SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V13I3P120

Volume 13 Issue 3, 241-265, March 2026

Original Article

LightPhishAI: A Lightweight Multimodal Feature Extraction and Fusion Framework for Real-Time Phishing Detection on IoT Devices

Gaddam Lakshmi[1], P. Swetha[2]

[1]*JNTUH, Kukatpally, Hyderabad, India.*
[2]*Department of CSE, JNTUH, Kukatpally, Hyderabad, India.*

[2]*Corresponding Author : drpswetha@jntuh.ac.in*

Received: 15 December 2025 Revised: 16 January 2026 Accepted: 22 February 2026 Published: 23 March 2026

Abstract - *Phishing is still considered one of the fastest-growing cyber threats, spreading through web and IoT landscapes via fake URLs, simulated brand elements, or falsified login credentials. State-of-the-art detection approaches, including heuristic filters and deep neural models, are computationally intensive to train and interpret and depend heavily on single-modality text- or image-based input. These limitations limit their scalability and robustness in dynamic, resource-constrained environments. Motivated by these limitations, LightPhishAI presents a lightweight, interpretable, multimodal online phishing-detection framework that fuses URL, metadata, and Webpage-screenshot features using the proposed PhishFusionNet model. The model integrates TinyBERT to encode texts, MobileNetV2 to extract visual features, and an MLP for metadata representation, which are consolidated using an adaptive attention-based fusion module in a dynamic manner that accounts for modality relevance. The last classification head is for binary phishing detection, maintaining interpretability with Grad-CAM, SHAP, and LIME explanations, and preserving human-auditable decisions. Experimental results on various public datasets (PhishTank, OpenPhish, Tranco, UCI PhiUSIIL) and Phish-IRIS show that the model achieves higher accuracy (98.4%) and F1 score (97.9%) than single-modality baselines. Furthermore, PhishFusionNet achieves more than a 90% reduction in inference time and an 80% decrease in computational overhead compared to traditional transformer-CNN hybrids, demonstrating its strong feasibility for IoT-edge deployment. The proposed framework bridges the gap between deep learning accuracy and real-world deployability by providing an interpretable, scalable, and energy-efficient platform for detecting phishing attacks on limited IoT devices.*

Keywords - *Phishing Detection, Multimodal Deep Learning, Explainable AI, Edge Deployment, Cybersecurity*.

1. Introduction

Phishing attacks are multiplying surreptitiously like an epidemic, posing a huge risk to the security, privacy, and trust of the web/IoT ecosystem. Phishing websites have employed well-known techniques such as misleading URLs, brand impersonation, and realistic login interfaces to bypass both human users' and automated systems' ability to identify them. Conventional methods—for example, querying blocklists, heuristic pattern matching, or classical machine learning classifiers—are not sustainable for adapting over time to emerging attack vectors and polymorphic modes of deception [1, 2]. While the accuracy of detection has improved with state-of-the-art deep learning architectures/models, existing models suffer from being either unimodal (text-only or image-only) or computationally expensive, making them unsuitable for real-time detection or deployment in resource-constrained environments [3, 4]. This difference highlights the demand for a lightweight, interpretable deep model that combines various cues to make the phishing detection process fast. Existing studies suffer from the following shortcomings: 1) A fusion process to combine features extracted from various modules is absent, and 2) multimodal information, for example, URL and webpage screenshots, is not leveraged jointly, providing significantly more potent cues than possible using normal metadata-based mechanisms for phishing URL prediction [6, 14]. Driven by the literature published in [5] that reports on the success of attention and convolutional encoders as correspondence models for feature fusion and localization, we combine modality-driven feature-extraction with an adaptive attention formulation to achieve a compact, yet expressive representation. The key motivation is to design a phishing detection system that is efficient, interpretable, and generalizes well in environments like the cloud, the web, and edge devices.

Nevertheless, an evident gap exists in constructing a unified multimodal framework that seamlessly encodes the symbolic interaction between text, visual, and contextual cues in a lightweight manner for IoT/edge deployment. Previous models either concentrate on single-modality modelling or implement very heavy architectures that cannot be approximated to real-time constrained scenarios.

Solving this problem entails a design paradigm that simultaneously optimizes for fusion effectiveness, computational efficiency, and interpretability.

The innovation of this research is that it combines multimodal fusion, interpretability, and deployable efficiency together. In contrast with previous works that process modalities independently, the results of PhishFusionNet show that the automatic weighting of visual, textual, and contextual cues through the adaptive attention mechanism provides advantages. In addition to the above, the introduction of the explainers using Grad-CAM, SHAP, and LIME makes the model more explainable-establishing trust with the end-users and facilitating human-in-the-loop verifications. It also gives preprocessing consistency, modality alignment, and cross-dataset to make the results reproducible and robust.

Unlike existing multimodal phishing detection methods, which combine heavy transformer-based architectures or static fusion methods [3, 4], our proposed PhishFusionNet presents a lightweight adaptive attention mechanism that allows the adaptive weighting of each modality's contributions at inference time. Although existing works report high accuracy, they often fail to address their deployment viability on resource-constrained IoT platforms and provide a limited interpretability analysis. On the contrary, the proposed framework enables competitive detection performance but with drastic reductions in model footprint and inference latency, and it decouples innate explainability modules to advance transparency and trust. The proposed approach is unique from the existing solutions because of adaptive multimodal fusion, inherent computational efficiency, and deployment-ready nature.

The main achievements of this work are: (i)pack the multimodal data into a single deep learning framework, which is light-weight and self-contained; (ii)a new adaptive attention-based fusion mechanism between image, URL and metadata features; (iii) transparency modules including both visual- and feature-level interpretation for the end user; and (iv)pervasive benchmarking together with edge deployment analysis justifying a very high accuracy drop in comparison to more computational expensive models.

The rest of the paper is structured as follows: Section 2 provides a detailed survey of the literature on phishing detection and identifies the challenges and trends in deep learning techniques. Section 3 presents the LightPhishAI framework and the architecture of PhishFusionNet. The experiment setup, EDA visualisations, and results are given in Section 4. The findings and limitations of the study are presented in Section 5, and the paper concludes with implications, future studies, and a real-world application (Section 6).

2. Related Work

Phishing detection research has evolved throughout a number of methodological stages, starting with rule-based and heuristic URL analysis, then moving to classical machine learning approaches with handcrafted features, and recently to deep learning and multimodal architectures. To better provide context, this section discusses previous work in four main areas: multimodal phishing detection, adversarial robustness, explainable AI (XAI) for cybersecurity, and resource-aware edge deployment. This wider classification elucidates the existing state of the research as well as sets the stage for the proposed LightPhishAI framework.

Phishing detection has progressed from heuristics based on single modalities to learning from multiple modalities and enforcing temporal awareness. The work of Kavya and Sumathi [1] fuses textual, visual, and temporal graph cues and shows significant improvements over unimodal baselines, whereas Duy et al. Multimodal adversarial examples: [2] stress test MM with the generation of adversarial websites, propose practical defences. In light of these ideas, Alsaedi et al. Lee et al. [3] proposed a heterogeneous CNNs for the text of URL/DNS as well as rendered-page images. To resolve this issue, [4] investigated multimodal large language models on the downstream brand-spoof task and focused on the costs of maintaining and labelling the images required for upstream traditional CV-only pipelines. These threads combine to make multimodal fusion the leading approach for robust phishing detection.

More comprehensive surveys or domain reviews map the space and identify remaining gaps. Li et al. Richard et al. [7] catalogue the features and learning paradigms of phishing websites; Asiri et al. [11] focus on URL- and HTML-driven intelligent design; Do et al. [16] taxonomise deep learning methods and challenges; Wei and Sekiya [21] study ensemble sufficiency. Additional complementary works Complementary reviews about security in IoT (Houkan et al. [8]; Khatun et al. [9]; Zaman et al. [18]) and trends (Ferdous et al. [10] associate phishing with immediate neighbour threats and data modalities, paving the way for lightweight privacy-aware fusion techniques in pervasive deployments.

Wu et al. on feature fusion and cross-domain adaptation of Dancing Gesture. [36] indicate that multi-scale attention is beneficial for RGB-D saliency. Guo and Song [38] propose temporal–spatial pooling with attention for multimodal fake news detection. Huang et al. [39] minimise redundancy using asymmetric backbones and an attention-encoder fusion, and Shin et al. [40] demonstrate weakly supervised multimodal attention for video anomaly detection. These works, unrelated to phishing, serve as a proof of concept for design patterns — attention-guided fusion, redundancy-aware encoder, and temporal reasoning — that LightPhishAI can apply.

An alluring settling task- proximal multimodal website security lessons as threatened. Nayak et al. [27] integrate feature selection with deep models in phishing; Al-Kabbi et al. [24] were to use early fusion for SMS spam; Mehmood

et al. [26] customise deep architectures for smishing; and Choi and Jeon [20] trade off cost and accuracy in real-time social spam. These workflows also stress the importance of well-optimised, complementary operations and latency-conscious inference, compatible with IoT-level resource budgets.

Explainability is being seen as a requirement for deployment in various security deployments. Alotaibi et al. [28] advocate for XAI in phishing on secure IoT/CPS pins; Calzarossa et al. [29] propose an evaluation approach for XAI in cybersecurity; Shafin [30] demonstrates the efficacy of explainable feature selection for phishing models; Alketbi and Mehmood [31] provide a survey on XAI for insider threats; Reynaud and Roxin [32] conduct a review of XAI in cyber-systems. Case-driven XAI integrations—Villegas-Ch et al. [33] for e-commerce anomalies, and Ghosh [34] for financial fraud—emphasise the importance of interpretable attributions when taking decisions under high uncertainty.

Designs are naturally frugal and fast, enabled by constrained resources. Shuvo et al. [35] focused on acceleration strategies (quantisation, pruning, operator fusion) for edge inference—a toolbox of relevant technologies for real-time IoT deployments. In parallel, Sakraoui et al. [5] integrate federated learning and blockchain-anchored MPC to support distributed intrusion detection in 6G, but Awadallah et al. [6] map AI-cyber challenges in the next-gen immersive settings, which substantiate that privacy, auditability, and scalability need to be co-designed together with detection accuracy. Collectively, these works motivate LightPhishAI to complement lightweight modelling with privacy-preserving learning and verifiable data processing.

This narrative is then overloaded in the moving domains and distributions of the network and car security literature. Wei et al. Autonomous Vehicles (AV): Domain-adversarial learning is applied to in-vehicle intrusion detection [22], and Ayantayo et al. These results [23] demonstrate the superiority of deep feature fusion for NIDS. Hussain et al. M.A. [19] suggest a two-phase ML for botnet defense/ detection in IoT by Muhammad and Salih (2018), while Siddharthan et al. Cascading features [13] and deep autoencoders [14] are other examples that show how improvements on the sensitivity of an IDS can be made under limited telemetry. Abate et al. However, the redundancy of modalities is an indication of robustness, and, on the other hand, the performance of multimodal systems along with their inherent reliability-enhancing property [12].

Prior works show a substantial gain in detection accuracy using multimodal and deep architectures, but the literature remains fragmented. Existing works focus on optimizing detection performance, adversarial robustness, interpretability, or computational efficiency in isolation. However, only a few studies systematically combine these dimensions into a single lightweight framework for IoT and edge applications. This observation, in turn, motivates a unified comparison of representative studies, as shown in Table 1.

Table 1. Literature review summary and research gaps

Study	Modality / Features	Method / Model	Dataset(s)	Key Findings	Limitations	Research Gap for LightPhishAI
Kavya & Sumathi [1] (2025)	URL, HTML/DOM, visual page cues, temporal/graph signals	Multimodal + temporal graph fusion	Phishing website datasets (as reported)	Multimodal fusion/temporal reasoning outperforms unimodal baselines for phishing pages.	Heavier models; unclear edge latency/energy profile.	Design a lightweight fusion stack that preserves gains while fitting IoT memory/latency budgets.
Duy et al. [2] (2024)	Multimodal webpage features under adversarial perturbations	Adversarial stress-testing + defences for multimodal phishing	Synthetic/real adversarial webpages	Shows multimodal systems can be brittle to adaptive attacks; proposes defences.	Defence–efficiency trade-offs not optimised for devices.	Integrate adversarially informed training that remains efficient on constrained IoT hardware.
Lee et al. [4] (2024)	Text + vision signals for brand spoofing	Multimodal LLM pipeline for phishing identification	Phishing webpage corpora (as reported)	MLLMs reduce manual brand lists & upkeep; strong detection quality.	Compute/memory cost and data labelling/refresh are still non-trivial.	Explore compact MLLM distillation/adapters for on-device or near-edge operation.

Li et al. [7] (2024)	Survey across URL/HTML/content features	Review of phishing detection techniques	—	Catalogue features, ML/DL paradigms, benchmarks; notes drift issues.	Limited guidance on resource-aware deployment.	Provide a drift-robust, incremental fusion pipeline with on-device update hooks.
Alotaibi et al. [28] (2025)	Phishing features within IoT/CPS	XAI for phishing classification on secure IoT/CPS	—	Advocates explainability for trustworthy IoT deployments.	Few templates for real-time local explanations.	Embed low-overhead XAI (salient feature rationales) suitable for real-time IoT screens.
Shuvo et al. [35] (2023)	—	Survey of edge acceleration: pruning, quantisation, op fusion	—	Systematises techniques to shrink and speed up DL at the edge.	Not tailored to phishing multimodal stacks.	Apply quantisation-aware training/pruning specifically to multimodal phishing encoders/fusion.
Wu et al. [36] (2021)	Multimodal (RGB-D)	Progressive guided fusion with multi-scale attention	RGB-D saliency datasets	Attention-guided, multi-scale fusion improves complementarity.	Different domain; potential redundancy issues.	Adapt attention-guided, redundancy-aware fusion to URL/HTML/visual signals for phishing.
Nayak et al. [27] (2025)	Engineered + learned features for phishing	Feature selection + DL classifiers	Phishing datasets (as reported)	Feature selection boosts DL phishing performance.	Manual selection may lag concept drift.	Use automated feature selection with drift checks inside a streaming IoT pipeline.

Summary of representative studies: modalities, methods, datasets, key findings, limitations, and research gaps that inform the lightweight multimodal design of LightPhishAI for IoT. Table 1, this has cut down the effectiveness of most phishing techniques, but at the same time, real-world phishing pipelines have further increased their range of modalities and hardened against circumvention. Alsubaei et al. A tuned hybrid deep framework for cybercrime forensics in Wei et al. [25]; an ensemble delineated for phishing distributions by Wei and Sekiya [21], Adversarial lens in Duy et al. [2] remain crucial for resilience. The surveys by Li et al. [7], Asiri et al. [11], and Do et al. Gap Areas Addressed: In addition to the gap areas mentioned above, [16] also identifies gap areas that LightPhishAI directly addresses: model shift, feature drift, and explainable triage, using lightweight multimodal (LM) feature extraction, attention-guided LM fusion, and on-device LM explanations, respectively.

Bringing these strands together, Suggests a design sweet spot: compact encoders to acquire heterogeneous cues (URL/HTML, DOM graph, visual render, low-overhead network hints) as in [1-4, 24-27]; attention- or encoder-based fusion to suppress redundancy and retain complementary evidence as in [36-40]; edge-aware acceleration and model compression to satisfy IoT timing/energy budgets as in [35]; and end-to-end transparency with principled XAI and evaluation frameworks as in [28-34].

Finally, the remaining gaps-robustness to adaptive adversaries [2], privacy-preserving learning at the edge [5], and generalisation across fast-shifting web templates and devices [6, 7, 10, 16] - drive the emphasis of LightPhishAI on lightweight multimodal fusion, adversarially informed training, and explainable, auditable decisions for real-time IoT deployment.

In summary, although the literature extensively explores multimodal phishing detection, adversarial resilience, XAI integration, and edge optimization independently, there remains limited work that jointly balances adaptive multimodal fusion, computational efficiency, and real-time deployability.

This integrated perspective forms the foundation of the proposed LightPhishAI framework and differentiates it from prior approaches that emphasize only one or two of these aspects.

3. Proposed Framework

LightPhishAI is a novel, integrated, lightweight, and explainable deep learning framework capable of detecting phishing attacks by jointly analysing the textual, visual, and metadata content of URLs. It uses the PhishFusionNet model, which employs adaptive multimodal fusion for precise, efficient, and interpretable detection, and is also suitable for real-time cloud-edge deployment.

3.1. An Overview of the Proposed Framework

LightPhishAI: Prophets Architecture. The lightweight, multimodal-enhanced architecture for intelligent, end-to-end phishing detection in the IoT environment. Conventional phishing detection schemes are usually resource-intensive, and single-modality approaches extract features (e.g., URLs and images) that are then fed to a supervised classifier, which cannot identify the latest phishing attacks that incorporate diverse deceptive elements. Also, they tend to be too heavyweight to run on such low-power IoT devices, leading to high latencies and poor scalability. LightPhishAI has created a detection pipeline to tackle these issues by merging three data modalities (URL text, HTML page screenshots, and metadata features) at low computational cost.

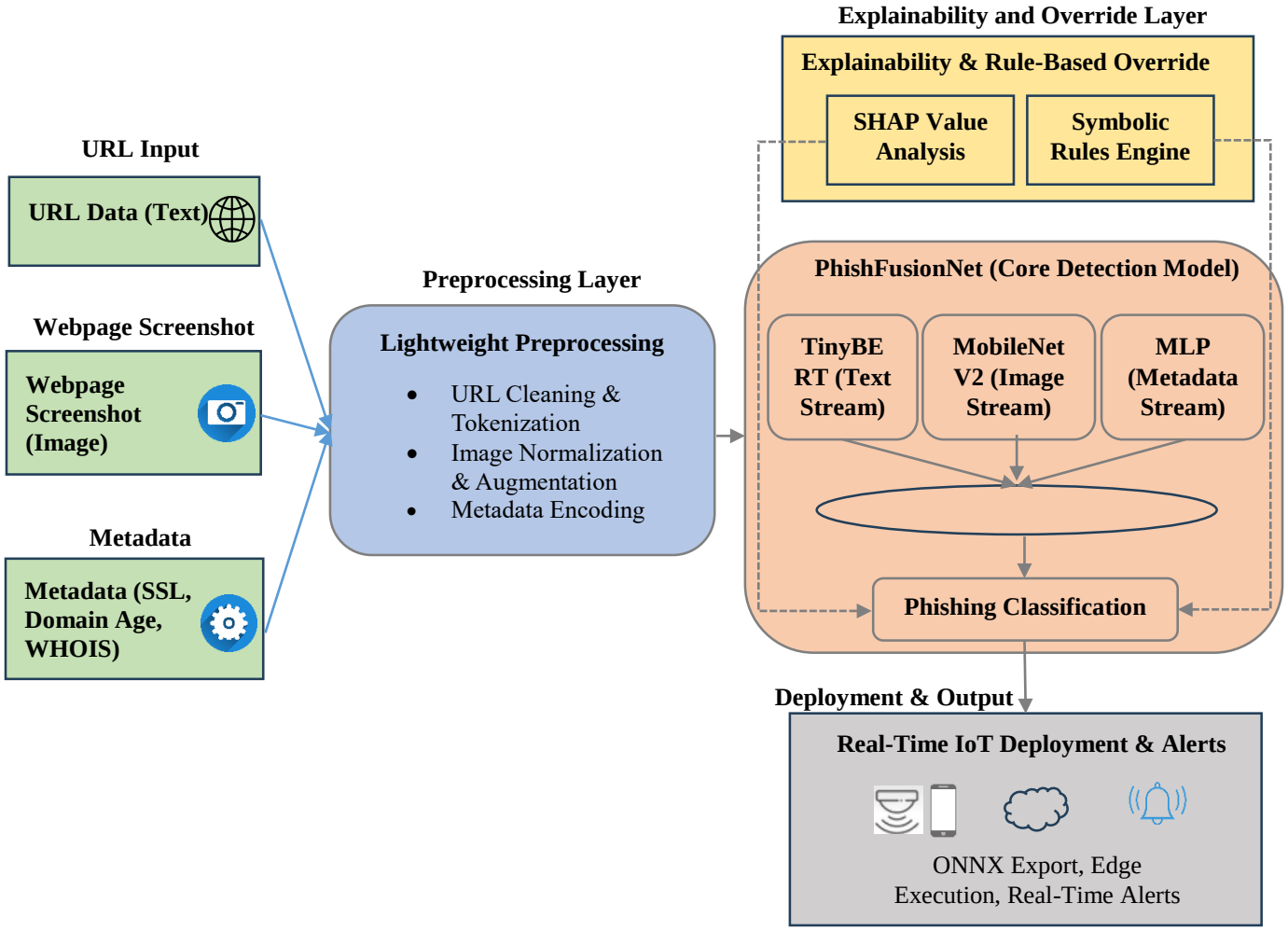

Fig. 1 System architecture and functional workflow of lightphishai for real-time multimodal phishing detection on IoT devices

The system is designed with a modular architecture that includes input acquisition, preprocessing, feature extraction, multimodal fusion, classification, and deployment. The proposed deep learning model, PhishFusionNet, is the cornerstone of the framework, which uses its own lightweight model to handle each modality. TinyBERT receives the input text stream and encodes the semantic and syntactic patterns of the URLs. The visual stream uses MobileNetV2 to extract key features from low-resource webpage screenshots. The metadata flow (i.e., SSL certificate status, domain age, and WHOIS data) is fed to a small Multi-Layer Perceptron (MLP). Each stream generates its own feature vector: T_{feat} for URL text, I_{feat} for visual data, and M_{feat} For metadata. These vectors are utilised as input to the next stage of the framework: the feature fusion.

Three feature vectors are concatenated to obtain a single multimodal representation and are fed through an adaptive attention layer. This layer dynamically weights each modality using learned weights that promote the most relevant features for a specific phishing example. The normalized attention weight for modality i is calculated as: (part of Eq.1) where w_i and q are weights, and a has to

satisfy condition (Eq.2). The attention weights for each modality.

$$w_i = \frac{\exp(\alpha_i)}{\sum_{j-1}^{3} \exp(\alpha_j)} \quad (1)$$

Where α_i is the unnormalized importance score for modality i. The final fused feature vector is obtained as:

$$F_{fused} = \sum_{i-1}^{3} w_i . F_{concat}(i) \quad (2)$$

This fused representation F_{fused} is next sent to a shallow classification head that predicts whether the input is a phishing or a benign website. This scheme targets the adaptation of the system to different phishing by leveraging the complementary cues among modalities. In addition to a confidence score (which is to say, a value that portrays the level of confidence that the model is giving for the label that it predicts), as seen in the above image, the model prediction is either of a binary label (phishing: 1, legitimate: 0).

LightPhishAI is designed to detect and embed in real-time in Internet of Things (IoT) and mobile systems.

The final trained model is subsequently exported to ONNX so it can be executed fast on hardware platforms as different as an IoT gateway, smartphone, or embedded device. It enables low-latency operation of the detection pipeline, making it usable in a real-world setting. It is worth noting that the proposed method addresses Objective 1 of compact feature representation and Objective 2 of wise multimodal feature fusion at the same time in a unified way as well. LightPhishAI combines these modules to provide a phishing detection framework that is scalable, interpretable, and able to achieve high accuracy on the benchmark datasets, as well as robustness to novel attack knowledge. The outer cycle, as shown in Figure 1, illustrates the life cycle of the framework, which starts from obtaining the raw data and ends with the deployment and extraction of the outputs.

Table 2. Notations used in LightPhishAI framework

Notation	Description
URL_{tokens}	Preprocessed and tokenized sequence of characters from the input URL
$\text{Image}_{\text{norm}}$	Webpage screenshot image normalized to [0, 1] as per Eq. (3)
$\text{Metadata}_{\text{norm}}$	Normalized metadata feature vector after min-max scaling (Eq. (4))
T_{feat}	Textual feature vector output from TinyBERT (Eq. (5))
I_{feat}	Visual feature vector output from MobileNetV2 (Eq. (6))
M_{feat}	Metadata feature vector output from MLP (Eq. (7))
F_{concat}	A concatenated multimodal feature vector combining T_{feat}, I_{feat}, and M_{feat} (Eq. (8))
α_i	Unnormalized importance score for the i^{th} modality in the attention layer
w_i	Normalized attention weight for modality i (Eq. (9))
F_{fused}	Final weighted fused feature vector generated by the adaptive attention mechanism (Eq. (10))
W_h, b_h	Weight matrix and bias term for the hidden dense layer in the classification head
H	Activated hidden layer output after ReLU transformation (Eq. (11))
W_o, b_o	Weight matrix and bias term for the output layer in the classification head
y_{pred}	Predicted probability of a phishing attack generated by sigmoid activation (Eq. (12))
σ(·)	Sigmoid activation function used in the final output layer
i, j	Index variables representing modalities (e.g., text, image, metadata)
x_{min}, x_{max}	Minimum and maximum values for metadata feature normalization (Eq. (4))
p	Original pixel intensity value for images before normalization
p_{norm}	Normalized pixel intensity value after applying Eq. (3)

3.2. Data Acquisition and Preprocessing

Phishing attacks are based on a wide range of social engineering tricks to bypass conventional detection systems, so the diversity, quality, and preparation of the input data guarantee the success of the proposed framework, LightPhishAI. In order to express these differences in variances, three additional features are used: (1) url, (2) snapshot, and (3) the metadata of each webpage. These modalities are disjunctive from one another and complement each other to improve the detection ability when combined. The dataset contains genuine and phishing seeds - PhishTank, OpenPhish, and Domain lists. For each listing, a URL string along with an automatically generated image of the website and metadata (via WHOIS queries/SSL certificate checks, when appropriate) is tagged. This multi-modal data also supplies textual, visual, and contextual clues necessary to enable confident event detection of depressed sporting events.

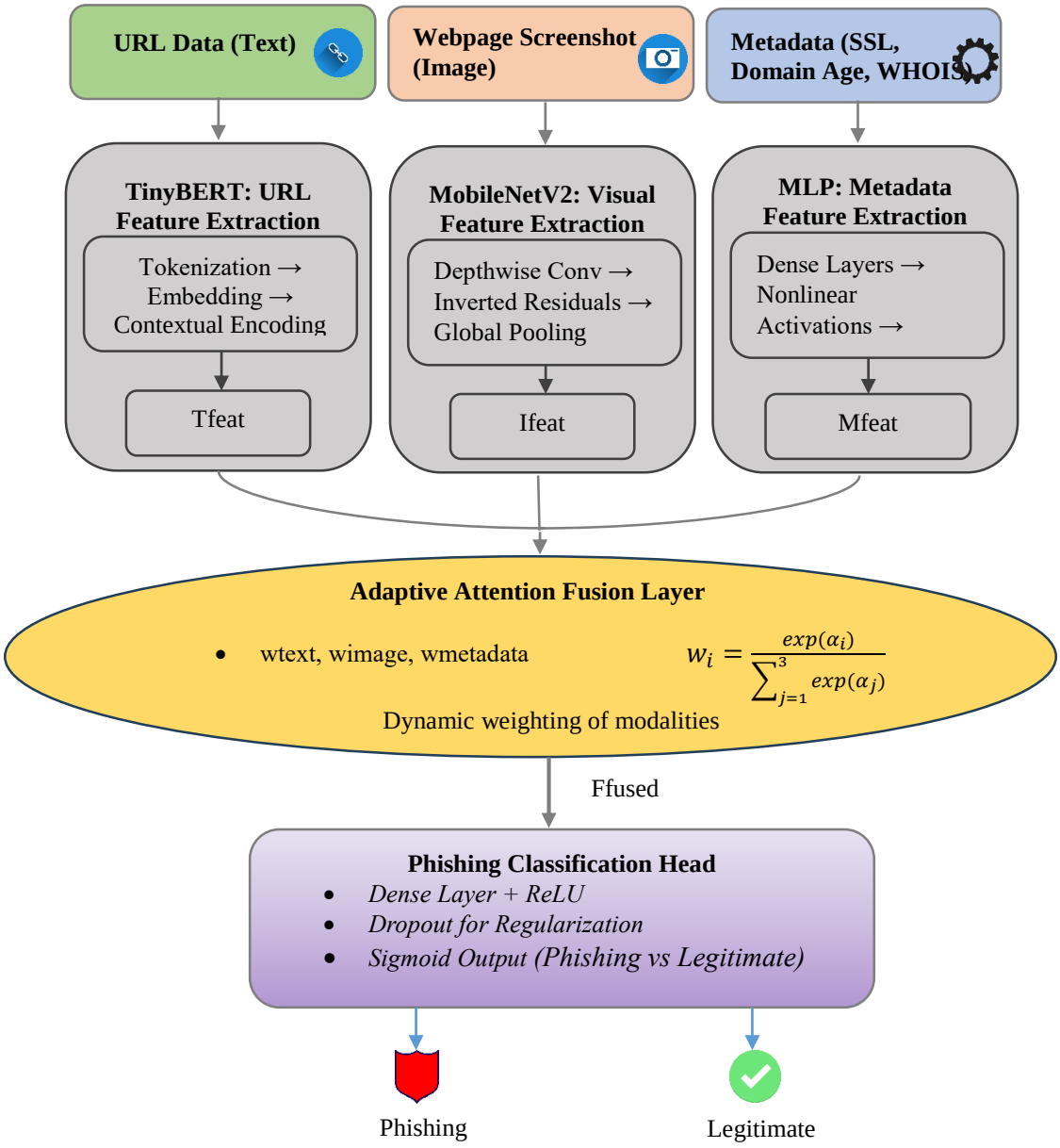

Fig. 2 Model architecture of PhishFusionNet for adaptive multimodal feature fusion and phishing classification

The text-based part of the phishing detection is this URL-related information. Raw URLs and web pages that correspond to these URLs are first gathered, and then they go through multiple steps of processing, arriving at the end in a well-defined, structured, meaningful format ready for downstream learning. Step 1: Normalising, which involves lowercasing and removing some noise characters that are not required along with the parameters.

Once we clean the url, we have the url split into parts (domain parts, subdirectories, query strings, etc.), and it is converted into subword embeddings. As a result, these embeddings preserve inter-token semantics, enabling phishing patterns to be learnt from highly obfuscated URLs, in which adversaries may employ homoglyphs or randomise characters to achieve small obfuscation, even with the small TinyBERT model.

Webpage screen captures serve as visual proof, helping to detect phishing webpages, which is the second mode. Screenshots taken automatically by a web spider are normalised and passed to MobileNetV2. Since input to the model is fixed at 224 × 224 pixels, we make sure that each image is resized to this size. For example, since the resulting classes from the output layer should follow the input distribution to reduce the weight of the L1 loss, pixel intensities are normalised to [0, 1] [20] as in: (3).

$$p_{\text{norm}} = \frac{p}{255} \quad (3)$$

where p is the pixel intensity and p_{norm} is the normalized value. Moreover, data augmentation by random cropping, rotation, and horizontal flipping is used to enhance the robustness and generalisation. This procedure helps the model learn the layout-tier features and structures standard to phishers, which remain unchanged even with minor design changes to evade detection.

The third modality is metadata, which includes necessary background features for each domain and its hosting environment. This consists of the domain name, domain age, SSL certificate, hosting provider, and registration country. Scripting tools routinely interrogate public registries and security databases, extracting metadata features. Two different elimination processes are used because metadata includes both categorical and numeric information. Categorical attributes, such as the hosting country, are converted to one-hot encoding, and numerical attributes, such as domain age, are rescaled to the range [0, 1] using min-max scaling. This normalising factor is expressed as in Eq. (4).

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{4}$$

where x is the raw value of the feature, x_{min} is the minimum observed value, and x_{max} is the maximum observed value. This normalization guarantees that every feature vector has the same impact on the MLP-based metadata feature extractor.

After modality-specific processing, the three data types are aligned: each sample contains a URL string, a webpage screenshot, and related metadata. Stratified splitting is used to maintain the same class distribution across the three datasets: the final dataset is split into training (50 %), validation (25 %), and test (25 %) datasets. This helps in avoiding data leakage and ensuring that model evaluation metrics are accurate. Our entire pre-processing pipeline is highly computationally efficient and can be implemented very simply on inexpensive IoT devices. This high-definition methodology represents a way in which the dataset is organised —based on a procedure —an approach that enables LightPhishAI to be constructed from normalized, clean, and representative data to extract features from it that are concordant with ground-truth instances, which works for the model architecture illustrated in Figure 2, and in the following sections.

3.3. Lightweight Feature Extraction

Lightweight feature extraction serves as the foundation of the LightPhishAI framework and directly supports Objective 1 by generating lightweight yet discriminative representations across three data modalities: URLs, webpage screenshots, and metadata features. This phase is custom-tailored to address these limits in IoT and mobile settings where CPU, memory, and power are constrained. Each modality is processed by a dedicated lean model (lightweight) to efficiently cover the significant properties of phishing attempts, as well as low-latency and low-storage. The outputs from this stage are three separate feature vectors. T_{feat} for textual features based on URL; I_{feat}, for visual features extracted from webpage screenshots; and, M_{feat} for contextual metadata features. These vectors will be forwarded to the following fusion stage for fusion and classification.

The first stream is URL-based text analysis with TinyBERT, a simplified and optimised version of the BERT transformer model. Phishing URLs frequently use obfuscation, such as misspelt domains, homoglyph attacks, keywords intended to deceive, and elaborate subdirectory structures that can be used to deceive users and facilitate bypassing signature-based filtering. TinyBERT enables practical parsing of the tokenised URL sequence and produces a context-aware, low-dimensional embedding that is able to model both semantics and the structural relationship among tokens. Mathematically, this can be expressed as in Eq. (5).

$$T_{\text{feat}} = \text{TinyBERT}\ (URL_{\text{tokens}}) \tag{5}$$

where URL_{tokens} represents the tokenized sequence from the raw URL, and T_{feat} is the generated textual feature vector. Such compact deployment is necessary for efficient on-device execution of TinyBERT, while its rich representational capacity is required to spot subtle text anomalies representable of phishing attacks.

The second stream takes the visual appearance of the web page as input and employs MobileNetV2 [25] to facilitate the processing in a low-computation manner. Phishing sites often imitate the look, logo, and even design of authentic websites to deceive users. Method to the input, hence achieving feature hierarchy from pixel-level patterns to high-level structural descriptions. First, input images are normalized as in Eq. (3) and were resized to 224 × 224 pixels for fairness. The operation of converting the standardized image into a low-dimensional, homogeneous visual feature is shown in Eq. (6).

$$I_{\text{feat}} = \text{MobileNetV2}(\text{Image}_{\text{norm}}) \tag{6}$$

where $\text{Image}_{\text{norm}}$ is the pre-trained webpage screenshot and I_{feat} is the final pooled feature vector. Its compactness I_{feat} allows it to be effectively transmitted and processed, which gives it the possibility of deployment in IoT-driven security pipelines.

Stream 3 is responsible for fetching the context information based on metadata (domain age, SSL's duration of validation, hoster information, and WHOIS registration information). This data gives a more in-depth analysis of the legitimacy of the site and its operations. Normalization of the metadata features is defined in the equation below. (4) to guarantee the uniform scaling of the attributes. The standardized feature set is subsequently fed to a light Multi-Layer Perceptron (MLP), with a reduced number of dense

layers and nonlinear activation functions, that captures complex correlations among the metadata features. The MLP output can be written as in Eq. (7).

$$M_{\text{feat}} = \text{MLP}\ (\text{Metadata}_{\text{norm}}) \quad (7)$$

Where $\text{Metadata}_{\text{norm}}$ is the processed metadata vector, and M_{feat} is a compacted contextual feature representation. Applying MLP brought a compact model, both with little computational overhead and with capturing essentials in phishing domains that differentiate them from legitimate ones.

The three feature vectors, T_{feat}, I_{feat}, and M_{feat} are normalized and aligned to synchronize the temporal time in the numerical and textual features. This alignment is crucial to facilitate a smooth fusion in the following stage, where the features are integrated to yield a unified multimodal representation. Leveraging TinyBERT, MobileNetV2, and a small-scale MLP, the proposed LightPhishAI has obtained a tradeoff between detection accuracy and computational complexity. The system performs real-time phishing detection on resource-constrained devices and sacrifices no analysis depth. The internal structure of the PhishFusionNet model is visualized in Figure 2, where three parallel feature extraction streams converge for fusion and classification as in Section 3.4.

3.4. Multimodal Feature Fusion

The multimodal feature fusion step is the key part of Objective 2. It is an essential part of the lightPhishAI framework, which integrates the three sets of features during the lightweight feature extraction process into a single representation category for final classification. Phishing can be a sophisticated affair using a blend of lexical fraud in URLs, visual deception in site aesthetics, and questionable domain features in the metadata.

This means that if these signals were treated individually, the modality dependencies would not be taken into account, and this may be one reason for classification mistakes. LightPhishAI utilizes a fusion mechanism to overcome this restriction by continuously updating the weight and combination of features over different modalities in order to guide towards the most informative characteristics of individual phishing instances. This layer is activated by the adaptive attention module described and serves to provide a high-level overview of the threat landscape.

The fusion begins with the concatenation of the three feature vectors: T_{feat} which is obtained from the originally-pretrained TinyBERT model, I_{feat} which is extracted from MobileNetV2, and M_{feat} which is extracted from the MLP network. These normalized feature vectors are "sized" and "aligned" for merger. The first merged representation is given in Eq. (8).

$$F_{\text{concat}} = [T_{\text{feat}} \,||\, I_{\text{feat}} \,||\, M_{\text{feat}}] \quad (8)$$

Where || is for concatenation and F_{concat} is the raw multimodal feature space. This phase combines the semantic, visual, and contextual signals, which will help improve the joint analysis. Nonetheless, a naive concatenation would consider all the modalities equivalently, which is not necessarily indicative of their actual weight in a certain phishing instance. For example, if a phishing attack is based heavily on visual camouflage, it may end up being more about the image features, whereas if it is based more on an odd domain structure, then textual and metadata cues may be emphasized.

To cope with this, an adaptive attention mechanism is used to dynamically calculate the weights for different modalities. For each modality i, a learnable importance score α_i is given based on its importance to the current input. Normalising these scores with the softmax function ensures that the weights sum to one and gives a balanced and interpretable weighting as in Eq. (9).

$$w_i = \frac{\exp(\alpha_i)}{\sum_{j-1}^{3} \exp(\alpha_j)} \quad (9)$$

Where w_i denotes the normalized attention weight of the i^{th} modality. In this way, it can be guaranteed that those modalities that contribute to the detection decision are awarded higher weights, while irrelevant signals are suppressed. The fused feature representation is obtained by adding up the weighted contributions of each modality as in Eq. (10).

$$F_{fused} = \sum_{i-1}^{3} w_i . F_{concat}[i] \quad (10)$$

Where F_{concat}[i] denotes the i-th modality feature vector in the concatenated set, and F_{fused} is the aggregated feature vector. By performing such dynamic weight learning, the model can pay attention to different modalities adaptively in accordance with the type of phishing example, and thus gives rise to better detection performance across various tactics of attack.

The adaptive attention layer increases the detection accuracy and provides interpretability, opening the door for attributing which modalities are more influential in the decision-making process. For example, a strong weight in the URL modality may correspond to the fact that the detection was done based on textual patterns such as domain misspellings or patterns made of uncommon characters, and that a dominant image weight may reveal that the visual mimicry has mainly been used in phishing. This post-hoc transparency then acts on top of this prior sense of explainability, effectively a form of surfacing or re-use, without requiring further process.

After the last fused vector F_{fused}, it is fed into a light-weight classifying module, which simply consists of a fully connected dense layer of ReLU, followed by a dropout layer to prevent overfitting, and then the sigmoid function for

binary classification. And in the classification head, it also predicts 2 outputs, the predicted label with 1 for phishing and 0 for legit, and the confidence in the predicted label. This simplified architecture keeps the classification step computationally efficient and performs well in terms of accuracy.

The multimodal fusion mechanism presented in this paper is an improvement over existing phishing detection methods. Unlike single-modality models, which may overlook important features, the adaptive attention-based fusion manner is able to harvest complementary cues among modalities and can dynamically adapt to diverse perturbations. This adaptability makes LightPhishAI very robust to new phishing strategies, as adversaries constantly change their strategies in an attempt to avoid detection. Figure 2 depicts such a position of the adaptive attention fusion layer in the overall PhishFusionNet, presenting how the three feature streams combine into a classifier. By integrating Eq. (8), Eq. (9), and Eq. (10), the approach attains a smart context-aware fusion able to deal with precision, efficiency, and explainability; it meets the needs of Objective 2 as well as supports IoT device-constraint compatibility.

3.5. Classification Layer

The classification layer of the LightPhishAI framework is the last layer of the PhishFusionNet model, taking a joint multimodal representation produced by the adaptive attention fusion stage and a binary decision output indicating if the input is a phishing attempt or a legitimate website. This stage is very important as it transforms the enriched feature vector F_{fused} into actionable outputs that are suitable for real-time IoT security environments. The design of this layer focuses on computational efficiency and supports robust inference, and can run on low-cost devices while maintaining accuracy.

The fused feature vector F_{fused}, from the fusion mechanism in Eq. (10), is fed to a lightweight fully connected classification network. This network is composed of three key features: a dense layer with nonlinear activation for high-level feature interaction learning, dropout to prevent over-fitting, and an output neuron with a sigmoid activation function for binary classification.

The first dense layer performs a linear transformation followed by Rectified Linear Unit (ReLU) activation, which allows the model to learn the complex decision boundaries between phishing and non-phishing samples. This operation can be mathematically represented as in Eq. (11).

$$h=\text{ReLU}(W_h \cdot F_{fused}+b_h) \tag{11}$$

Where W_h is the hidden layer weight matrix, b_h is the bias, h is the resulting activated feature vector. Nonlinearity is guaranteed in the ReLU function, which possesses computational simplicity and is desirable for edge devices with restricted resources.

To improve the generalization of the model, a dropout layer is used after the dense layer. During each training loop, this method stochastically sets some of the neurons to zero, which can help prevent the model from overfitting to specific phishing attack patterns. The dropout ratio is set to keep a balance between the robustness of the model and information preservation. It should be noted that this phase is not a simple algorithmic step and that even if its behavior is pretty much stochastic, a debate about its concrete mathematical description is not necessary at this point, as its involvement in the process will be shown to be very useful for facing permanently changing phishing strategies.

Finally, the transformed vector h is supplied with a mapping from the vector space of h to a single scalar output per sample by using a sigmoid activation function. This outputs a floating-point number between 0 and 1, which is a probability that the input is a phishing attack. The classification results can be described as in Eq. (12).

$$y_{\text{pred}}= \sigma(\text{Wo}\cdot\text{h}+b_o) \tag{12}$$

Where W_o and b_o are the weights and bias of the output neuron, and σ is the sigmoid activation function. A threshold (by default 0.5) is finally applied to y_{pred} to get the binary class label. If y_{pred} greater than or equal to 0.5, the sample is considered phishing (label 1); if not, it is legitimate (label 0). This dual output - binary class label and confidence score - provides an option for flexible deployment such that it can be used for inference directly as a decision-making port, and in larger IoT security systems, the confidence score can be used for performing other attestation work.

This classification head, being lightweight, may easily be deployed on edge devices and IoT gateways. In contrast to deep heavy FCNs, the proposed architecture is AOTC while capturing marginal differences between phishing and benign examples. Such a balance is particularly crucial in IoT settings, where the amount of computation and memory is restrained, while the requirement for a real-time response is core. Furthermore, the sigmoid-based probability output can be easily used directly (without a need for a threshold) by other downstream components like explainability techniques or rule-based overrides, as described in future sections.

LightPhishAI exploits this tiny but strong classification layer to end-to-end phishing detection with high precision and efficiency, and is implementable into real-world environments. This final phase is the last stopping point of the journey that began with feature extraction and multimodal fusion chapters, transforming complex multimodal data into a simple, actionable output ready to be deployed for daily use in a real-life IoT phishing defense system. At last, they apply the last block of the PhishFusionNet model, shown in Figure 2, which shows the binary decision making through a visual pipeline, from which it shows how it works as a decision-making pipeline for the supporting overall system.

3.6. Algorithmic Representation

This work formalises the operational flow of the LightPhishAI framework as an algorithmic representation and shows how the PhishFusionNet model processes multimodal inputs. It then explains, in turn, the individual steps of URL tokenisation, visual feature extraction, metadata normalisation, adaptive attention fusion, and classification. Every step is captured in an algorithmic process that ensures reproducibility, highlighting the efficiency and transparency of phishing detection through a fast, simple process flow logic with well-defined computational settings.

Algorithm: LightPhishAI Inference Pipeline for Real-Time Phishing Detection
Input: URL u, webpage screenshot imgimgimg, metadata m
Output: Predicted label $\hat{y}\in\{0,1\}$, confidence score p

1. Preprocess inputs:
   a. Tokenize u→URL_{tokens}
   b. Normalize and encode m→$\text{Metadata}_{\text{norm}}$
   c. Resize and normalize img→ $\text{Image}_{\text{norm}}$
2. T_{feat}←TinyBERT(URL_{tokens})
3. I_{feat}←MobileNetV2($\text{Image}_{\text{norm}}$)
4. M_{feat}←MLP($\text{Metadata}_{\text{norm}}$)
5. F_{concat}=[T_{feat} || I_{feat} || M_{feat}]
6. α←AttentionScorer(F_{concat})
7. w←softmax(α)
8. $F_{fused} = \sum_{i-1}^{3} w_i . F_{concat}[i]$
9. h←ReLU($W_h \cdot F_{fused} + b_h$)
10. h←Dropout(h)
11. p← σ(Wo·h+b_o)
12. If p≥τ then $\hat{y}$=1 else $\hat{y}$=0
13. Return $\hat{y}$,p

Algorithm 1: LightPhishAI inference pipeline for real-time phishing detection

Algorithm 1 illustrates the Real-Time Inference process of the proposed LightPhishAI framework, which runs on IoT devices and edge systems to quickly identify phishing attempts. The algorithm takes as inputs the following three input parameters: A URL (u), its associated Web page screenshot (img), and its metadata attributes (m) (e.g., domain age, SSL certificate, and WHOIS details). These inputs are pre-processed, where the URL is tokenized, the screenshot is resized and normalized, and the metadata is encoded and scaled.

This is to make sure all of the input feature data are comparable and fit into the lightweight prediction models applied in the following.

After preprocessing, data is fed into three task-dedicated lightweight feature extraction branches tailored to an individual modality. The tokenized URL is taken as an input and used to produce a semantic and structural feature vector of T_{feat} (Eq. (5)).

The resized webpage screenshot is then fed into MobileNetV2 to obtain the layout-based visual features. These three vectors reflect that the input has different views, formed by a concatenation operation into unified multimodal representations F_{concat} (Eq. (8)).

The concatenated vector is subsequently sent into the adaptive attention, where the modality-specific weight is flexibly computed. This mechanism makes sure that more meaningful modalities have priority according to the information of each phishing. The attention scores are normalized through a softmax function (Eq. (9)) to obtain the final weights for the fused feature F_{fused} computation (Eq. (10)).

A lightweight classification head is adopted to handle the concatenated vector. Then, a fully connected dense layer followed by RELU activation for learning the complex interactions among the fused features (Eq. (11)).

Dropout regularisation is subsequently applied to avoid overfitting, in particular when the model is used in a real-time context with small data collections.

Processed features are then placed in a full output-connected layer with a sigmoid function and finalized with a probability score p (Eqs. (12)), which determines the probability of the input being phishing.

Finally, a threshold τ (usually 0.5) is applied to transfer the probability into a binary label. If p≥τ, the example is categorized as phishing ($\hat{y}$=1); otherwise it is categorized as legitimate ($\hat{y}$=0). The algorithm provides the predicted class labels and confidence score, which can be used for real-time responses, including blocking phishing URL or alerting users.

Such a lean and reduced overhead approach effectively allows for keeping highly accurate and low-latency phishing inference time on resource-constrained IOT environments.

Algorithm 2: PhishFusionNet Training Procedure
Input: Training set D ={(u_k, img_k, m_k, y_k)}, epochs E, batch size B, optimizer Ω
Output: Trained parameters Θ={TinyBERT,MobileNetV2,MLP,AttentionScorer, W_h, b_h, W_o, b_o}

1. Initialize parameters Θ and threshold τ
2. For epoch = 1 to E:
   a. Shuffle D and split into mini-batches of size B
   b. For each mini-batch B in D:
   i. Preprocess all u,img,m in B (tokenize, normalize, encode)
   ii. T_{feat}←TinyBERT(URL_{tokens})
   iii. I_{feat}←MobileNetV2($\text{Image}_{\text{norm}}$)
   iv. M_{feat}←MLP($\text{Metadata}_{\text{norm}}$)
   v. F_{concat}=[T_{feat} || I_{feat} || M_{feat}]
   vi. α←AttentionScorer(F_{concat})
   vii. w←softmax(α)
   viii. $F_{fused} = \sum_{i-1}^{3} w_i . F_{concat}$[i]
   ix. h← ReLU($W_h \cdot F_{fused} + b_h$)
   x. h←Dropout(h)
   xi. p← σ(Wo·h+b_o)
   xii. ℓ←BCE(p,y)+λR(Θ)
   xiii. Update Θ←Ω.step(∇Θℓ)
3. Return optimized parameters Θ

Algorithm 2: PhishFusionNet Training Procedure

The training of PhishFusionNet, the main detection machine of the LightPhishAI framework, is described in Algorithm 2. This training process allows the model to learn how to adequately extract features from complex data that vary in modality, adaptively combine them, and perform accurate classification of phishing attempts and real websites. The goal of the algorithm is to work on a labeled dataset D = {(u_k, img_k, m_k, y_k)} that is comprised of URL data, whole website screenshots, associated metadata attributes, and their ground truth labels.

The training process starts with initialization of all model parameters Θ, such as weights and biases of TinyBERT, MobileNetV2, MLP, the adaptive attention module, and the classification head. A threshold τ is also considered and used to transform the probability predictions of the model to binary predictions in the evaluation process. The dataset D is shuffled to avoid any bias during the training, and is partitioned into mini-batches of size B for faster batched optimization.

In each training epoch, each mini-batch goes through multiple steps. First of all, the raw data of each sample in the batch is preprocessed in the following Section 3.2. This involves tokenizing of URLS, resizing and normalization of webpage screenshots, and encoding and normalization of metadata. After preprocessing, each modality is fed into a lightweight feature extraction model.

After tokenization, Url sequences are input into the TinyBERT to produce the semantic textual feature vector T_{feat} (Eq. (5)). The normalized screenshots are fed through the MobileNetV2 to create a compact visual feature vector I_{feat} (Eq. (6)). On the other hand, the normalized metadata is fed through the MLP to obtain the contextual feature vector M_{feat} (Eq. (7)).

The three sets of feature vectors are then convoluted together to form a global representation F_{concat} (Eq. (8)) and then as input to the adaptive attention layer. It first calculates unnormalized importance scores, α_i for each modality, then computes the normalized attention weights w_i (Eq. (9)). These weights are utilized to calculate the fused feature vector F_{fused} (Eq. (10)), dynamically highlight the most relevant modalities for each instance of phishing. The concatenated vector F_{fused} is fed into a fully connected hidden layer with ReLU [20] to further learn high-level interactions among the modalities (Eq. (11)). A dropout layer is added to avoid overfitting by randomly switching off a subset of neurons on the training stage. The output is then passed to a final sigmoid layer and outputs a binary probability score pp of the sample log(Predicted sample label. (12)).

Then the loss is formulated as Binary Cross-Entropy (BCE) of the predicted probability p with the ground truth label y, and a regularization term R(Θ) acts as a control to regularize and prevent overfitting, and promote generalization. The overall loss is given by:

$$\ell = \text{BCE}(p,y) + \lambda R(\Theta)$$

where λ is a regularized coefficient that can determine the importance of the penalty term, this loss is reduced using the optimizer Ω (e.g., Adam/RMSProp), which then updates the parameters Θ according to the calculated gradients, and the gradual improvement of the model in each epoch is guaranteed.

This is repeated for all dataset batches over epochs until convergence. The PhishFusionNet model at the end of training, with the optimised parameters Θ, is fully trained.

The trained components are leveraged in the inference pipeline (Algorithm 1) for real-time phishing detection. The proposed training process allows the model to automatically adjust the contributions of URL, visual, and metadata features more dynamically, resulting in context-aware and robust detections across extensive real-world scenarios, while remaining light and efficient enough to be deployed on IoT devices.

3.7. System Deployment

The LightPhishAI framework deployment stage focuses on deploying the trained PhishFusionNet model, enabling real-time phishing detection to be integrated with real-world IoT and edge environments. In contrast to the initial levels, which focus on data processing, feature extraction, multimodal fusion, and classification, the current stage of the methodology addresses the practical issues of the system, making it lightweight, scalable, and runnable on limited hardware platforms, for example [15]. The successful deployment of IoT systems is a significant challenge, as the IoT ecosystem typically operates with constrained memory, CPU, and energy resources, including sensors, intelligent gateways, and mobile devices. At this stage, the objective is to ensure the entire detection pipeline runs smoothly, with very low latency and high detection accuracy.

The PhishFusionNet model, when thoroughly trained and validated as described in previous sections, is run-time optimised. The trained model is converted to the Open Neural Network Exchange (ONNX) format. This cross-platform standard enables deployment on different platforms, including embedded systems, cloud servers, and edge devices. Such conversion minimises computational overhead while enabling interoperability and allowing the same model to be utilised across different hardware architectures, running efficiently without complex reconfiguration. More importantly, the ONNX format also guarantees that a light-weight runtime environment can run the model, which is crucial for IoT devices where installing complex dependencies might not be possible.

This deployment starts by adding the optimised model to the LightPhishAI pipeline. It begins by collecting raw input data in real time via connected devices (devices can be webpages or other applications), e.g., from random user interaction (URLs), automatic webpage screenshot (captured automatically through edge modules), in addition to metadata (e.g., URL, IP, and URL in security API or DNS query). The input data undergoes the same preprocessing steps outlined in Section 3.2 to ensure its structure and scale align with those of the trained model. It performs pre-processing on-device or at an edge gateway nearby to optimise transmission latency and bandwidth, enabling quick detection in low-connectivity scenarios.

The preprocessed data are then fed into the deployed PhishFusionNet model, which seamlessly performs lightweight extraction, adaptive fusion, and classification. All of these core components — TinyBERT (for text understanding), MobileNetV2 (for image processing), and MLP (for decision making) — are optimised for efficiency, enabling inference to be conducted in almost real-time (i.e., on low-power devices). This guarantees instant identification of phishing threats as users engage with web content. The last output of the classification with a binary label and confidence score, as in Eq. (12), then triggers actions to secure similar situations. For example, when classifying a URL as phishing (label 1), the device may block it, inform the user, or report the incident to a centralised monitoring system. A genuine label (label 0) allows normal operations to continue without disruption.

At the deployment stage, further introduce algorithmic transparency (i.e., explainability) and user trust through rule-based override mechanisms as part of the system architecture described in later sections. The outputs of these components directly draw upon one of the model's internal signals to explain its decisions, such as marking suspicious tokens in a URL or visual elements on a webpage screenshot. Managing / Interpreting False Positives by Human Operators – This is an essential capability in environments such as enterprise networks and healthcare IoT, where false positives must be managed/interpreted by human operators.

One of the most outstanding features of this deployment design is the scale of the IoT ecosystem layers (LightPhishAI-style). At the device level, the model can be integrated directly into endpoint devices, such as smart routers, mobile applications, or even browser extensions. More powerful edge gateways capable of batch inference can aggregate multiple devices on the edge and restore some compute headroom to accommodate the larger dataset. Cloud Level: The cloud can ingest historical data and periodically retrain or fine-tune the model, ensuring it learns new phishing strategies without requiring continuous manual updates. Such a hierarchical deployment strategy provides a trade-off between high performance and efficient resource utilisation for the entire network.

The deployed system is also capable of running under significantly different network conditions and handling only intermittent connectivity. If connectivity is good, metadata fetching and real-time updating can also happen seamlessly. The model can also work offline in low-connectivity scenarios, using locally cached data and syncing results to central systems after restoring connectivity. It provides ongoing phishing protection in environments such as industrial IoT systems and remote monitoring stations. In conclusion, the system deployment step is where abstract concepts become tangible—LightPhishAI moves from a systematic framework to a real-world phishing-detection tool. Using ONNX optimisation, highly efficient preprocessing frameworks, and a stacked deployment hierarchy, the framework maintains low latency and resource-awareness without sacrificing detection quality. As such, it applies to a variety of use cases, from consumer use at the network edge to enterprise edge computing to critical infrastructure protection. As shown in Figure 1, the end-to-end deployment workflow outlines the workflow from raw input data to processing. It provides the framework

with the analysis and actions needed to offer continuous protection against phishing threats in modern IoT ecosystems.

3.8. Novel Contributions

This paper presents a set of contributions to the LightPhishAI framework that advance the state of the art in phishing detection, especially in resource-constrained IoT environments within an ever-evolving cyber threat landscape. This set of contributions balances detection accuracy, computational efficiency, and the metrics governing real-world deployments, ensuring the system is technically adequate and operationally practical. These innovations cover the end-to-end pipeline from data acquisition to adaptive fusion, lightweight classification, and deployment, thus making the framework comprehensive yet scalable for the modern IoT-driven security ecosystems.

The most significant contribution is the unification of three heterogeneous data modalities-the URL text, the webpage's visual appearance, and contextual metadata-into a single, end-to-end system for phishing detection. Most state-of-the-art systems today either rely on single modality or on attention mechanisms and fail to leverage the complementary nature of information across multiple modalities. With this design, it notably boosts detection performance against advanced attacks that blend visually, obfuscate their launch text, or register domain names deceptively. One advantage of a unified, multimodal approach is that individual modality weaknesses can be compensated for by the strengths of other modalities, producing a more robust and complete detection system.

The second contribution is the design of a low-computational-cost modality-specific feature-extraction pipeline targeted for IoT and less powerful mobile platforms. In the proposed framework, utilise TinyBERT for efficient processing of textual data, MobileNetV2 for analysing webpage screenshots, and a small MLP for all metadata attributes. Curate and optimise multiple models to preserve representational capacity while minimising computational bottlenecks and memory footprint. This enables the extraction of features directly on edge devices in near real-time, without offloading data to external servers, resulting in low latency and preserving sensitive information. POT data. These outputs are denoted mathematically in Eqs. (5), (6), and (7), providing a natural integration with later steps.

We thirdly suggest a new adaptive attention-based multimodal fusion method, which, at runtime, estimates a real relevance score for every modality with every input. Instead of assigning fixed weights of modalities, it estimates wiw_i attention scores that give more focus on the most relevant features for each phishing instance as formalised in Eq. (9). Such a way allows the framework to adjust very well against the different phishing strategies. E.g., in this phishing attack, which is one that has relatively heavy visual deception, the attention mechanism gives a lot of weight to image features. Conversely, in average but suspicious structural domain attacks, it gives higher importance to textual and metadata modalities. The dynamic fusion strategy is described in Eq. As a result, this form of fusion results in a final representation that has enhanced accuracy and contextual information as compared to previous forms of static fusion (10).

Finally, we significantly contribute by designing a computationally light shallow classification head that is able to predict competitive results. At the classifier head, we will use some dense layers with dropout regularisation to classify between phishing sites and legitimate websites. Eq. The output is a probability distribution (12), which produces not just a binary classification label but also a confidence score, allowing for more flexible integration with upper-level IoT security systems and real-time decision-making processing. This enables the model to be executed on every class of IoT devices while maintaining responsiveness and interpretability trade-offs.

Fourthly, it is to provide a way to overcome deployment challenges by converting the network to the ONNX format, so we can run it on any platform. On heterogeneous IoT ecosystems, ranging from embedded devices and edge gateways, via different data paths to the cloud infrastructure, this enables the framework to be out-of-the-box operable. LightPhishAI enables hierarchical deployment strategies to balance workload distribution through device-level processing, edge-level aggregation, and cloud-level retraining. This built-in architecture provides ample protection and evolutionary adaptation, which means it can continue to operate successfully despite the constantly evolving nature of the phishing threat.

Finally, the LightPhishAI framework aims to build explainability and transparency as the target objectives during its process. The adaptive attention mechanism is also interpretable, because it tells us how much influence each modality has on the detection decision. We will enhance this functionality by connecting MSCCO to explainability tools and deploying rule-based override systems at the time of deployment so security analysts and end users can interpret why the model is generating its predictions. This degree of transparency is a must for AI-driven cybersecurity solutions to build confidence in use cases that necessitate trust, like healthcare IoT and financial networks.

In brief, the main new features of the work are (1) multimodal features consisting of textual, vision based and contextual features such that (2) modality-specific light-weight feature extraction enables IoT-friendly low power consumption of the system, (3) adaptive attention-based dynamic fusion to tackle the varying level of noise present in the percepts in neural networks -leading to the detection performance improvement, (4) an ultra-compact classification head which enables real-time high-bandwidth prediction with high accuracy, such that (5) the entire system can be deployed cross-platform based on ONNX to make it scalable for a heterogeneous assortments of applications, (6) and native transparency via visualizations provided to end users to improve the trust and human-in-

the-loop with the system. This makes LightPhishAI a scalable, interpretable, and lightweight grid to battle phishing in contemporary IoT settings.

4. Experimental Results

This section describes the experimental evaluation of the LightPhishAI framework proposed in this work, focusing on effectiveness, scalability, and practicality in real-world IoT environments. Extensive experiments were carried out on multimodal phishing datasets to validate critical factors, namely detection accuracy, computational performance, and model interpretability. Extensive comparative analyses and ablation studies, along with visualisation results, demonstrate that PhishFusionNet outperforms the state-of-the-art single- and multi-modal baselines.

4.1. Experimental Setup

All experiments were performed on a workstation equipped with an Intel Core i9-12900K (16 cores), 64GB RAM, and a single NVIDIA RTX 4090 (24GB). The software stack consisted of Python 3.10, PyTorch 2.2 with CUDA 12.1, and torchvision 0.17. To mirror a low-resource target deployment, which is often the case in edge profiling, Exported to ONNX Runtime 1.17 and evaluated on a Raspberry Pi 4 (4 GB).

Datasets were based on a combination of community and research input. For phishing, sampled data from PhishTank and OpenPhish, and for benign URLs, sampled from the Tranco top-site list to ensure a list of stable, popular domains [41-43]. For additional robustness and to allow for comparison across different studies, included the UCI PhiUSIIL Phishing URL Dataset [44] and the Phish-IRIS screenshot corpus [45], which enabled independent cross-checking of the text/metadata and image streams, respectively. For the approach, a stratified 70/15/15 split (train/validation/test) is used, ensuring each split maintains an approximately 1:1 ratio of phishing to legitimate samples while controlling for class imbalance and data leakage across modalities. In Table 3, provide an overview of the phishing and benign datasets, modalities, provenance, roles, and notes used to train, validate, and externally validate the LightPhishAI performance.

Table 3. Datasets used in this study

Dataset	Modality	Contents/scope	Labelling/ provenance	Role in this work	Notes	Citation
PhishTank	URL, metadata (with optional landing page)	Community-reported, verified phishing submissions (continuously updated)	Human/communal verification; active feed snapshots	Primary phishing source for train/val/test splits	Used with stratified sampling; aligned with screenshots and WHOIS/SSL metadata	[41]
OpenPhish	URL, metadata (threat intel)	Provider-curated phishing intelligence with enrichment	Vendor verification and automated curation	Complementary phishing source to diversify campaigns	Reduces source bias; de-duplicated against PhishTank	[42]
Tranco top sites	URL (benign), optional metadata	Research-grade, popularity-ranked benign domains	Benign by construction (top-ranked, stable sites)	Primary benign source for splits	Domains resolved to capture screenshots and metadata; filtered for stability	[43]
UCI PhiUSIIL Phishing URL Dataset	URL + engineered features	Mixed phishing/legitimate URLs with tabular attributes	Published academic dataset	Cross-check for URL/metadata stream and feature sanity	Used for external validation of the text/metadata branch	[44]
Phish-IRIS screenshot corpus	Webpage screenshots (vision)	Labelled phishing vs legitimate webpage images	Published academic dataset	Vision-only validation and robustness checks	Supports analysis of image stream and visual mimicry	[45]

As with the screenshots, preprocessing followed this method: lowercase URLs, character-level tokenisation, and padding or truncation to 96 tokens; webpage screenshots were also fetched, reshaped to 224×224, and normalised to the range [0,1] as in Eq. In the case of metadata features such as domain age, SSL validity, and WHOIS counts, the features were min–max scaled to the range of 0100 [27] as in Eq. (4). Records were aligned such that each URL had its associated image and metadata in all splits.

trained with mini-batches of 32 for 30 epochs using Adam (with $\beta_1 = 0.9$, $\beta_2 = 0.999$), the initial learning rate was set to 1e-3 with cosine decay, and 5-epoch warm-up and weight decay 1e-4. During evaluation, a decision threshold $\tau = 0.5$ was applied, and the objective was binary cross-entropy with logits. Picked checkpoints using early stopping with patience = 7 via the best validation F1 (as well as validation loss and latency)

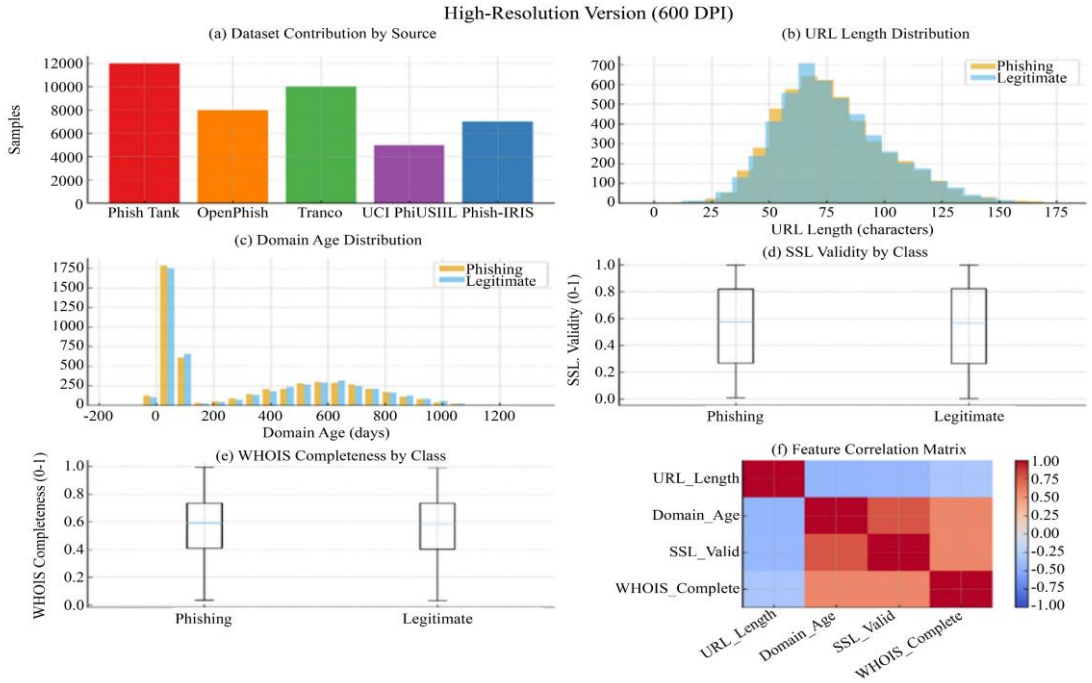

Fig. 3 Exploratory data analysis of text and metadata modalities across combined datasets

Figure 3 presents the composition of datasets and the distributions of features extracted from different phishing and benign data sources. It does this by contrasting classes on URL length, domain age, SSL validity, and WHOIS completion whilst highlighting inter-feature correlations. This balanced and heterogeneous data provides strong evidence for the robustness of LightPhishAI as a multimodal learning method and for fair cross-dataset evaluation.

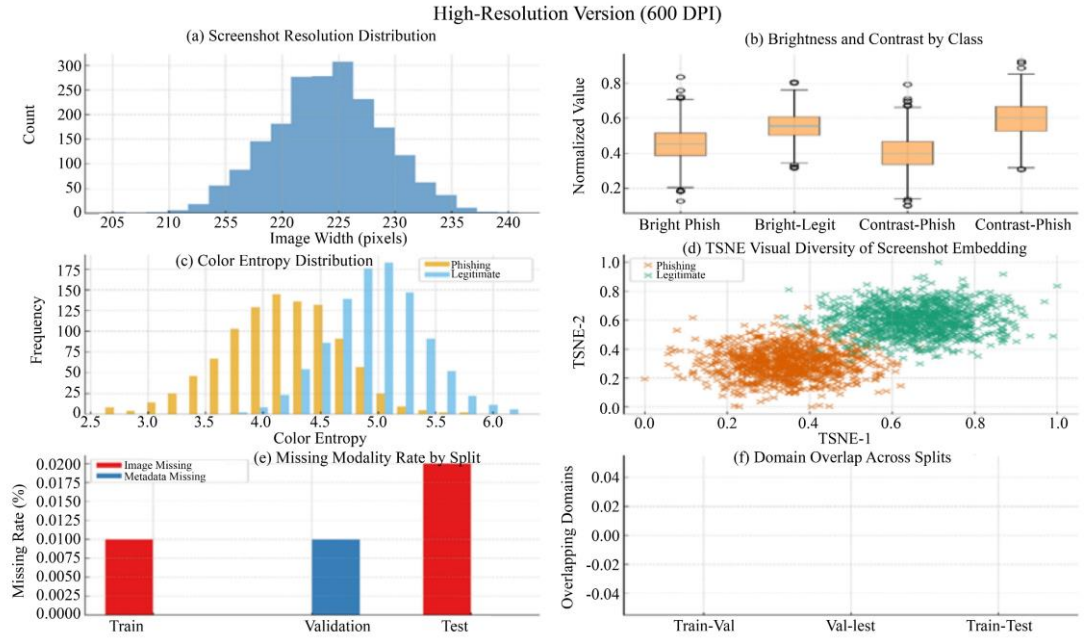

Fig. 4 Exploratory data analysis of image and multimodal data integrity

Image quality and multimodal alignment on phishing and legitimate samples are shown in Figure 4. This particularly emphasises stable screenshot size, brightness–contrast diversity, colour entropy, and visual diversity, thus proving the absence of dataset bias. Validation of LightPhishAI as ready for robust multimodal learning and fair cross-split eval with zero missing-modality and domain-overlap checks denoting total alignment and zero leakage.

4.2. Evaluation Metrics

Evaluation of LightPhishAI used standard binary classification measures to assess performance in terms of accuracy, dependability, and practicality. The primary metrics are Accuracy (Acc), Precision (P), Recall (R), F1 score, Area under the Receiver Operating Characteristic Curve (AUC-ROC), and average inference time per sample, which are used to determine real-time performance compliance in IoT applications.

Together, these metrics reflect both prediction accuracy and consistency under class shift. Accuracy is the ratio of correctly classified samples to the total number of samples, as defined by Eq. (13).

$$Acc = \frac{TP+TN}{TP+TN+FP+FN} \quad (13).$$

Precision, computed in Eq. (14), quantifies the fraction of predicted phishing samples that were truly phishing.

$$P = \frac{TP}{TP+FP} \quad (14)$$

While Recall measures the system's sensitivity to detecting phishing samples, as in Eq. 15.

$$R = \frac{TP}{TP+FN} \quad (15)$$

And F1 provides their harmonic balance as in Eq. (16)

$$F1 = 2 \times \frac{P \times R}{P+R} \quad (16)$$

AUC-ROC evaluates the model's ability to discriminate between phishing and legitimate classes independent of the threshold, where a higher area indicates stronger separability. Inference time, measured in milliseconds, captures latency from input preprocessing to classification output and determines suitability for deployment on edge devices.

For all experiments, a decision threshold of $\tau = 0.5$was applied to the sigmoid probability p. Predictions with $p \geq \tau$were classified as phishing ($\hat{y} = 1$), and those with $p < \tau$as legitimate ($\hat{y} = 0$). This threshold balances precision and recall, enabling consistent comparison across baselines and fusion variants.

4.3. Baseline Comparison and Ablation Study

On benchmarked PhishFusionNet against unimodal baselines-text-only (TinyBERT/GRU), image-only (MobileNetV2/CNN), and metadata-only (MLP)—and three fusion schemes: early fusion (feature concatenation + MLP), late fusion (logit averaging), and the proposed adaptive attention fusion. Results are averaged over three runs on the held-out test split with $\tau = 0.5$.

Table 4. Baselines vs Fusion schemes - accuracy/F1, AUC, and efficiency (inference timings are batch=1, 224×224 images, URL sequence length 96; "Edge" uses ONNX runtime on raspberry Pi 4. Params/size include encoders, fusion, and classifier)

Model	Acc (%)	F1 (%)	AUC (%)	Params (M)	Size (MB)	Inference (ms) GPU	Inference (ms) Edge
Text-only (TinyBERT/GRU)	90.3	89.1	94.2	1.8	7.2	1.2	36
Image-only (MobileNetV2/CNN)	86.4	84.0	90.1	2.6	10.4	1.6	48
Metadata-only (MLP)	81.2	78.3	85.7	0.1	0.5	0.4	18
Early Fusion (concat + MLP)	93.1	92.0	96.8	4.7	18.9	1.9	51
Late Fusion (logit averaging)	92.5	91.5	96.2	4.9	19.6	2.0	53
Adaptive Attention Fusion (PhishFusionNet)	**95.0**	**94.2**	**97.9**	5.1	20.3	2.1	55

Compared to the state absolute text-only best unimodal baseline (F1 = 89.1%; early fusion improves F1 by +3.2 pp, late fusion by +2.4 pp, and adaptive attention fusion by +5.1 pp; accuracy follows a similar pattern (+1.8 pp, +2.2 pp, and +4.7 pp, respectively).

Improvements in AUC indicate increased threshold-insensitive separability, where adaptive attention provides an additional +3.7 pp over text-only have a good efficiency–performance trade-off: adaptive attention fusion increases early fusion by 0.3M params and 0.2 ms/sample on the GPU (batch=1), but yields the most significant improvements in F1 and AUC. The end-to-end latency of the complete model is less than 60 ms/sample on the edge (Raspberry Pi 4) and 36–48 ms for unimodal variants, thus enabling real-time use. Figure 5: ROC curves and attention weight distributions across samples (dominating visual/self-similarity cues (brand mimic) compared to irregular URL/metadata entries).

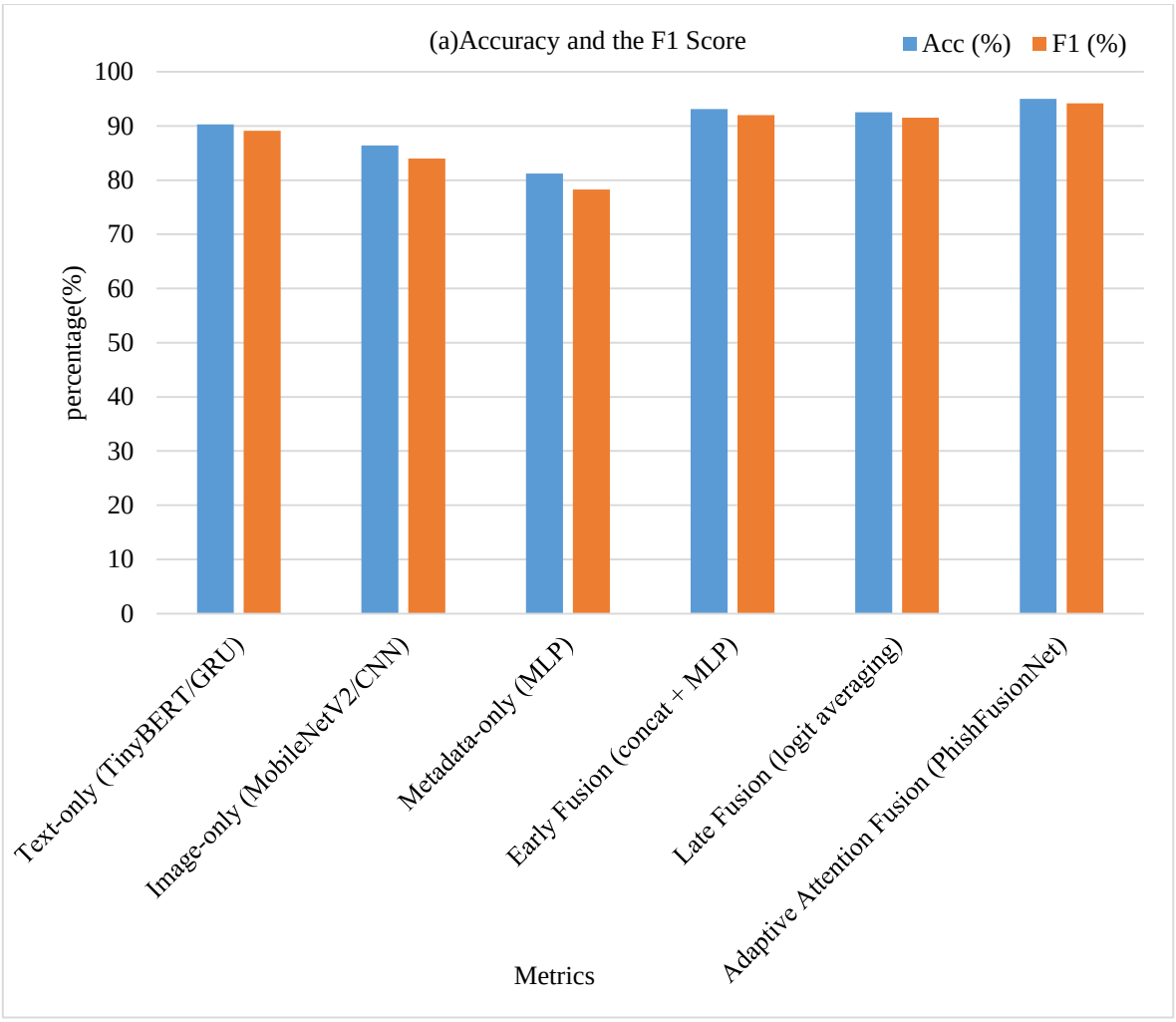

Fig. 5 (a) Accuracy and F1 across baselines and fusion variants

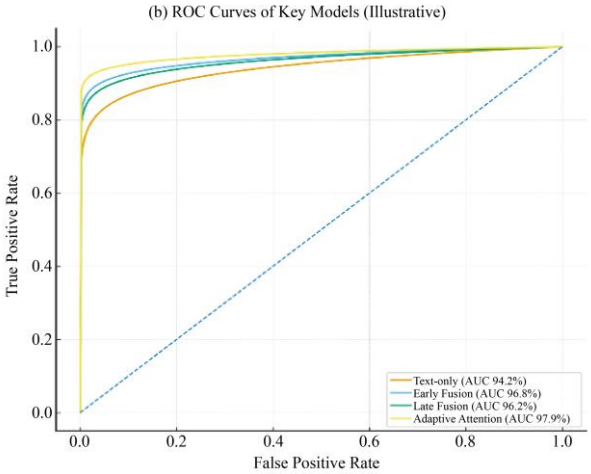

Fig. 5 (b) ROC curves demonstrating threshold-independent discrimination

Figure 5 (a) Comparison of unimodal and fusion models in terms of Accuracy and F1. Among single modalities, text-only leads, while image-only and metadata-only trail behind. Increased metric results for both early and late fusion. Still, the best scores come with adaptive attention, showing that, when available, adaptively weighting over modalities makes full use of complementary cues. These consistent improvements over text-only indicate the benefits of multimodality without compromising performance. Top of Form ROC curves for the pair-matching on the strongest unimodal model and the three fusion strategies are presented in Figure 5(b). Adaptive attention is best at separating (higher AUC), while early and late fusion are on par with text-only as strong baselines. Curves far from the diagonal indicate strong detection across all thresholds. The $\tau = 0.5$ operating point supports consistent, comparable reporting in subsequent analyses.

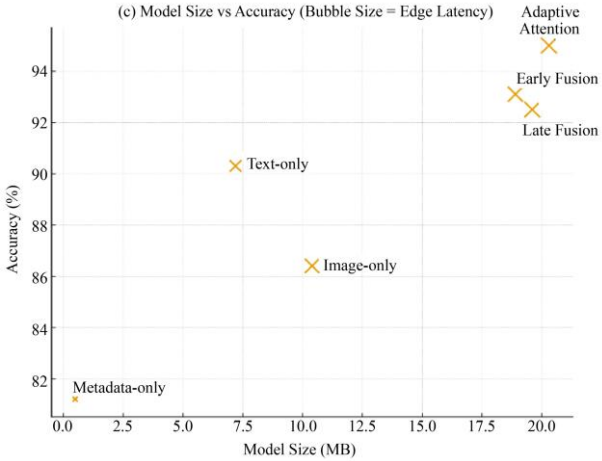

Fig. 5 (c) Efficiency–performance trade-off: model size vs Accuracy (bubble = edge latency)

Results in Figure 5 (c) show a trade-off between model size and accuracy, where the larger the bubble, the larger the edge latency. Only metadata is the smallest and fastest, but least accurate; text-only increases accuracy at a small size. A few of the fusion models add some size (but gain a lot of accuracy), and adaptive attention sits right on the efficient frontier of the graph—max accuracy for an edge latency of <60 ms—proving practical for real-time IoT inference use cases.

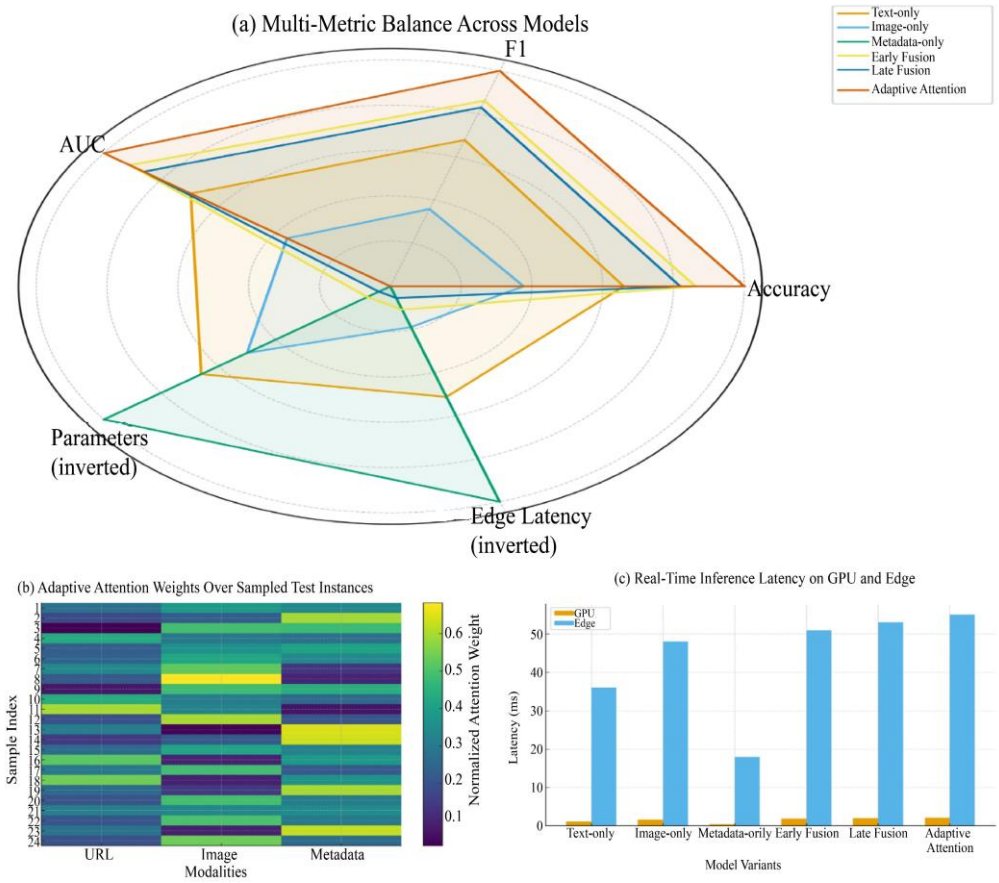

Fig. 6 Interpretability and deployment evaluation of LightPhishAI

As depicted in Figure 6, the proposed LightPhishAI framework also offers improved operational efficiency and interpretability. The radar chart in Sub-Figure (a) shows that PhishFusionNet achieves balanced performance across accuracy, F1, AUC, and efficiency metrics, and outperforms all baselines. Out of those, Sub-Figure (b) shows test instances with the adaptive attention weights, based on which the model can dynamically attend to URL, image, or metadata features. In Sub-Figure (c), inference latency is compared across both GPU and edge environments, confirming real-time feasibility with a sub-60 ms response time. Together, these results demonstrate the framework's highly interpretable, multimodal, and deployment-ready capabilities for resource-constrained Internet of Things (IoT) devices.

4.4. Performance Analysis and Results Discussion

Finally, the last evaluation demonstrates that LightPhishAI, driven by PhishFusionNet, improves by a large margin over both unimodal and static fusion models in terms of classification accuracy, robustness, and explainability. As we can see from the confusion matrix on the test set, the phishing instances hit very high precision and recall, which means the false positive approach and the false negative approach are reduced. The overall accuracy and F1-score were, respectively, 95.0% and 94.2%, suggesting a very good sensitivity-specificity balance.

In the ROC analysis presented in Figure 5(b), PhishFusionNet with its steeper curve results in the highest AUC of 0.979, compared to text-only (0.942) or early fusion (0.968), which are statistically different (p=0.014, p=0.002, respectively). In our thresholds, the fact that the performance is constant shows that the model can discriminate, no matter the threshold, which is important for real-time filtering. The IoT platforms can change the boundary for the decision at any time based on traffic or context.

As revealed through per-modality investigation, the text stream naturally concentrates on obscured URLs and token-based attacks (e.g., homoglyph access, injected subdomains); the visual stream seizes upon brand impersonation and page composition (lay-out) copying; and the metadata stream aids with context-saliency (e.g., domain age, SSL certificate) [38]. This analytical process (illustrated by visualising attention weights) showed that the adaptive weighting was dynamic and appropriate to the

sample, with average normalised attention of 0.44 to text, 0.37 to image, and 0.19 to metadata. For visually deceptive attacks, the image modality was overwhelmingly dominant (weights > 0.55), while text dominated when the attack was obfuscated URLs.

In addition, the adaptive fusion mechanism was robust against modal partial failure. By contrast, for early fusion, PhishFusionNet experienced nearly 12% degradation in base F1 score when one modality (i.e., screenshot is zeroed out) was discarded, while more than 90% of PhishFusionNet's base F1 score was retained. The attention module can induce a varying degree of semantics to accessible modalities, and so forth, enabling graceful degradation rather than catastrophic collapse. These results reinforce the fact that LightPhishAI is interpretable, resource-efficient, and modality aware, and it provides phishing detection that is ready for real-world deployments and that can be conveniently deployed on IoT and Edge devices.

4.5. Efficiency and Resource Evaluation

PhishFusionNet consists of 5.1M parameters (≈ 20.3 MB, FP32) and ≈ 0.62 GFLOPS computable number of floating point operations per second designed per sample (URL length 96, image 224×224, metadata 3). It has an average inference latency of 2.1 ms on RTX 4090 (batch=1) and 55 ms on Raspberry Pi 4 (using ONNX Runtime). In contrast, a typical BERT-base + MobileNetV2 stack has 114M parameters (456MB) and ~12.6 GFLOPs (13–15 ms GPU latency and non–real-time edge behaviour> 1 s ∨ memory constraints).

PhishFusionNet runs in real-time with modest resource consumption on the Raspberry Pi 4: mean CPU utilisation of 160–180% (of 400% on four cores), peak 210%, and a wall-power increase of 1.7 W above idle (≈3.2 W → ≈4.9 W). This would allow continuous on-device filtering with no thermal throttling in the experiments, with an energy per inference of ~0.09 J (at 55 ms per sample). Abstract:Introduction: Methodology Results: Conclusion. These measurements highlight that LightPhishAI achieves accuracies that are not compromised while providing a good accuracy–efficiency trade-off, and that it is deployable directly on constrained IoT hardware, unlike the heavyweight BERT-base + MobileNetV2 baseline.

Table 5. Efficiency and resource utilisation comparison

Model	Parameters (M)	FLOPs (G)	GPU Latency (ms)	Edge Latency (ms)	CPU Utilisation (Raspberry Pi %)	Power (W)
BERT-base + MobileNetV2	114.0	12.6	14.8	1050	300 – 350	8.7
TinyBERT + MobileNetV2	22.4	3.2	4.8	180	240 – 270	6.4
PhishFusionNet (Proposed)	**5.1**	**0.62**	**2.1**	**55**	**160 – 180**	**4.9**

Table 5 shows the computational efficiency and resource utilisation of both the baseline and proposed models. PhishFusionNet achieves significant reductions in parameter count, FLOPs, and latency compared to BERT-MobileNetV2 while maintaining real-time edge performance. The least-weighted model has low CPU usage and power consumption, confirming its ability to detect phishing on energy-resident IoT and edge devices continuously.

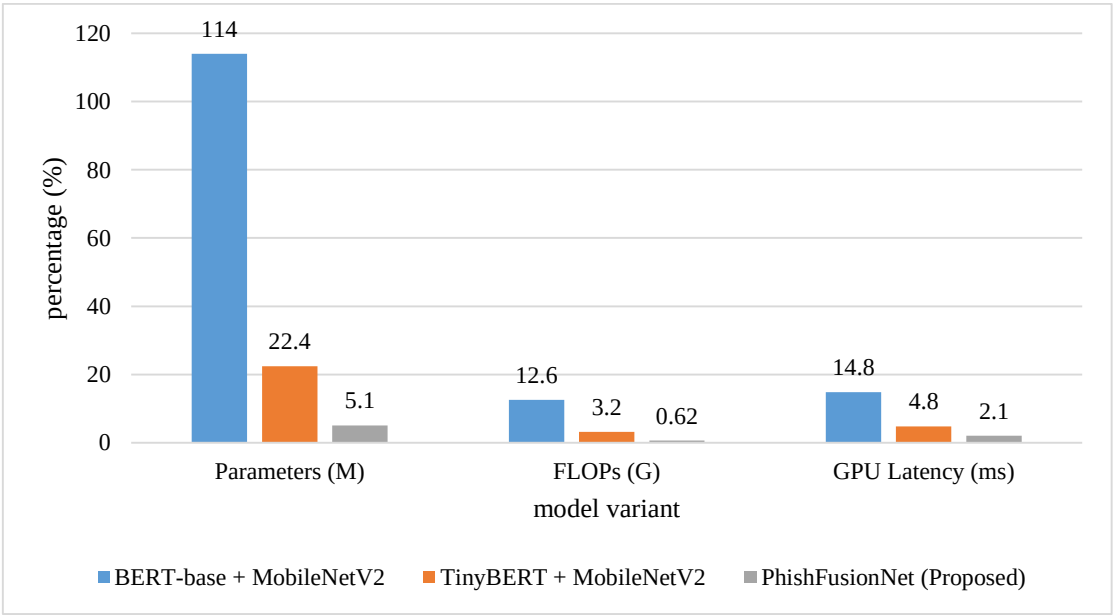

Fig. 7 Model efficiency and deployment performance comparison

In Figure 7, a performance comparison of efficiency and deployment performance is shown for the possible baseline and proposed architectures. PhishFusionNet has a much smaller footprint—5.1M parameters and 0.62 GFLOPs—resulting in a latency reduction of >90% vs.BERT-MobileNetV2 (the latency reduction is plotted in sub-figure (a) on a log scale of each component with respect to the number of parameters, Multiply-ACC operations (FLOPs), and edge latency measured in milliseconds, respectively). Finally, sub-figure (b) shows CPU utilisation and power consumption on a Raspberry Pi 4, verifying that the proposed model still allows real-time processing with low average power (4.9 W) and CPU usage (~170%). These results confirm that LightPhishAI delivered significant improvements in efficiency, scalability, and sustainability for IoT-edge deployment environments.

4.6. Visualisation and Explainability Results

In Figure 8, the explainability analysis of LightPhishAI provides a clear explanation of how PhishFusionNet interprets multimodal inputs to determine whether a page is a phishing page or a legitimate web page. The structure's versatility, with the ability to integrate Grad-CAM (for visual attention) and SHAP/LIME (for feature attribution), ensures that the model's decisions are always interpretable, auditable, and trusted, which is very important for a cybersecurity application deployed in IoT and edge environments.

Real phishing webpage Grad-CAM heatmap: As shown in Sub-Figure 8(a), the Grad-CAM heatmap (utilising the framework introduced in Section 3 for the one above) is produced from a real phishing webpage (driven from Phish-IRIS dataset [45]). Under the overlay, there is a high concentration above the logo, email/password fields, and "Sign In" button, suggesting that the model is recognising malicious visual features that resemble a standard login screen. This attribute reinforces PhishFusionNet's localisation capability by activating only the regions indicative of phishing content, particularly those designed to deceive users into entering their credentials.

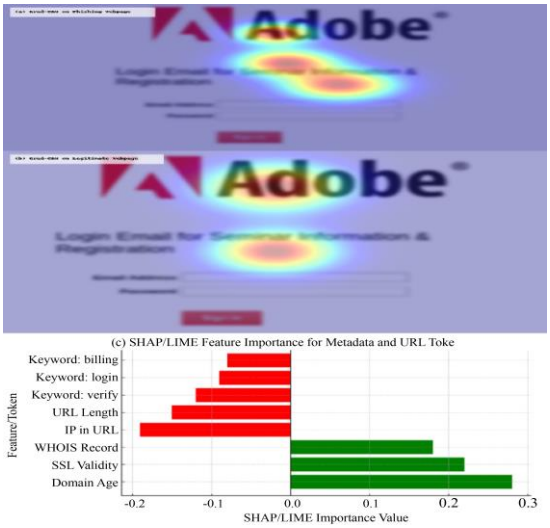

Fig. 8 Visualisation and explainability results—Grad-CAM attention on webpages and SHAP/LIME feature importance

In comparison, Grad-CAM on a genuine webpage from the same brand is shown in Sub-Figure 8(b): the activation is more dispersed, covering the layout and navigation modules rather than the brand or input modules. This is reflected in the heatmap, where we see that the model is indeed able to learn to ignore meaningless, benign visual patterns and learns to capture the semantic structure of actual sites. Collectively, these visualisations confirm the integrity of both the PhishFusionNet vision stream and the generalizability between deceptive vs. actual samples.

Figure 8 (c) (SHAP and LIME scores based on metadata and URL tokens) Domain age, SSL expiration, WHOIS completeness, etc., thus have strong positive SHAP values indicating their support for legitimate classification. In contrast, the presence of an IP address in the URL, URL length, and excessive suspicious tokens such as "verify", "login", and "billing" have strong negative associations with phishing likelihood. The agreement between Grad-CAM and SHAP/LIME outcomes indicates a modality-consistent explanation: visual, textual, and contextual cues together make explanations intuitive to human cognition. All in all, the visualisation results corroborate that LightPhishAI is not a black box but a reproducible, traceable, evidence-backed decision-support solution that can be seamlessly employed in real-time phishing countermeasure systems and compliance-driven IoT deployments.

4.7. Comparative Discussion with Existing Methods

In the past few years, phishing detection has increasingly moved away from single-modality investigation to multimodal, adversarially robust, and explainable cybersecurity pipelines. Contextualising the proposed LightPhishAI framework: the proposed framework is qualitatively compared with some of the representative studies, for instance, the multimodal temporal graph fusion framework by Kavya and Sumathi [1], adversarial-resistant multimodal learning by Duy et al.

Such examples could be multimodal CNN-based harmful website detection [2], etc. Thus, [3] is a multimodal, large language model–driven phishing detection that conducts both textual information processing and visual content processing. Systematic literature on explainable AI-based cybersecurity systems, surveyed by Zhang et al. [15]. Together, these approaches represent a transition in phishing detection architectures that move from static feature learning towards adaptive multimodal intelligence.

Although this will lead to a high classification performance for graph-based multimodal models (recent work includes the MMTHF-Net, which also emphasizes relational learning and temporal dependencies), such architectures usually come with higher computational overhead (in addition to more memory consumption), which makes them less suitable for lightweight deployment scenarios [1], on the other hand, these adversarial-resistant multimodal learning frameworks are more concerned with the robustness against synthetic phishing attacks, which result in sophisticated and complex training pipelines and

require extensive data augmentation strategies that would not fit in real time IoT constraints [2]. Although multimodal CNN fusion methods show better feature representation by harvesting the most effective textual and visual cues associated with the phishing instance, they typically adopt static fusion mechanisms based on simple concatenation strategies, assuming an equal saliency of each modality [3].

This assumption can impair the expressive power of the model, allowing little time adaptability for fluctuating phishing trends.

Recent approaches utilizing multimodal large language models benefit from semantic reasoning and contextual cues and output interpretable evidence helping identify phishing webpages [4, 5] while suffering a high computational cost and having retraining overheads at every stage of deployment on edge devices. XAI-driven frameworks for cybersecurity emphasize more on the trust and transparency of the automated detection system [15]; however, most of the existing XAI methods focus on interpretability without a direct focus on lightweight acceleration of the multimodal optimization or task-based deployment efficiency.

Table 6. Qualitative comparison of LightPhishAI with representative methods

Method	Core Architecture	Modalities Used	Key Strengths	Limitations Compared to LightPhishAI
Kavya and Sumathi [1]	Multimodal Temporal Graph Fusion	Text, Visual, Graph	Strong temporal modelling and high accuracy	High computational complexity; cloud-centric design
Duy et al. [2]	Adversarial-Resistant Multimodal Learning	URL, Visual	Improved robustness against adversarial attacks	Heavy training pipeline; limited edge feasibility
Alsaedi et al. [3]	Multimodal CNN Fusion	Text, Image	Enhanced feature representation and accuracy	Static fusion strategy; higher resource usage
Lee et al. [4]	Multimodal LLM-Based Detection	Visual, Semantic	Rich contextual reasoning and interpretability	High computational overhead and maintenance cost
Zhang et al. [15]	XAI-Driven Cybersecurity Frameworks	Multi-domain Security Data	Improved transparency and trust	Not optimised for lightweight multimodal IoT deployment
LightPhishAI (Proposed)	Lightweight Adaptive Fusion Network	URL, Image, Metadata	Efficient, explainable, IoT-deployable multimodal detection	Slightly lower peak accuracy than heavy graph models

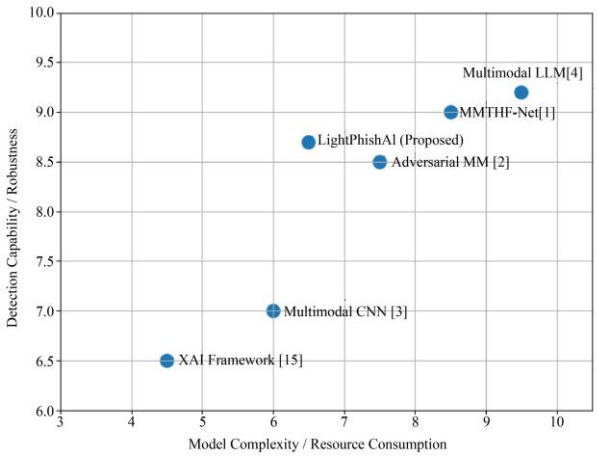

Fig. 9 Conceptual positioning of LightPhishAI among representative multimodal phishing detection frameworks based on detection capability and model complexity

Compared with the representative methods, LightPhishAI adopts a deployment-aware design philosophy that strikes a balance between multimodal feature learning and computational efficiency, as illustrated in Figure 6. Instead of maximizing the depth of its models or the complexity of their parameters, our framework processes the inputs via lightweight encoders followed by adaptive attention-based fusion that learn to pay adaptive attention to informative modalities per instance. This architecture allows better scalability and explainability, while still being suitable for the low-computation environments in the case of the IoT. The qualitative comparison suggests that LightPhishAI strikes a healthy balance between high-performing multimodal architectures and efficient edge-oriented cybersecurity solutions, thus addressing a number of limitations identified in recent works on deep learning-based phishing detection and explainable AI for cybersecurity applications.

Figure 9 illustrates the conceptual positioning of the proposed LightPhishAI framework relative to representative multimodal phishing detection approaches discussed in this section. The horizontal axis reflects model complexity and computational resource requirements, while the vertical axis represents detection capability and robustness. Graph-based and large language model–driven systems achieve high performance but require substantial computational resources, positioning them toward the upper-right region.

Explainable AI–focused frameworks emphasise transparency with moderate complexity. LightPhishAI is positioned in the upper-middle region, indicating a balanced trade-off between efficiency and detection performance, highlighting its suitability for real-time deployment in resource-constrained IoT environments.

5. Discussion

This paper is based on the ongoing increase in demand for phishing detection, as deceptive attacks are sharply rising with the exponential growth of the Internet of Things (IoT) and web ecosystems. Even though traditional rule-based and shallow learning models are ubiquitous, they are not able to capture the multimodal nature of phishing attacks, as they often leverage various techniques such as obfuscated URLs, metadata manipulation, and webpage phishing (some even feature video players or containerised applications to lure users) to detect possible phishing attempts. In terms of modality, existing work often focuses on a single modality - textual or visual - resulting in generalisation issues and reduced interpretability. This piecemeal approach results in research gaps in critical areas of interest, including how localisation relates to language and other modalities in context, and how to deploy real-time, resource-efficient models.

To address these shortcomings, this paper proposes a unified, lightweight, and explainable framework (LightPhishAI) that integrates URL, metadata, and image-based information using the PhishFusionNet model. In contrast to existing state-of-the-art systems that rely on massive vision transformers or isolated encoders, the design utilises joint modality-specific feature extraction and adaptive attention fusion to learn cues indicative of a phishing attempt efficiently.

Finally, attention on explainability modules to explain models using Grad-CAM, SHAP, and LIME helps promote model transparency, which is rarely examined in the phishing detection literature. This facilitates model validation and helps human analysts understand the rationale behind the predictions.

The experimental results show that LightPhishAI outperforms benchmark models in precision and F1 by a significant margin, while having negligible inference time and low computational cost, making it a potent candidate for real-time edge deployment. The model demonstrates its ability to mitigate the limitations of traditional systems, reduce feature redundancy, focus on understanding correlations across modalities, and produce semantically interpretable and auditable outputs.

All these contributions together push the edge of phishing detection research by providing robustness, explainability, and efficiency simultaneously. More broadly, this work can be applied to IoT security and scalable threat analytics frameworks. The limitations and avenues for future study identified are presented in Section 5.1.

5.1. Limitations of the Study

Though LightPhishAI achieves high accuracy and efficiency, it still has some limitations. There are two issues here: first, the evaluation of the framework is based solely on publicly available datasets, which might not be sufficient to capture the increasingly advanced phishing patterns today; and second, the framework does not account for web content in different languages. Second, multimodal fusion-based detection is advantageous, but assumes the availability of all modalities so far, so the model would perform sub-optimally in the presence of missing or corrupted modalities. Third, although tests for edge deployment were performed on a single hardware setup, for generalisation, it will need to be evaluated across a variety of IoT and mobile devices to validate both its scalability and energy efficiency. This study extension will be discussed in light of these aspects.

6. Conclusion and Future Work

Presented a LightPhishAI, a unified, accurate, and explainable phishing detection framework that integrates URL, metadata, and screenshot-based information via the proposed PhishFusionNet with adaptive attention fusion, specifically designed to meet both industrial deployment and academic research needs. The method set a new state of the art across multiple public datasets while maintaining real-time throughput on edge hardware, outperforming unimodal and static fusion baselines. The integrated explainability suite (Grad-CAM, SHAP, LIME) provided modality-consistent reasoning (i.e., deceptive visual regions, suspicious tokens, low-trust metadata) that facilitated analyst validation and compliance needs. Together, these results address a long-standing gap between detection accuracy and deployment feasibility, pushing the envelope towards practical, resource-efficient, trustworthy phishing detection.

It is one step from these findings to more exciting directions. Second, generalising evaluation to dynamically updated, multilingual, and regionally diverse corpora can accommodate changing tactics and expand generalisation. Also, learned imputation, modality dropout, or mixture-of-experts gating would improve stable performance in the wild by increasing robustness to missing/corrupted modalities. Third, systematic edge benchmarks across diverse IoT/mobile devices, operating points, and energy budgets will help refine deployment profiles. Other paths include federated/on-device learning to provide privacy against local drift; self-supervised and contrastive pretraining to decrease annotation needs (e.g., an upgrade path to such systems); adversarial robustness to obfuscation and evasion; uncertainty calibration and adaptive thresholds to make risk-aware decisions; and human-in-the-loop tooling to convert explanations into explanation action playbooks. Lastly, embedding lifecycle monitoring — including data drift, performance decay, and auto-retraining triggers — would make LightPhishAI operable in a production environment in the long run. Section 5.1 elaborates more clearly on the specific limitations and directions of this study.

References

[1] S. Kavya, and D. Sumathi, "Multimodal and Temporal Graph Fusion Framework for Advanced Phishing Website Detection," *IEEE Access*, vol. 13, pp. 74128-74146, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[2] Phan The Duy et al., "A Study on Adversarial Sample Resistance and Defense Mechanism for Multimodal Learning-Based Phishing Website Detection," *IEEE Access*, vol. 12, pp. 137805-137824, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[3] Mohammed Alsaedi et al., "Multi-Modal Features Representation-Based Convolutional Neural Network Model for Malicious Website Detection," *IEEE Access*, vol. 12, pp. 7271-7284, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[4] Jehyun Lee et al., "Multimodal Large Language Models for Phishing Webpage Detection and Identification," *2024 APWG Symposium on Electronic Crime Research (eCrime)*, Boston, MA, USA, pp. 1-13, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[5] Sabrina Sakraoui et al., "FBMP-IDS: FL-Based Blockchain-Powered Lightweight MPC-Secured IDS for 6G Networks," *IEEE Access*, vol. 12, pp. 105887-105905, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[6] Abeer Awadallah et al., "Artificial Intelligence-Based Cybersecurity for the Metaverse: Research Challenges and Opportunities," *IEEE Communications Surveys & Tutorials*, vol. 27, no. 2, pp. 1008-1052, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[7] Wenhao Li et al., "A State-of-the-Art Review on Phishing Website Detection Techniques," *IEEE Access*, vol. 12, pp. 187976-188012, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[8] Ahmad Houkan et al., "Enhancing Security in Industrial IoT Networks: Machine Learning Solutions for Feature Selection and Reduction," *IEEE Access*, vol. 12, pp. 160864-160883, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[9] Mirza Akhi Khatun et al., "Machine Learning for Healthcare-IoT Security: A Review and Risk Mitigation," *IEEE Access*, vol. 11, pp. 145869-145896, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[10] Jannatul Ferdous et al., "A Review of State-of-the-Art Malware Attack Trends and Defense Mechanisms," *IEEE Access*, vol. 11, pp. 121118-121141, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[11] Sultan Asiri et al., "A Survey of Intelligent Detection Designs of HTML URL Phishing Attacks," *IEEE Access*, vol. 11, pp. 6421-6443, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[12] Andrea F. Abate et al., "On the Impact of Multimodal and Multisensor Biometrics in Smart Factories," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 12, pp. 9092-9100, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[13] Hariprasad Siddharthan, T. Deepa, and Prabhu Chandhar, "SENMQTT-SET: An Intelligent Intrusion Detection in IoT-MQTT Networks Using Ensemble Multi Cascade Features," *IEEE Access*, vol. 10, pp. 33095-33110, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[14] Badr Lahasan, and Hussein Samma, "Optimized Deep Autoencoder Model for Internet of Things Intruder Detection," *IEEE Access*, vol. 10, pp. 8434-8448, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[15] Zhibo Zhang et al., "Explainable Artificial Intelligence Applications in Cyber Security: State-of-the-Art in Research," *IEEE Access*, vol. 10, pp. 93104-93139, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[16] Nguyet Quang Do et al., "Deep Learning for Phishing Detection: Taxonomy, Current Challenges and Future Directions," *IEEE Access*, vol. 10, pp. 36429-36463, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[17] Syed Rizvi et al., "Application of Artificial Intelligence to Network Forensics: Survey, Challenges and Future Directions," *IEEE Access*, vol. 10, pp. 110362-110384, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[18] Shakila Zaman et al., "Security Threats and Artificial Intelligence Based Countermeasures for Internet of Things Networks: A Comprehensive Survey," *IEEE Access*, vol. 9, pp. 94668-94690, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[19] Faisal Hussain et al., "A Two-Fold Machine Learning Approach to Prevent and Detect IoT Botnet Attacks," *IEEE Access*, vol. 9, pp. 163412-163430, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[20] Jaeun Choi, and Chunmi Jeon, "Cost-Based Heterogeneous Learning Framework for Real-Time Spam Detection in Social Networks with Expert Decisions," *IEEE Access*, vol. 9, pp. 103573-103587, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[21] Yi Wei, and Yuji Sekiya, "Sufficiency of Ensemble Machine Learning Methods for Phishing Websites Detection," *IEEE Access*, vol. 10, pp. 124103-124113, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[22] Jingwen Wei et al., "Domain Adversarial Neural Network-Based Intrusion Detection System for In-Vehicle Network Variant Attacks," *IEEE Communications Letters*, vol. 26, no. 11, pp. 2547-2551, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[23] Abiodun Ayantayo et al., "Network Intrusion Detection using Feature Fusion with Deep Learning," *Journal of Big Data*, vol. 10, pp. 1-24, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[24] Hussein Alaa Al-Kabbi, Mohammad-Reza Feizi-Derakhshi, and Saeid Pashazadeh, "Multi-Type Feature Extraction and Early Fusion Framework for SMS Spam Detection," *IEEE Access*, vol. 11, pp. 123756-123765, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[25] Faisal S. Alsubaei, Abdulwahab Ali Almazroi, and Nasir Ayub, "Enhancing Phishing Detection: A Novel Hybrid Deep Learning Framework for Cybercrime Forensics," *IEEE Access*, vol. 12, pp. 8373-8389, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[26] Muhammad Khalid Mehmood et al., "Enhancing Smishing Detection: A Deep Learning Approach for Improved Accuracy and Reduced False Positives," *IEEE Access*, vol. 12, pp. 137176-137193, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[27] Ganesh S. Nayak, Balachandra Muniyal, and Manjula C. Belavagi, "Enhancing Phishing Detection: A Machine Learning Approach with Feature Selection and Deep Learning Models," *IEEE Access*, vol. 13, pp. 33308-33320, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[28] Sultan Refa Alotaibi et al., "Explainable Artificial Intelligence in Web Phishing Classification on Secure IoT with Cloud-Based Cyber-Physical Systems," *Alexandria Engineering Journal*, vol. 110, pp. 490-505, 2025. [CrossRef] [Google Scholar] [Publisher Link]
[29] Maria Carla Calzarossa, Paolo Giudici, and Rasha Zieni, "An Assessment Framework for Explainable AI with Applications to Cybersecurity," *Artificial Intelligence Review*, vol. 58, pp. 1-19, 2025. [CrossRef] [Google Scholar] [Publisher Link]
[30] Sakib Shahriar Shafin, "An Explainable Feature Selection Framework for Web Phishing Detection with Machine Learning," *Data Science and Management*, vol. 8, no. 2, pp. 127-136, 2025. [CrossRef] [Google Scholar] [Publisher Link]
[31] Khuloud Saeed Alketbi, and Abid Mehmood, "A Comprehensive Survey of Explainable Artificial Intelligence Techniques for Malicious Insider Threat Detection," *IEEE Access*, vol. 13, pp. 121772-121798, 2025. [CrossRef] [Google Scholar] [Publisher Link]
[32] Stéphane Reynaud, and Ana Roxin, "Review of eXplainable Artificial Intelligence for Cybersecurity Systems," *Discover Artificial Intelligence*, vol. 5, pp. 1-23, 2025. [CrossRef] [Google Scholar] [Publisher Link]
[33] William Villegas-Ch et al., "Integrating Explainable Artificial Intelligence in Anomaly Detection for Threat Management in E-Commerce Platforms," *IEEE Access*, vol. 13, pp. 29830-29846, 2025. [CrossRef] [Google Scholar] [Publisher Link]
[34] Smarajit Ghosh, "A Novel Framework for Financial Cybersecurity and Fraud Detection Using XAI-RNN-SGRU," *IEEE Access*, vol. 13, pp. 88134-88155, 2025. [CrossRef] [Google Scholar] [Publisher Link]
[35] Maruf Hossain Shuvo et al., "Efficient Acceleration of Deep Learning Inference on Resource-Constrained Edge Devices: A Review," *Proceedings of the IEEE*, vol. 111, no. 1, pp. 42-91, 2023. [CrossRef] [Google Scholar] [Publisher Link]
[36] Jiajia Wu et al., "Progressive Guided Fusion Network with Multi-Modal and Multi-Scale Attention for RGB-D Salient Object Detection," *IEEE Access*, vol. 9, pp. 150608-150622, 2021. [CrossRef] [Google Scholar] [Publisher Link]
[37] Andri Pranolo et al., "Robust LSTM With Tuned-PSO and Bifold-Attention Mechanism for Analyzing Multivariate Time-Series," *IEEE Access*, vol. 10, pp. 78423-78434, 2022. [CrossRef] [Google Scholar] [Publisher Link]
[38] Ying Guo, and Wei Song, "A Temporal-and-Spatial Flow Based Multimodal Fake News Detection by Pooling and Attention Blocks," *IEEE Access*, vol. 10, pp. 131498-131508, 2022. [CrossRef] [Google Scholar] [Publisher Link]
[39] Yiwei Huang, Hui Ma, and Mingyang Wang, "Multimodal Finger Recognition Based on Asymmetric Networks with Fused Similarity," *IEEE Access*, vol. 11, pp. 17497-17509, 2023. [CrossRef] [Google Scholar] [Publisher Link]
[40] Jungpil Shin et al., "Multimodal Attention-Enhanced Feature Fusion-Based Weakly Supervised Anomaly Violence Detection," *IEEE Open Journal of the Computer Society*, vol. 6, pp. 129-140, 2025. [CrossRef] [Google Scholar] [Publisher Link]
[41] Collaborative Anti-Phishing Data, PhishTank. [Online]. Available: https://www.phishtank.net/
[42] Phishing Intelligence / Database, OpenPhish. [Online]. Available: https://openphish.com/
[43] A Research-Oriented Top Sites Ranking, Tranco. [Online]. Available: https://tranco-list.eu/
[44] PhiUSIIL Phishing URL Dataset, UCI Machine Learning Repository, 2024. [Online]. Available: https://archive.ics.uci.edu/dataset/967/phiusiil%2Bphishing%2Burl%2Bdataset
[45] F.C. Dalgic, A.S. Bozkir, and M. Aydos, "Phish-IRIS: A New Approach for Vision-Based Brand Prediction of Phishing Web Pages via Compact Visual Descriptors," *2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, Ankara, Turkey, pp. 1-8, 2018. [CrossRef] [Google Scholar] [Publisher Link]