Rather than applying a single-pass classifier to $\tilde{\mathrm{x}}_i$, XAR-Detect performs iterative agentic reasoning. At reasoning step $s \in \{1, \dots, S\}$,

$$\mathrm{a}_i^{(s)} = \mathrm{Agent}\left(\tilde{\mathrm{x}}_i, \mathcal{R}_i, \mathrm{a}_i^{(S-1)}\right), \qquad (5)$$

where $\mathrm{a}_i^{(s)}$ denotes the internal reasoning state. The final prediction is obtained from the terminal state:

$$\hat{y}_i = \arg\max_{c \in \mathcal{Y}} P\left(y_i = c \mid \mathrm{a}_i^{(S)}\right). \qquad (6)$$

To provide transparent decisions, XAR-Detect combines global SHAP attribution with instance-level attention analysis [12-14]. The global contribution of each feature is represented as

$$f(\tilde{\mathbf{x}}_i) = \mathbb{E}[f] + \sum_{j=1}^{2d} \phi_j(\tilde{\mathbf{x}}_i), \qquad (7)$$

where ϕ_j denotes the marginal contribution of feature dimension j. Instance-level attention attribution is computed as

$$\mathbf{A}_i = \frac{1}{N_l L_a} \sum_{n=1}^{N_l} \sum_{l=1}^{L_a} \mathrm{Attn}_{n,l}(\tilde{\mathbf{x}}_i), \qquad (8)$$

where N_l and L_a denote the numbers of encoder layers and attention heads, respectively. The overall learning objective is

$$\min_{\theta} \mathcal{L}(\theta) = \mathcal{L}_{cls}(\hat{y}_i, y_i) + \lambda_1 \mathcal{L}_{ret}(\mathcal{R}_i) + \lambda_2 \mathcal{L}_{exp}(\phi_j), \qquad (9)$$

where $\mathcal{L}_{cls}$ is the classification loss, $\mathcal{L}_{ret}$ penalises semantically misaligned retrievals, and $\mathcal{L}_{exp}$ promotes attribution consistency across semantically similar inputs. The coefficients $\lambda_1, \lambda_2 \geq 0$ control the contribution of the auxiliary objectives. Open-world robustness is expressed as

$$P(\hat{y}_i \neq y_i \mid \mathbf{x}_i \notin \mathcal{X}_{train}) \leq \epsilon, \qquad (10)$$

where ϵdenotes the tolerated zero-shot error on unseen attack classes.

3.2 Retrieval-Augmented Error Reduction

The first analytical result characterises the conditions under which relevant threat retrieval can reduce expected error for unseen-class detection. ***Proposition 1.*** Let $\epsilon_{base}(\mathrm{x}_i)$ and $\epsilon_{RAG}(\mathrm{x}_i)$denote the classification errors of the encoder-only baseline and XAR-Detect, respectively, for $\mathrm{x}_i \in \mathcal{X}_{unseen}$, where $\mathcal{X}_{unseen} \cap \mathcal{X}_{train} = \emptyset$. Assume that at least one retrieved signature $r_k^* \in \mathcal{R}_i$satisfies

$$\cos(r_k^*, \mathrm{x}_i) \geq \tau, \tau \in (0,1). \qquad (P1.1)$$

Then, under the assumption that the retrieved context contributes positive label-relevant information beyond the local encoder representation,

$$\mathbb{E}_{\mathrm{x}_i \sim \mathcal{X}_{unseen}}[\epsilon_{RAG}(\mathrm{x}_i)] \leq \mathbb{E}_{\mathrm{x}_i \sim \mathcal{X}_{unseen}}[\epsilon_{base}(\mathrm{x}_i)] - \Delta(\tau, K), \qquad (P1.2)$$

where $\Delta(\tau, K) > 0$ represents the expected gain from relevant retrieval. ***Analytical justification.*** For unseen inputs, the local encoder representation may lie outside the training support and therefore carry high epistemic uncertainty. When condition (P1.1) holds, the retrieved context contributes additional threat-relevant evidence to the augmented representation. Hence,

$$I(\tilde{\mathrm{x}}_i; y_i) \geq I(\mathrm{e}_i; y_i),$$

where $\mathrm{e}_i = \mathrm{Enc}(\mathrm{x}_i)$. If the retrieved evidence provides positive conditional mutual information,

$$I(\mathrm{c}_i; y_i \mid \mathrm{e}_i) > 0,$$

then, under these assumptions, Fano's inequality [22] implies a lower expected classification-error bound for the augmented representation than for the encoder-only baseline. Accordingly, the expected gain $\Delta(\tau, K)$ *can incrarse with* retrieval relevance and the probability of obtaining an informative signature.

3.3 Bounded Zero-Shot Generalisation

The second analytical result relates the expected zero-shot error to the coverage of relevant signatures in the external knowledge base. ***Proposition 2.*** Let

$$\rho(\mathcal{K}, \mathcal{X}_{unseen}) \in [0,1]$$

denote the proportion of unseen attack classes for which at least one signature $r \in \mathcal{K}$satisfies $\cos(r, \mathrm{x}) \geq \tau$. Define the generalisation error as

$$\mathcal{E}(f, \mathcal{P}) = \mathbb{E}_{\mathrm{x} \sim \mathcal{P}}[1\{f(\mathrm{x}) \neq y\}]$$

Then the zero-shot error of XAR-Detect is bounded by

$$\mathcal{E}(f_{XAR}, \mathcal{X}_{unseen}) \leq \rho\, \mathcal{E}_{base}^{(seen)} + (1-\rho)\, \mathcal{E}_{max}, \qquad (P2.1)$$

where $\mathcal{E}_{base}^{(seen)}$is the baseline error on represented attack classes and $\mathcal{E}_{max} \leq 1$is the worst-case error on uncovered classes. ***Proof sketch.*** Partition the unseen attack space into a covered subset

$$\mathcal{C}^+ = \{\mathrm{x} \in \mathcal{X}_{unseen} : \exists r \in \mathcal{K}, \cos(r, \mathrm{x}) \geq \tau\}$$

and an uncovered subset

$$\mathcal{C}^- = \mathcal{X}_{unseen} \setminus \mathcal{C}^+.$$

By definition, the relative sizes of these subsets are ρ and $1-\rho$. For samples in $\mathcal{C}^+$, Proposition 1 supports reduced error through relevant retrieval. For $\mathcal{C}^-$, no retrieval

guarantee is available, so the error is bounded by $\mathcal{E}_{max}$. Under these assumptions, the bound becomes tighter as ρ increases, providing analytical motivation for continuously updating the knowledge base with newly confirmed threat signatures.

4 Proposed System Framework: XAR-Detect

Building on the open-world formulation in Section 3, XAR-Detect operationalises contextual encoding, retrieval-augmented threat grounding, agentic reasoning, and dual-layer explainability within a unified detection pipeline. The encoder produces the local representation in Eq. (4), the RAG module retrieves and aggregates external threat evidence as defined in Eqs. (3)– (4), the reasoning agent implements Eqs. (5)–(6), and the explainability module realises Eqs. (7)– (8). The complete workflow is illustrated in Fig. 1.

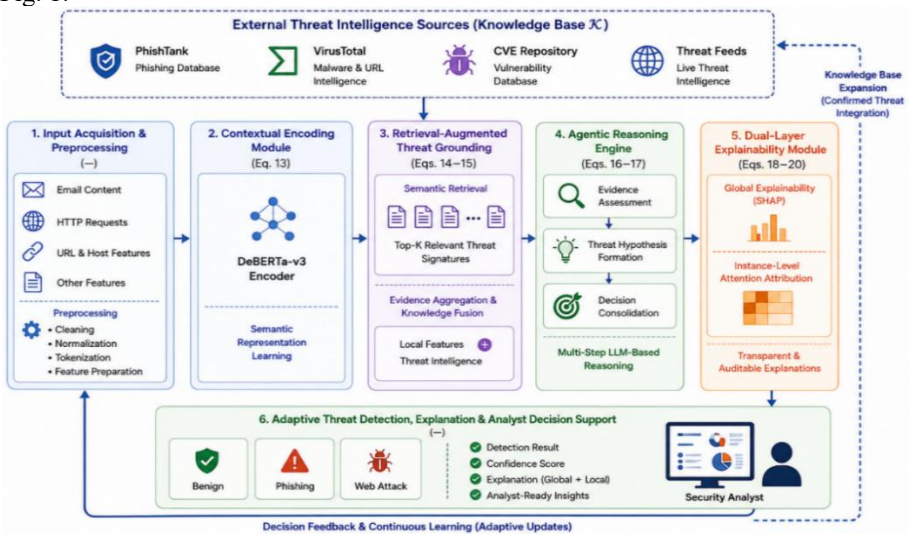


Fig. 1. Overall architecture of XAR-Detect, showing input preprocessing, DeBERTa-based contextual encoding, retrieval-augmented threat grounding, multi-step agentic reasoning, dual-layer explainability

4.1 System Architecture and Preprocessing

For an input email or HTTP request (x_i), XAR-Detect follows four sequential stages:

$$\mathrm{x}_i \rightarrow \mathrm{e}_i \rightarrow \tilde{\mathrm{x}}_i \rightarrow \hat{y}_i \rightarrow \mathrm{z}_i, \qquad (11)$$

where e_i is the contextual encoding, $\tilde{\mathrm{x}}_i$ is the retrieval-augmented representation, $\hat{y}_i$ is the detection decision, and z_i is the explanation output. Specifically, Step 1 performs Input Acquisition & Preprocessing; Step 2 applies the Contextual Encoding Module; Step 3 performs Retrieval-Augmented Threat Grounding using external threat-intelligence sources; Step 4 applies the Agentic Reasoning Engine for evidence assessment, threat-hypothesis formation, and decision consolidation; Step 5 generates transparent

explanations through the Dual-Layer Explainability Module; and Step 6 provides Adaptive Threat Detection, Explanation & Analyst Decision Support, with analyst-validated feedback supporting subsequent knowledge-based updates. As shown in Fig. 1, the pipeline combines email and HTTP inputs, external threat-intelligence sources, multi-step reasoning, analyst-facing explanations, and feedback-based knowledge-base updates [6-8]. Raw samples were drawn from the Nazario, Enron, and Subhadeep email corpora and the SR-BH web-attack benchmark [8]. Preprocessing included data cleaning, tokenisation, feature normalisation, and class-imbalance handling. Synthetic Minority Over-sampling Technique (SMOTE) was applied only to the training partition to prevent information leakage into the validation and test sets [23].

$$\mathcal{X}_{train}^{bal} = \mathrm{SMOTE}(\mathcal{X}_{train}, \mathcal{Y}_{train}). \qquad (12)$$

This prevents information leakage while reducing class-frequency bias in the encoder and retrieval stages.

4.2 Contextual Encoding and Retrieval-Augmented Grounding

XAR-Detect employs DeBERTa-v3 as the contextual encoder because its disentangled attention mechanism separately represents content and relative position, supporting both semantic phishing analysis and structural interpretation of HTTP requests [9-10]. For token sequence $\{h_{i,1}, \ldots, h_{i,T}\}$, the contextual representation is obtained through mean pooling:

$$\mathbf{e}_i = \frac{1}{T}\sum_{t=1}^{T} h_{i,t}\,, \qquad (13)$$

where $T$is the sequence length and $\mathrm{e}_i \in \mathbb{R}^d$is used as the semantic retrieval query. At inference time, the RAG module searches a threat knowledge base $\mathcal{K}$integrating PhishTank, VirusTotal, and CVE repositories. For each query e_i, the top-K signatures are selected by cosine similarity:

$$\mathcal{R}_i = \mathrm{TopK}_{r\in\mathcal{K}} \frac{\mathrm{e}_i^{\top}\mathrm{r}}{\|\,\mathrm{e}_i\,\|\;\;\|\,\mathrm{r}\,\|}. \qquad (14)$$

The retrieved signatures are combined through attention-weighted aggregation:

$$\mathrm{c}_i = \sum_{k=1}^{K} \alpha_{ik}\,\mathrm{r}_k^i, \alpha_{ik} = \frac{\exp(\mathrm{sim}(\mathrm{e}_i, \mathrm{r}_k^i))}{\sum_{j=1}^{K}\exp(\mathrm{sim}(\,\mathrm{e}_i, \mathrm{r}_j^i))}, \qquad (15)$$

and fused with the local representation:

$$\tilde{\mathrm{x}}_i = \mathrm{e}_i \oplus \mathrm{c}_i.$$

This mechanism implements the retrieval condition introduced in ***Proposition 1***. When relevant threat evidence is available, the augmented state supplies information absent from the encoder-only representation and supports detection of attack classes excluded from training.

4.3 Agentic Reasoning and Detection

The augmented representation is processed by an LLM-based agent through three structured reasoning stages: (1) Evidence assessment—evaluates consistency between local features and retrieved signatures; (2) Hypothesis formation—generates candidate

benign, phishing, or web attack labels with confidence estimates; and (3) Decision consolidation—resolves competing hypotheses and produces the final prediction. GPT-4o was instructed to return a confidence score for each of the three candidate classes. These scores were normalised across the benign, phishing, and web-attack classes to obtain the final class-confidence distribution, and the class with the highest normalised confidence was selected as the prediction. These confidence values are used as decision scores rather than calibrated probabilistic estimates. This design is informed by recent agentic cybersecurity frameworks that use multi-step reasoning, tool interaction, and evidence refinement [17–21]. At step s, the reasoning state is updated as

$$\mathrm{a}_i^{(s)} = \mathrm{Agent}\left(\tilde{\mathrm{x}}_i, \mathcal{R}_i, \mathrm{a}_i^{(s-1)}\right), \qquad (16)$$

and the terminal state is mapped to the final class:

$$\hat{y}_i = \arg\max_{c \in \mathcal{Y}} P\left(y_i = c \mid \mathrm{a}_i^{(S)}\right). \qquad (17)$$

Unlike a single-pass classifier, this process allows the model to reassess retrieved evidence, refine candidate hypotheses, and consolidate a confidence-aware decision. It therefore provides inference-time adaptation without the repeated policy-training cycles required by reinforcement learning-based detectors such as HARE [15].

4.4 Dual-Layer Explainability and Feedback

XAR-Detect combines global and instance-level explanation mechanisms to support transparent decision analysis. The global layer applies SHAP over the augmented representation [12-14]:

$$\phi_j = \mathrm{SHAP}_j(\tilde{\mathrm{x}}_i, \hat{y}_i), \qquad (18)$$

where ϕ_j measures the contribution of each encoder feature and retrieved threat dimension to the prediction. This enables population-level analysis of the features and threat-intelligence sources that consistently influence detection outcomes. The instance-level layer aggregates token attention across encoder layers and heads:

$$\mathbf{A}_i = \frac{1}{N_l L_a} \sum_{n=1}^{N_l} \sum_{l=1}^{L_a} \mathrm{Attn}_{n,l}(\tilde{\mathbf{x}}_i), \qquad (19)$$

producing a sample-specific attribution map over input tokens and retrieved evidence. The final explanation output is

$$\mathrm{z}_i = \{\phi_i, \mathrm{A}_i, \gamma_i\}, \qquad (20)$$

where γ_i denotes the prediction confidence. Together, these outputs provide model-wide attribution, instance-level evidence inspection, and confidence-aware decision support. As illustrated in Fig. 1, analyst-validated detections can be incorporated into the threat knowledge base through periodic updates. This feedback mechanism expands knowledge coverage ρ, supporting the coverage-dependent zero-shot bound established in Proposition 2.

5 Experimental Setup and Data Processing

This section describes the experimental design used to evaluate XAR-Detect under balanced, imbalanced, and zero-shot conditions. It also specifies the dataset preparation, baseline configuration, implementation settings, and evaluation procedures used to ensure that all methods are compared under consistent experimental conditions.

5.1 Evaluation Design

The experimental evaluation assessed five aspects of XAR-Detect: overall performance against representative baselines, generalisation to unseen attack classes, the contribution of individual components through ablation, sensitivity to the RAG parameters and, and the faithfulness and stability of the dual-layer explainability mechanism. Experiments were conducted under both balanced and imbalanced data distributions to evaluate controlled classification performance and operational robustness under realistic class imbalance. Unless otherwise stated, the proposed framework and all baseline methods were evaluated using the same dataset partitions, preprocessing procedures, and evaluation metrics to ensure a consistent comparison.

Datasets and Preprocessing. Four public benchmark datasets were used to cover phishing emails, legitimate organisational messages, and web attack traffic [8]. Their characteristics are summarised in Table 2. The Nazario Phishing Corpus contains real-world phishing emails collected between 2004 and 2023. Following [6], the 2015–2023 subset was selected to capture more recent credential-harvesting, malicious-URL, and social-engineering patterns, yielding 4,555 phishing emails. The Enron Email Corpus supplied 16,545 legitimate organisational emails. Only ham messages were retained to avoid overlap between conventional spam and phishing content. The Subhadeep Email Corpus contains both benign and phishing-labelled messages; however, because part of its malicious subset corresponded to ordinary spam, only 11,322 benign emails were retained, consistent with the preparation strategy in [6]. The SR-BH 2020 dataset provided approximately 36,000 web traffic instances spanning 13 attack categories and supported evaluation of the web attack detection component [8]. Because several benchmark sources predate the latest threat-intelligence records available to the RAG module, the zero-shot setting should be interpreted as unseen with respect to the model's training partition rather than necessarily as previously undocumented in external threat repositories.

Table 2. Dataset Statistics Used in the Evaluation of XAR-Detect

Dataset	Domain	Samples Used	Attack Samples/ Classes	Benign Samples	Collection Period
Nazario, Phishing Corpus [8]	Phishing Email	4,555	4,555 phishing emails	–	2015–2023
Enron, Email Corpus [8]	Legitimate Email	16,545	–	16,545	1998–2002
Subhadeep, Email Corpus [8]	Legitimate Email	11,322	–	11,322	N/A
SR-BH 2020 [8]	Web Attack Traffic	~36,000	13 attack classes	Included	2020

All datasets underwent cleaning, label verification, duplicate removal, tokenisation, feature normalisation, and stratified train-validation-test partitioning. Two evaluation

settings were then constructed. In the balanced setting, equal numbers of benign and malicious samples were selected through random under sampling of the majority class. In the imbalanced setting, the natural distribution was preserved, with attacks representing approximately 15–20% of the data. SMOTE [23], as defined in Eq. (12), was applied only to the training partition to prevent leakage into validation and test sets.

Baselines and Evaluation Metrics. *Seven* baselines were selected to represent the methodological categories reviewed in Section 2. SVM with TF-IDF served as the classical supervised baseline, while BERT-FCNN represented transformer-based deep learning. The SSL Ensemble [8] was the principal self-supervised comparator, combining BERT-derived contextual representations with CNN-based structural features. HXRF [14] represented explainable detection through XAI-guided feature selection and Random Forest classification. HARE [15] represented adaptive reinforcement learning with human feedback. PhishDebate [19] and CyberRAG [11] were included as agentic baselines, with CyberRAG providing the closest comparison because it combines external retrieval with LLM-based reasoning. All reproduced baselines were evaluated using the same dataset partitions, preprocessing pipeline, and evaluation metrics as XAR-Detect; where original implementations or published configurations were used, their reported architectural and training settings were retained as closely as possible within the common evaluation protocol.

Experimental Protocol. A stratified 70/10/20 split was applied to all datasets, with the test partition reserved for final evaluation. For zero-shot testing, complete attack classes were excluded from training, following the unseen-class setting in Section 3. RAG sensitivity was assessed through a grid search over retrieval count and similarity threshold, with optimal values selected on the validation set. Five ablation settings isolated the contribution of each architectural component. Explainability was evaluated using SHAP faithfulness and attention stability [12–14], [25]. Faithfulness was measured by the correlation between attribution magnitude and performance degradation after masking the corresponding feature. Stability was computed as the cosine similarity between attribution maps for semantically equivalent paraphrased inputs, assessing consistency with Eq. (19). Each experimental configuration was repeated across multiple independent runs, and the reported results were aggregated across these runs to reduce sensitivity to random variation. For the GPT-4o reasoning component, identical prompts, retrieval settings, and decision instructions were maintained across runs to control output variability.

Implementation Details. XAR-Detect used DeBERTa-v3-base as the contextual encoder [9–10] and GPT-4o as the three-stage reasoning agent described in Section 4.3. The same reasoning prompt, class definitions, retrieval configuration, and decision instructions were maintained across all GPT-4o evaluations to improve reproducibility. The threat knowledge base combined over 500,000 PhishTank URLs, the VirusTotal API, and the CVE/NVD feed, with 768-dimensional embeddings indexed using FAISS Flat L2 [23]. KernelSHAP used 100 background samples over 2,000 test instances. The encoder was fine-tuned with AdamW [26] for 10 epochs using a batch size of 32 and validation-based early stopping. Retrieval, regularisation, and decision thresholds were selected through validation-set grid search. Experiments were conducted on an NVIDIA A100 80-GB GPU using PyTorch 2.3 and Transformers 4.40.

6 Results And Analysis

In this section, we examine how XAR-Detect performs in relation to the theoretical claims developed in Section 3, the framework components introduced in Section 4, and the experimental protocol described in Section 5. We report the results under balanced, imbalanced, and zero-shot conditions, and then analyse the contribution of individual components through ablation, retrieval sensitivity, reasoning-depth, and explainability evaluations.

Overall Detection Performance. Table 3 compares XAR-Detect with seven representative baselines under the balanced evaluation setting. XAR-Detect achieved the strongest overall performance, with 98.1% accuracy, 97.9% precision, 97.6% recall, 97.8% F1, 2.1% FPR, and 0.991 AUC. It improved F1 by 2.4 percentage points over the SSL Ensemble [6] and by 1.1 points over CyberRAG [14], while reducing FPR by 2.7 points relative to [6]. These findings support the retrieval-augmented error-reduction argument in *Proposition 1*.

Table 3. Overall detection performance under the balanced evaluation setting

Method	Accuracy	Precision	Recall	F1	FPR	AUC
SVM + TF-IDF	91.2	90.8	91.5	91.1	8.9	0.934
BERT-FCNN	96.1	95.8	96.4	96.1	4.2	0.971
SSL Ensemble [8]	95.4	95.1	95.7	95.4	4.8	0.968
HXRF [14]	94.3	94.0	94.6	94.3	5.7	0.955
HARE [15]	95.9	95.5	96.2	95.8	4.1	0.972
PhishDebate [19]	96.4	96.1	96.7	96.4	3.8	0.976
CyberRAG [11]	96.8	96.5	97.0	96.7	3.3	0.979
XAR-Detect	**98.1**	**97.9**	**97.6**	**97.8**	**2.1**	**0.991**

Robustness Under Class Imbalance. *Table 4* reports performance when attacks represent approximately 15% of the data. Although class imbalance reduced all baseline scores, XAR-Detect retained 96.2% F1 with 3.8% FPR. Its F1 decreased by only 1.6 points, compared with 5.3 points for the SSL Ensemble and 3.6 points for CyberRAG. This indicates that the training-only SMOTE [23 procedure in Eq. (12), together with retrieval-grounded reasoning, improves resilience to class-frequency shifts.

Table 4. Performance under the imbalanced evaluation setting

Method	F1	FPR	F1 drop
SVM + TF-IDF	83.4	16.6	−7.7
BERT-FCNN	91.2	8.8	−4.9
SSL Ensemble [6]	90.1	9.9	−5.3
HARE [11]	91.8	8.2	−4.0
CyberRAG [14]	93.1	6.9	−3.6
XAR-Detect	96.2	3.8	−1.6

Zero-Shot Generalisation. *Fig. 2.* compares balanced seen-class F1 with zero-shot F1 for attack classes excluded entirely from training. XAR-Detect achieved 88.3% zero-shot F1 and 11.7% FPR, exceeding the SSL Ensemble [6] by 34.2 percentage points and CyberRAG [14] by 14.1 points. Although zero-shot performance remained below the seen-class result of 97.8%, XAR-Detect exhibited the smallest degradation among

the evaluated methods. This supports the coverage-dependent generalisation analysis in *Proposition 2* and indicates that external threat retrieval and agentic evidence synthesis improve open-world detection.

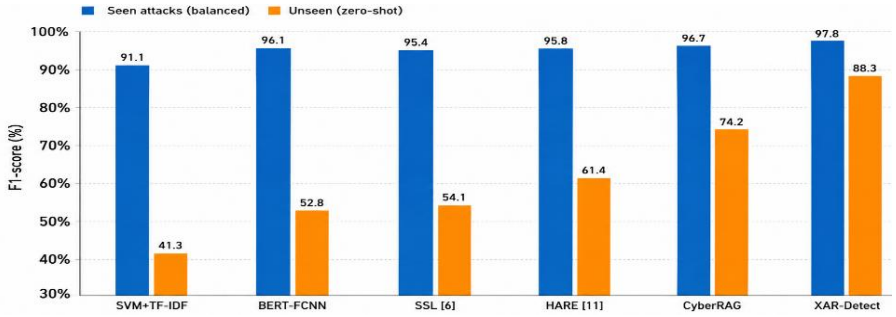


Fig. 2. Comparison of balanced seen-class and zero-shot F1 across baseline methods

Component Ablation. *Table 5* isolates the contributions of contextual encoding, retrieval augmentation, agentic reasoning, and explainability. The encoder-only configuration achieved 94.8% F1. Adding RAG increased F1 to 96.1%, whereas the agent without RAG achieved 95.7%. Combining both components produced 97.8% F1, showing that retrieval and multi-step reasoning provide complementary gains. Removing XAI did not affect F1 or FPR, confirming that the explanation module is non-interventional and does not modify the detection decision.

Table 5. Ablation results — balanced evaluation.

Condition	Configuration	F1	FPR	Gap vs full
A1	Encoder only (DeBERTa + FCNN)	94.8	5.2	−3.0
A2	RAG, no agent	96.1	3.9	−1.7
A3	Agent, no RAG	95.7	4.3	−2.1
A4	Full pipeline, no XAI	97.8	2.1	0.0*
A5	***Full XAR-Detect***	***97.8***	***2.1***	—

The combined improvement of 3.0 points exceeded the sum of the isolated gains, indicating a positive interaction between retrieved threat evidence and agentic reasoning. These findings support the modular design in Section 4 and the retrieval assumption underlying *Proposition 1*.

Retrieval Sensitivity. *Fig. 3.* reports F1 as a function of retrieval count K and similarity threshold τ. Performance increased as retrieval relevance improved and reached 97.8% F1 at the selected configuration. Increasing K beyond the optimum produced a slight decline because marginally relevant signatures diluted the attention-weighted representation in Eq. (15). The analysis therefore indicates that retrieval quality is more influential than retrieval volume.

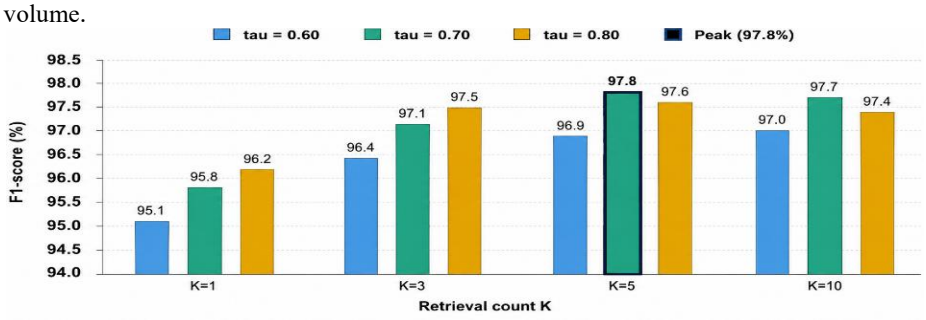


Fig. 3. Sensitivity of XAR-Detect to retrieval count $K$and similarity threshold τ

Agentic Reasoning and Explainability. *Fig. 4 (a)* shows that increasing reasoning depth improves balanced F1 from 96.1% to 97.8% and zero-shot F1 from 74.2% to 88.3%, while reducing FPR from 3.9% to 2.1%. This confirms the value of iterative evidence assessment, hypothesis formation, and decision consolidation, particularly for unseen attacks. Fig. 4(b) shows that XAR-Detect achieves higher explanation faithfulness (0.847 vs. 0.612) and stability (0.913 vs. 0.701) than the post-hoc baseline, indicating more reliable and consistent attributions.

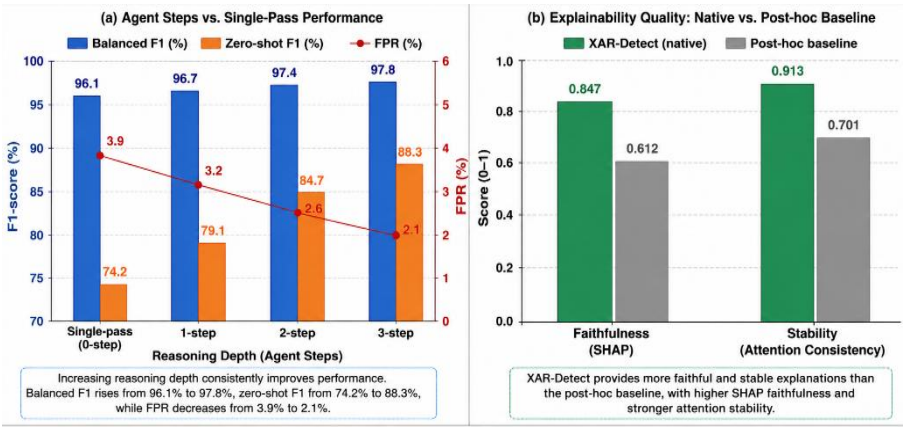


Fig. 4. Agentic reasoning and explainability analysis: ***(a)*** performance across reasoning depth and ***(b)*** faithfulness and stability relative to the post-hoc baseline.

7 Discussion, Limitations and Future Work

The results in Section 6 show that retrieval-augmented grounding, agentic multi-step reasoning, and dual-layer explainability each contribute to XAR-Detect's performance and transparency. The most important finding is the 88.3% zero-shot F1 on attack classes excluded from training, as shown in *Fig. 2.* This result supports the coverage-dependent generalisation analysis in *Proposition 2* and indicates that retrieval-grounded inference improves adaptation to unseen attacks rather than only enhancing seen-class performance. The ablation results in Table 5 further demonstrate that retrieval and agentic reasoning are complementary. RAG alone improved F1 by 1.3 percentage points, while the agent alone contributed 0.9 points. Their combination produced a 3.0-point gain over the encoder-only baseline, showing that retrieved evidence expands the information available to the agent, while iterative reasoning identifies the signatures most relevant to the final decision. The explainability analysis in Fig. 4 reported faithfulness of 0.847 and stability of 0.913, indicating that the native explanation mechanism provides more reliable and consistent attributions than post-hoc SHAP applied to the static baseline. These findings support both predictive performance and audit-oriented decision analysis.

Limitations. First, zero-shot performance depends on the knowledge-based coverage density ρ. As formalised in *Proposition 2,* attack classes without sufficiently similar signatures in $\mathcal{K}$ may approach the worst-case error $\mathcal{E}_{max}$. Although the feedback loop in *Fig. 1* supports periodic knowledge expansion, entirely novel attack families remain challenging. Moreover, because some benchmark samples predate the most recent threat intelligence available through the RAG knowledge sources, the zero-shot results should be interpreted as detection of attack classes excluded from model training rather than as evidence that every evaluated threat was previously undocumented in external repositories. Evaluation on newly emerging threat samples would provide a stronger test of performance against genuinely novel attacks.

Second, the reasoning module relies on GPT-4o, introducing API dependency and inference latency that may restrict deployment in high-throughput environments. These operational costs are not captured by the reported F1 and FPR results. External API calls may also introduce usage costs that increase with query volume and reasoning depth. In privacy-sensitive environments, transmitting email content, HTTP-request information, or derived representations to an external reasoning service may require additional data-minimisation, access-control, and organisational compliance measures. Operational deployment would therefore need to balance detection performance against latency, cost, privacy requirements, and the level of automation appropriate for the security workflow. Rather than replacing analysts, XAR-Detect can be integrated as decision support in which detections, retrieved evidence, and explanations are presented for analyst verification and escalation**.**

Third, explanation faithfulness and stability were evaluated quantitatively, but their practical usefulness to security analysts has not yet been assessed through a human-subjects study. Claims about operational interpretability should therefore remain limited to the technical evidence reported in Section 6. Accordingly, the present results establish quantitative explanation faithfulness and stability, but do not by themselves demonstrate improved analyst trust, usability, or decision quality.

Future Work. Automated threat-signature extraction from incident logs, threat feeds, and dark-web sources could expand $\mathcal{K}$and increase ρ. Future evaluation should also incorporate recently observed and time-isolated threat samples to distinguish training-unseen attacks from threats that may already be represented in external intelligence sources. Lightweight local models such as LLaMA-3 or Mistral could reduce latency and remove dependence on external APIs. Such local deployment may also reduce recurring API costs and provide greater control over privacy-sensitive security data. The federated-ready design introduced in *Section 4* could be extended to support collaborative threat-knowledge updates without sharing raw email or HTTP data. *Finally*, structured analyst studies should examine whether the explanation output reduces triage time, false escalation, and investigation effort in operational security environments.

8 Conclusion

This paper presented XAR-Detect, an explainable agentic RAG framework that integrates DeBERTa-v3 encoding, live threat-intelligence retrieval, multi-step LLM

reasoning, and dual-layer explainability for phishing and web attack detection. The theoretical analysis linked unseen-class performance to retrieval relevance and knowledge-base coverage. Experiments across the Nazario, Enron, Subhadeep, and SR-BH benchmarks achieved 97.8% F1, 2.1% FPR, and 88.3% zero-shot F1. The explanation module further achieved faithfulness of 0.847 and stability of 0.913. These findings show that XAR-Detect provides a label-efficient, knowledge-grounded, and transparent foundation for adaptive cyber threat detection, while practical deployment remains subject to knowledge-based coverage, API-related constraints, and further validation with security analysts.

Acknowledgment. The authors gratefully acknowledge the support and research environment provided by the Department of Computing, ML and Software Engineering, University of Turku, Finland, AUIB, Bangladesh, Xiamen University Malaysia and Islamic University of Madina.

Funding. The authors received no external funding for this research.

Ethics Declaration. This study used publicly available cybersecurity benchmark datasets and threat-intelligence resources and did not involve the recruitment of human participants, animal experiments, or the collection of new personally identifiable information by the authors.

Declaration of Generative AI Use. The authors used generative AI tools solely for language refinement and improving clarity. All scientific content, including the methodology, experimental design, data analysis, figures, and results, was developed, critically reviewed, verified, and edited by the authors.

Declaration of Competing Interest. The authors declare that they have no known competing financial interests or personal relationships.

Data Availability. The datasets used in this study are publicly available from their respective repositories and original sources, as cited in the manuscript. Data and supporting materials related to the study are available from the corresponding author upon reasonable request.

References

1. Anti-Phishing Working Group, “Phishing Activity Trends Report, 4th Quarter 2024,” Mar. 2025. Available: https://www.apwg.org/
2. “K12 SIX Contributes to Verizon 2025 Data Breach Investigations Report,” K12 SIX. Accessed: Jul. 14, 2026. [Online]. Available: https://www.k12six.org/news/k12-six-contributes-to-verizon-2025-data-breach-investigations-report
3. J. Zorabedian, “Surging data breach disruption drives costs to record highs | IBM.” Accessed: Jul. 14, 2026. [Online]. Available: https://www.ibm.com/think/insights/whats-new-2024-cost-of-a-data-breach-report
4. R. Jabir, J. Le, and C. Nguyen, “Phishing Attacks in the Age of Generative Artificial Intelligence: A Systematic Review of Human Factors,” *AI*, vol. 6, no. 8, p. 174, Jul. 2025, doi: 10.3390/ai6080174.
5. “Directive (EU) 2022/2555 on measures for a high common level of cybersecurity across the Union, amending Regulation (EU) No 910/2014 and Directive (EU) 2018/1972, and repealing Directive (EU) 2016/1148 (NIS 2 Directive) – European Sources Online.” Accessed: Jul. 14, 2026. [Online]. Available: https://www.europeansources.info/record/proposal-for-a-directive-on-measures-for-a-high-common-level-of-cybersecurity-across-the-union/
6. 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, Feb. 2023, doi: 10.1016/j.jksuci.2023.01.004.

7. S. Kavya and D. Sumathi, “Staying ahead of phishers: a review of recent advances and emerging methodologies in phishing detection,” *Artif Intell Rev*, vol. 58, no. 2, p. 50, Dec. 2024, doi: 10.1007/s10462-024-11055-z.
8. P. N. Yeboah, A. S. M. Kayes, W. Rahayu, E. Pardede, and S. Mahbub, “A framework for phishing and web attack detection using ensemble features of self-supervised pre-trained models,” *Journal of Network and Computer Applications*, vol. 250, p. 104474, Jun. 2026, doi: 10.1016/j.jnca.2026.104474.
9. P. He, X. Liu, J. Gao, and W. Chen, “DeBERTa: Decoding-enhanced BERT with Disentangled Attention,” Oct. 06, 2021, *arXiv*: arXiv:2006.03654. doi: 10.48550/arXiv.2006.03654.
10. R. K. Singh, M. K. Sachan, and R. B. Patel, “Cross-domain sentiment classification using decoding-enhanced bidirectional encoder representations from transformers with disentangled attention,” *Concurrency and Computation*, vol. 35, no. 6, pp. 1–1, Mar. 2023, doi: 10.1002/cpe.7589.
11. F. Blefari, C. Cosentino, F. A. Pironti, A. Furfaro, and F. Marozzo, “CyberRAG: An agentic RAG cyber attack classification and reporting tool,” *Future Generation Computer Systems*, vol. 176, p. 108186, Mar. 2026, doi: 10.1016/j.future.2025.108186.
12. K. M. M. Uddin, N. Biswas, S. T. Rikta, Md. Nur -A-Alam, and R. Mostafiz, “Explainable Machine Learning for Phishing Site Detection: A High-Efficiency Approach Using Boosting Models and SHAP,” *The Journal of Engineering*, vol. 2025, no. 1, p. e70110, Jan. 2025, doi: 10.1049/tje2.70110.
13. B. V. Pavani, D. Mahitha, and B. U. Maheswari, “Enhancing Online Safety: Phishing URL Detection Using Machine Learning and Explainable AI,” in *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, Kamand, India: IEEE, Jun. 2024, pp. 1–6. doi: 10.1109/ICCCNT61001.2024.10723976.
14. R. Alsakarnah, M. Z. Masoud, and A. Ghababsheh, “Hybridizing Explainable AI (XAI) for Intelligent Feature Extraction in Phishing Website Detection,” *Electronics*, vol. 15, no. 2, p. 350, Jan. 2026, doi: 10.3390/electronics15020350.
15. S. Khan, “Human in the loop reinforcement learning framework for adaptive phishing detection,” *Discov Computing*, vol. 28, no. 1, p. 311, Dec. 2025, doi: 10.1007/s10791-025-09849-y.
16. K. Evans *et al.*, “RAIDER: Reinforcement-aided Spear Phishing Detector,” 2021, doi: 10.48550/ARXIV.2105.07582.
17. H. Wang and B. Hooi, “Automated Phishing Detection Using URLs and Webpages,” 2024, *arXiv*. doi: 10.48550/ARXIV.2408.01667.
18. T. Cao *et al.*, “PhishAgent: A Robust Multimodal Agent for Phishing Webpage Detection,” *AAAI*, vol. 39, no. 27, pp. 27869–27877, Apr. 2025, doi: 10.1609/aaai.v39i27.35003.
19. W. Li, S. Manickam, Y. Chong, and S. Karuppayah, “PhishDebate: An LLM-Based Multi-Agent Framework for Phishing Website Detection,” 2025, *arXiv*. doi: 10.48550/ARXIV.2506.15656.
20. T. Koide, N. Fukushi, H. Nakano, and D. Chiba, “ChatSpamDetector: Leveraging Large Language Models for Effective Phishing Email Detection,” 2024, *arXiv*. doi: 10.48550/ARXIV.2402.18093.
21. M. S. Islam, L. G. Jaimes, and A. S. Dina, “MA-IDS: Multi-Agent RAG Framework for IoT Network Intrusion Detection with an Experience Library,” 2026, *arXiv*. doi: 10.48550/ARXIV.2604.05458.
22. Elements of Information Theory, 2nd Edition | Wiley,” Wiley.com. Accessed: Jul. 14, 2026. [Online]. Available: https://www.wiley.com/en-dk/elements-of-information-theory-2nd-edition-p-9780471241959
23. N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: Synthetic Minority Over-sampling Technique,” *jair*, vol. 16, pp. 321–357, Jun. 2002, doi: 10.1613/jair.953.
24. J. Johnson, M. Douze, and H. Jegou, “Billion-Scale Similarity Search with GPUs,” *IEEE Trans. Big Data*, vol. 7, no. 3, pp. 535–547, Jul. 2021, doi: 10.1109/TBDATA.2019.2921572.
25. S. Lundberg and S.-I. Lee, “A Unified Approach to Interpreting Model Predictions,” 2017, *arXiv*. doi: 10.48550/ARXIV.1705.07874.
26. N. Gkouti, G. Paraskevopoulos, and A. Potamianos, “Should I try multiple optimizers when fine-tuning pre-trained language models?” in *Proc. 18th Conf. European Chapter of the Association for Computational Linguistics (EACL)*, 2024, pp. 2657–2671.