A Hybrid AI Framework for Phishing Detection: Integrating LLM-Based URL Classification, HTML Code Analysis, and Screenshot Evaluation

Polepalle Venakata Sai Harsha
Computer Science and Engineering
Amrita Vishwa Vidyapeetham
Amritapuri, India
amenu4cse21443@am.students.amrita.edu

Rajulapati Harshavardhan
Computer Science and Engineering
Amrita Vishwa Vidyapeetham
Amritapuri, India
amenu4cse21446@am.students.amrita.edu

Lakkireddy Sahana Reddy
Computer Science and Engineering
Amrita Vishwa Vidyapeetham
Amritapuri, India
amenu4cse21233@am.students.amrita.edu

Jishnu A
Centre for Internet Studies and Artificial Intelligence
Amrita Vishwa Vidyapeetham
Amritapuri, India
jishnua@gmail.com

Prabaharan Poornachandran
Centre for Internet Studies and Artificial Intelligence
Amrita Vishwa Vidyapeetham
Amritapuri, India
praba@gmail.com

V.G.Sujadevi
Centre for Internet Studies and Artificial Intelligence
Amrita Vishwa Vidyapeetham
Amritapuri, India
sujap@gmail.com

***Abstract*—Phishing attacks continue to be one of the severe cybersecurity attacks, evolving continuously to evade traditional detection systems. This study suggests an all-encompassing approach to phishing website detection by using sophisticated techniques in URL classification, HTML code-based detection, and screenshot-based analysis. To identify phishing websites, we employ a vast language a model (LLM) that employs Retrieval-Augmented Generation (RAG) URL categorization model, that significantly improves detection accuracy by using both language understanding and retrieval-based methods. We also introduce an HTML code-based detection mechanism which probes webpage structures and embedded components to identify phishing attempts. In addition, we also suggest a screen shot-based detection method in which users may upload or take screen shots of websites to check for phishing behavior. We develop a simple-to-use web interface in which the user may choose among various input types such as URL, HTML code, or screenshot to detect phishing. Experimental results indicate that the combination of LLM-based URL classification, HTML code analysis, and screenshot-based detection enhances detection capability and resilience to attain a scalable, efficient, and universally accessible solution for phishing detection on a wide variety of internet sites.**

***Index Terms*—Phishing Identification, URL Classification, HTML Code Analysis, Large Language Model, Retrieval-Augmented Generation, Screenshot-Based Detection**

I. Introduction

The rapid development of the internet has changed communication, business, and exchanging information as well. The growth, however, has brought with it more cybersecurity threats too, one of the most common and most harmful of which is phishing attacks [25]. These attacks lead to loss of money, identity theft, and users not trusting online systems anymore [26]. Phishing methods evolve to infinity to evade traditional detection mechanisms, and thus, they are hard to the extreme to prevent [18]. Phishing detection systems typically rely significantly on blacklists [5]. They rely on heuristic approaches or rule-based approaches as well [24]. These approaches can detect known phishing sites; however, since known data is needed, they fail with new and unseen threats [22]. Since phishing attacks are very adaptive, much more sophisticated, scalable, and dynamic detection approaches are needed. By being capable of detecting more than rule-based detection patterns, machine learning and deep learning have emerged as effective tools to enhance phishing detection [23].

Machine learning algorithms tend to label phishing attacks with handcrafted URL-based features, whereas deep learning algorithms are able to learn complex patterns from raw data. But these approaches have limitations like class imbalance and novel attack strategy generalization [18]. In order to overcome these shortcomings, we introduce an end-to-end phishing detection system combining three dedicated models for various phishing attack channels. The first model is URL-based detection, which involves a Large Language Model (LLM) combined with Retrieval-Augmented Generation (RAG) for improved phishing classification through the inclusion of external knowledge to make better and context-based predictions [2], [7]. The second model adopts a screenshot-

based detection method via the Bootstrapped Language-Image Pretraining (BLIP) model that retrieves deep visual features from screenshots of websites in order to identify misleading webpage structures [10], [11]. The third model uses RoBERTa to examine the structure and textual content of HTML documents, detecting phishing signals in the source code of the webpage [16].

This multi-modal system enhances phishing detection precision through the utilization of unique feature representations from URLs, visual images, and HTML structure. Furthermore, we introduce an easy-to-use web interface where users are able to provide various input forms URLs, screenshots, or HTML documents for versatile and efficient phishing detection. Through these three detection models combined, our system evolves along with changing phishing methods, creating a strong line of defense against web threats [22].

II. Related Work

A. GPT-2

GPT-2 has shown promise in phishing website detection due to its ability to learn from huge amounts of unannotated information without constant monitoring [1]. With strong efficacy in text generation and classification, it identifies deceptive phishing tactics, such as misleading text and social engineering strategies [3]. One of its main advantages is its ability to understand and generate contextually suitable content without requiring domain-specific retraining, making it adaptable to evolving phishing techniques. Furthermore, GPT-2 can recognize elements of the phishing site, such as false warnings, imitation tech support messages, and misleading instructions [11]. Its capacity to extrapolate between different domains enables it to catch new phishing attacks, even those not explicitly covered during training, making it a dynamic and effective defense against ever-changing phishing threats.

Radford et al. (2019) presented GPT-2, a transformer-based model that supports zero-shot learning, enabling phishing detection with minimal labeled data [1]. GPT-2 identifies lexical, syntactic, and semantic distinctions in URLs, thereby improving detection accuracy compared to traditional rule-based methods. To enhance its effectiveness, GPT-2 is incorporated with Retrieval-Augmented Generation (RAG), which dynamically generates external information for real-time phishing detection. By utilizing the strengths of both prior contextual knowledge and contemporaneous retrieval, this hybrid approach has been found to be a superior strategy for combating evolving cyber threats, surpassing conventional ML models.

B. RAG

Lewis et al. (2020) introduced Retrieval-Augmented Generation (RAG), which enriched NLP models by integrating the retrieval value of information with generative capability for knowledge-driven applications [2]. Unlike traditional transformer models, RAG retrieves relevant information dynamically from outside sources, promoting contextual information and versatility. In our phishing detection method, RAG with GPT-2 is incorporated to augment URL classification through the retrieval of threat intelligence and historical phishing patterns. Real-time adjustment to evolving phishing tactics is permitted, making our model more robust than singular deep learning strategies.

Koide et al. (2023) experimented with ChatGPT usage in phishing site detection, demonstrating capability to analyze URLs and webpage information through contextual understanding [3]. Their study was on LLM-based detection, which maximized accuracy above traditional heuristic and ML methods. Our approach continues from this and integrates GPT-2 with Retrieval RAG (Enhanced Generation), enabling real-time access to phishing signatures and external threat data. The integration allows for better adaptability to evolving attack strategies, enhancing our phishing detection tool's effectiveness and robustness in real-world usage.

C. Screenshot

Visual similarity between phishing and legitimate websites has been widely studied as an effective approach for phishing detection. Traditional methods relied on heuristic-based techniques, such as extracting color histograms, structural layouts, and image matching. However, these approaches struggled with scalability and robustness against website variations. With advancements in deep learning, researchers have explored convolutional neural networks (CNNs) and other vision-based architectures for phishing detection. PhishZoo [12] was an early attempt that used website logos and visual similarity to classify phishing sites. More recent approaches, incorporate object detection techniques to analyze elements of the website such as logos, input forms, and browser toolbars, improving the accuracy of the classification. Lin et al. [10] introduced Phishpedia, a hybrid deep learning-based system that emphasizes logo recognition for phishing detection. By comparing website screenshots with a reference database of brand logos, Phishpedia improves the accuracy of distinguishing phishing sites from legitimate ones. Similarly, He et al. [11] proposed a vision-based model that integrates contrastive learning, enabling phishing detection without requiring extensive labeled data. Despite these advancements, challenges remain, particularly in detecting phishing sites that dynamically alter their visual appearance. Our proposed method builds upon existing research by employing deep learning techniques for screenshot-based classification while integrating textual cues and structural features for improved phishing detection.

D. HTML-Based Phishing Detection

1) ***Feature-Based Detection****:* Researchers have explored phishing detection by analyzing HTML features. AlEroud et al. [6] combined URL and HTML-based indicators to enhance detection accuracy, using feature selection techniques to identify key phishing markers. Similarly, Marchal et al. [15] focused on structured HTML elements, such as form fields, embedded JavaScript, and link analysis, to distinguish phishing websites from legitimate ones.

*2) **Deep Learning Approaches**:* Advancements in deep learning enhanced detection of phishing attacks. Zhang et al. [16] incorporated feature extraction from HTML with deep networks such as CNNs and RNNs and processed the structures of HTML as sequential data in classification. Bahnsen et al. [18] presented a model with the incorporation of HTML DOM graph analysis using a deep neural network, registering great accuracy in identification of phishing schemes.

*3) **Graph-Based Techniques**:* Graph-based techniques provide another method of phishing detection. Liu et al. [19] created a model that forms HTML DOM structures into graphs, with relationships between elements to enhance classification. Zhao et al. [20] introduced an HTML-based Graph Neural Network (GNN), which frames dependencies between HTML elements and improves detection accuracy.

*4) **End-to-End Machine Learning**:* Some work has investigated entirely automated phishing detection. Xie et al. [21] introduced WebPhish, an end-to-end deep learning model that works on raw URLs and HTML content without manual feature engineering, reaching state-of-the-art accuracy. Feroz and Meng [22] studied evasive phishing tactics and suggested an adaptive machine learning framework to defend against them.

III. Methodology

In this study, we present a multi-model phishing detector that examines URLs, web page screenshots, and HTML content with the aim of enhancing the accuracy of detection. Utilizing textual and visual characteristics, our proposed approach improves the capability of the system to successfully detect phishing attacks.

The detection system consists of three components. URL-based analysis, screenshot-based detection, and HTML-based categorization. URL-based analysis employs GPT-2 and RAG to search for anomalies in web URLs through patterns. Screenshot-based detection employs a vision-language model to scan the visual content of a website and extract text information. HTML-based categorization scans a web page's source code for structural and text properties to search for signs of abusive use.

*a) **Novelty of the Approach**:* Our method presents a number of innovations in phishing website detection across various modalities:

- **IP-Level Intelligence in URL Detection**: In contrast to standard models that take the occurrence of an IP address in a URL as a mere binary feature, our method retrieves the IP address and utilizes ISP metadata, geolocation, and IP reputation scores. This yields richer information on the hosting infrastructure, revealing malicious indicators that static lexical features might overlook.
- **Modality-Decoupled Architecture**: All input modalities — URL, web page screenshot, and HTML source code — are handled separately by modality-specific models (e.g., GPT-2 and RAG for URLs, BLIP and OCR for screenshots, and fine-tuned RoBERTa for HTML). Modularity provides robustness, scalability, and optimization to each domain individually.
- **Hybrid Model Integration**: Our model combines GPT-2 and Retrieval-Augmented Generation (RAG) to improve URL-based phishing detection. RAG facilitates the model to draw on external knowledge bases, enhancing its reasoning on rare or changing phishing patterns.
- **Dynamic Threat Adaptation**: RAG enables the model to tap into real-time contextual data, making it robust to new phishing attacks and rapidly changing threat environments.
- **Multi-Modal Detection**: The engine facilitates detection on various data forms — raw URLs, rendered page screenshots, or HTML. This provides high cross-platform and cross-reporting-format applicability with increased phishing coverage detection.

By combining these cutting-edge methods, our method offers an extensible and scalable phishing detection system, assist to reduce cybersecurity risks more efficiently.

*b) **Research Gap**:* Whereas previous studies have examined URL-based, image-based, or HTML-based phishing detection separately, little work has been done on an end-to-end multimodal system. Additionally, existing methods mostly neglect real-time adaptability and IP reputation attributes, particularly ISP and geolocation metadata. Our paper fills this gap by combining state-of-the-art NLP models (GPT-2 with RAG), visual comprehension (through BLIP + OCR), and semantic HTML analysis for comprehensive phishing detection across all modalities.

A. URL

*1) **Dataset preparation**:* The phishing website identification dataset includes URLs collected from public sources, e.g., public repositories like PishTank, Phishing.org, GitHub. So that we have a large and solid dataset, we use URLs of real sites and phishing sites with different types of attack behavior.

*a) **Feature Extraction**:* Each URL in the dataset is processed to extract a set of relevant features for classification. These features can be broadly categorized as:

- **Structural Features**: URL length, presence of special characters, number of subdomains, entropy measures, and counts of specific characters.
- **Domain-Based Features**: IP address, primary domain length, TLD type and length, number of subdomains, WHOIS-related information (including ISP, hostname, country, and region), and the use of HTTPS.
- **Lexical Features**: Tokenization of URL components, presence of suspicious keywords.

*2) **Model Selection**:* Choosing the right model for phishing detection must be a trade-off among accuracy, interpretability, generalization power, and computational efficiency. Our solution unifies GPT-2 in Large Language Model (LLM) with Retrieval-Augmented Generation (RAG) to benefit both from contextual awareness and external knowledge retrieval, greatly enhancing phishing detection [2], [4].

*a) **Generative Pre-trained Transformer**:* GPT-2 Medium is an OpenAI transformer model supporting a number of NLP tasks such as text generation, summarization, and classification [1]. It has 355 million parameters, a compromise between performance and computation. As an autoregressive, decoder-only model, it predicts the next word in a sequence based on multi-head self-attention and feedforward neural networks, hence having good contextual understanding [1]. Having been trained on a vast internet text corpus, it generates coherent and relevant text. However, despite having good language capabilities, it is purely reliant on training data and lacks real-time access to external knowledge.

*b) **Retrieval-Augmented Generation**:* RAG is a neural model that supports language models by introducing a retrieval mechanism to leverage helpful external information when generating text [2]. Unlike regular transformer models, which rely exclusively on pre-trained parameters, RAG combines a retriever and a generator to improve contextual understanding and factual accuracy [2], [7]. The retriever searches a pre-specified knowledge base, e.g., a repository of documents or web pages, to retrieve pertinent information, which is input to a generator like GPT-2 to produce well-informed, contextually aware text. The process is particularly beneficial for real-time knowledge-intensive operations, e.g., question answering, fact checking, and phishing site detection [8]. In dynamically incorporating external information, RAG avoids static language model vulnerabilities, reducing hallucinations and accuracy in high-speed shifting domains [2], [3].

*c) **Model Architecture and Integration**:* The phishing detection model is a hybrid model that consists of a fine-tuned GPT-2-based classifier as a URL classifier and a Retrieval-Augmented Generation (RAG) model as an explanation model for phishing attacks. Besides explaining phishing detection by identifying phishing URLs, the model also provides context-based explanations for why a given URL is being classified as phishing [4], [5].

The architecture includes three primary modules classification, retrieval, and generation, for smooth transition from detection to explicit attack explanation [23].

*d) **Phishing URL Classification using GPT-2**:* The classification module is GPT-2, which is fine-tuned to perform binary sequence classification [4]. The model is fine-tuned with a labeled URLs dataset, classifying between phishing websites and legitimate websites.

The GPT-2 tokenizer is responsible for tokenization, ensuring correct processing, padding, and truncation of URLs to meet the model's input requirements. The fine-tuning process is performed using the cross-entropy loss function, ensuring that the model effectively separates phishing URLs from legitimate ones [23].

*e) **Loss Function**:* For a binary classification task, the cross-entropy loss is defined as:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)]$$

where:

- y_i is the true label (1 for phishing, 0 for legitimate),
- $\hat{y}_i$ is the predicted probability of the URL being phishing,
- N is the total number of training samples.

*f) **Optimization and Learning Rate Scheduling**:* Training is optimized using the AdamW optimizer, which improves generalization by decoupling weight decay. A linear scheduler with warm-up dynamically adjusts the learning rate:

$$\eta_t = \eta_{\text{init}} \times \frac{\max(0, T-t)}{T}$$

where η_t is the learning rate at step t, ensuring smooth convergence.

During inference, a tokenized URL is passed to GPT-2, which classifies it as phishing or legitimate based on predicted logits:

$$\hat{y} = \text{argmax}(\sigma(Wx + b))$$

where W and b are learned parameters. This approach enhances phishing detection by leveraging contextual understanding.

*g) **Retrieval-Augmented Generation (RAG) for Explanation**:* Once a URL has been determined to be phishing, the system proceeds to retrieve relevant documents and generate an explanation of why the URL is a phishing threat. This is done using a Retrieval-Augmented Generation (RAG) model that integrates document retrieval with text generation to enhance transparency in phishing detection [2], [7], [8].

*h) **Document Retrieval using LlamaIndex**:* LlamaIndex is used as the retrieval module to efficiently index and store security-related datasets and research papers on phishing techniques and attack patterns [24], [28]. It processes and organizes a diverse set of authoritative sources, including:

- Alexa Top Sites Dataset (benign website dataset for comparison)
- Blacklist dataset from PhishTank (phishing URL repository)
- Phishing Attack Detecting System Using DNS and IP Filtering [24]
- Phishing and Countermeasures: Understanding the Increasing Problem of Electronic Identity Theft [26]
- PunyVis: A Visual Analytics Approach for Identifying Homograph Phishing Attacks [27]
- Technical Guide to Information Security Testing and Assessment [28]

To enhance retrieval accuracy, the system leverages the all-MiniLM-L6-v2 sentence-transformer model, which converts textual data into vector embeddings. When a phishing query is made, the model calculates cosine similarity between the query vector and indexed document vectors, ensuring precise and relevant knowledge retrieval for phishing detection and explanation generation.

The top-K most relevant documents are selected based on the highest cosine similarity scores, ensuring that the retrieved knowledge is closely related to the input phishing URL.

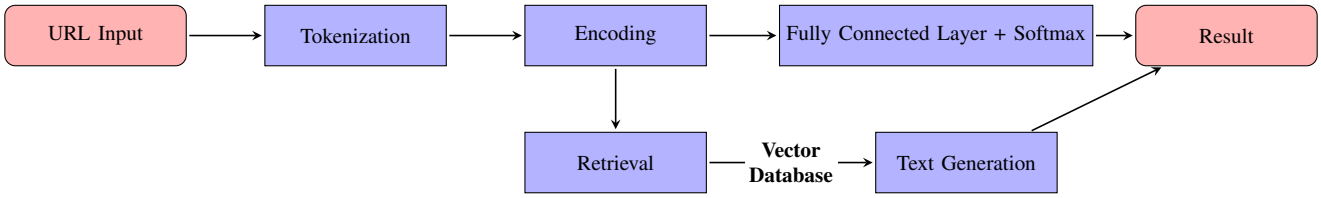

Fig. 1: Flowchart of the URL-based phishing detection system.

*i) **Phishing Explanation Generation using RAG**:* To enhance transparency, the system explains why a URL is flagged as phishing. This is achieved using facebook/rag-sequence-base, a RAG-based language model that generates detailed responses based on retrieved knowledge.

After retrieving the top-K relevant documents, they are provided as context to the RAG model, which generates a descriptive explanation of the phishing attack. The probability of generating a token w_t is computed as:

$$P(w_t|w_{1:t-1}) = \sum_{d \in D} P(w_t|w_{1:t-1}, d)P(d|w_{1:t-1})$$

where D is the set of retrieved documents, ensuring that the generated explanation is fact-based and contextually relevant. This approach helps users understand phishing risks more effectively.

B. Screenshot

*1) **Dataset**:* We employ the PhishIRIS [13] dataset, which is a highly utilized and established benchmark dataset for phishing website detection. The dataset is pre-defined and is specifically developed for phishing website detection based on visual features. It consists of the phishing website screen shots from true sources like PhishTank and OpenPhish. It further includes screenshots of real websites from highly reputed domains to equalize the dataset. All the sample websites are tagged with pre-defined classes that classify the website as phishing or genuine, which is suitable for supervised learning and model evaluation.

TABLE I: Dataset Sizes

Dataset	Legitimate Images	Phishing Images	Total Images
Training Set	400	913	1313
Validation Set	1000	539	1539

*2) **Data Preprocessing**:* To enhance the model's performance and improve generalization, we apply the following preprocessing techniques:

a) Image Normalization: All screenshots are resized to a fixed dimension (H, W, C) to ensure consistency across the dataset. Each image is normalized using:

$$X' = \frac{X - \mu}{\sigma} \tag{1}$$

where:

- X is the original pixel value,
- μ is the mean pixel value of the dataset,
- σ is the standard deviation.

b) Data Augmentation: To prevent overfitting and improve the robustness of the model, we apply augmentation techniques such as:

- **Random rotations**: Rotating images within $\pm 15°$.
- **Flipping**: Horizontal and vertical flips.
- **Brightness adjustments**: Random brightness shifts.

*3) **OCR-Based Text Extraction**:* Since phishing websites often attempt to deceive users by mimicking legitimate brands, extracting textual content from website screenshots is essential. Optical Character Recognition (OCR) is performed using Tesseract-OCR to extract embedded textual information from images. This extracted text is then transformed into numerical features for further processing [14].

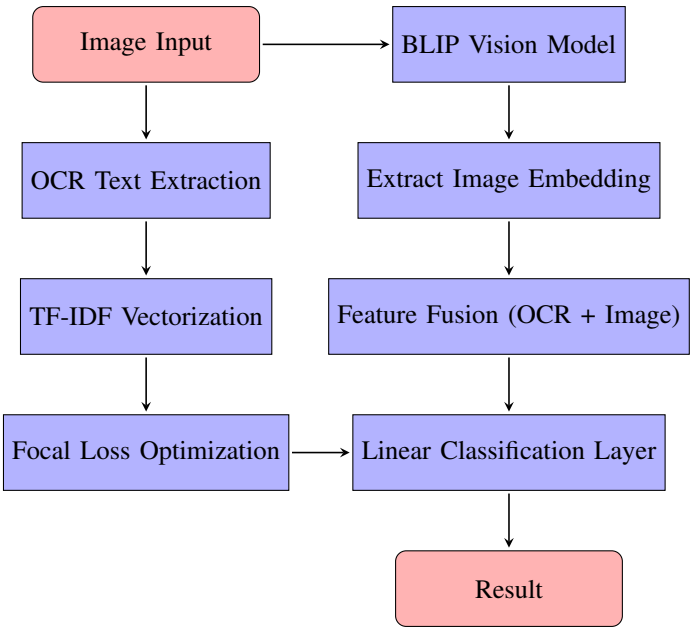

Fig. 2: Multimodal Phishing Detection using OCR and Image Embeddings

*4) **Text Vectorization Using TF-IDF**:* The extracted text is converted into a vectorized representation using Term Frequency-Inverse Document Frequency (TF-IDF). The TF-IDF score of a word w in document d is computed as [17]:

$$\text{TF-IDF}(w, d) = \text{TF}(w, d) \times \text{IDF}(w) \tag{2}$$

where:

- $\text{TF}(w, d) = \frac{\text{Number of times } w \text{ appears in } d}{\text{Total words in } d}$
- $\text{IDF}(w) = \log\left(\frac{N}{1+\text{DF}(w)}\right)$

where N is the total number of documents, and DF(w) is the number of documents containing the word w.

*5) **Multi-Modal Feature Fusion**:* Phishing detection is enhanced by combining visual and textual features using a multi-modal fusion mechanism. The BLIP (Bootstrapped Language-Image Pretraining) [9] model is used to extract 1024-dimensional deep visual features from each image. The fusion process is defined as follows:

$$V = f_{BLIP}(I) \tag{3}$$

where I is the input image, and f_{BLIP} represents the BLIP vision encoder. The extracted textual features T are mapped into the same feature space using a learnable weight matrix W_f:

$$T' = TW_f \tag{4}$$

where $W_f \in \mathbb{R}^{500\times1024}$ ensures compatibility between the textual and visual modalities. The final fused representation F is computed as:

$$F = V + \sigma(T') \tag{5}$$

where σ is the sigmoid activation function.

*6) **Model Architecture and Training**:* The fused representation F is passed through a fully connected classification layer:

$$y = W_c F + b \tag{6}$$

where y is the predicted phishing probability, W_c is a weight vector, and b is the bias term. The model is trained using the Focal Loss function to handle class imbalance:

$$FL(p_t) = -\alpha(1-p_t)^\gamma \log(p_t) \tag{7}$$

where $\alpha = 0.25$ and $\gamma = 2$. The optimizer used for training is Adam with a learning rate of 5×10^{-4}.

During inference, the model takes a website screenshot as input and generates a textual explanation, classifying it as phishing or legitimate based on learned patterns.

C. HTML Code Based

*1) **Dataset Preparation**:* For HTML-based phishing detection, we have created a phishing and normal webpage dataset. The dataset has been separated into a training set and a validation set and kept in various directories. Phishing samples have been collected from PhishTank and OpenPhish [15] and normal pages have been collected from reputable domains.

To improve data quality, HTML content was cleaned using `BeautifulSoup`. Unnecessary tags such as `<script>`, `<style>`, `<meta>`, and others were removed to eliminate non-essential content [16], [21]. URLs embedded within elements like `<a>`, `<img>`, and `<script>` were extracted and included in the cleaned text to capture critical phishing indicators. Additionally, URLs linked to advertisements (e.g., containing "ads") were filtered out to reduce noise. The final cleaned text and extracted URLs were concatenated and trimmed to a 2000-character limit for efficiency [18], [19]. If a page lacked content or contained errors, placeholders such as `"[EMPTY]"` or `"[ERROR]"` were used as fallback values.

For supervised learning, pages were labeled as 0 (legitimate) and 1 (phishing), ensuring clear classification. This enhanced preprocessing allows the RoBERTa model to capture meaningful patterns, improving phishing detection accuracy [25].

TABLE II: Dataset Sizes

Dataset	Legitimate Images	Phishing Images	Total Images
Training Set	6139	4126	10355
Validation Set	1535	1055	2590

*2) **Tokenization and Encoding**:* Since RoBERTa is a transformer model designed to handle textual data, the raw HTML files need to be tokenized before being passed into the model. The RoBERTa tokenizer is then used to transform HTML code into numerical forms, which can be easily processed [29]. The tokenizer pads and truncates so that all the input is of equal length 512 tokens. Tokenized inputs include input IDs as well as attention masks, which tell the model to pay attention to significant tokens but not to the padding tokens.

The tokenization process can be expressed as:

$$T = \text{Tokenizer}(X), \quad T = \{\text{input\_ids}, \text{attention\_mask}\} \tag{8}$$

Once tokenized, the dataset is formatted into a PyTorch-compatible structure by defining a custom dataset class. This class encapsulates tokenized inputs and labels, enabling seamless integration with the model during training.

*3) **Model Selection and Initialization**:* For phishing classification, the general architecture is a pre-trained model of RoBERTa [29]. RoBERTa is a suitable candidate for sequence classification tasks as it has been pre-trained on large databases with a masked language modeling task [1]. To make the model a binary classifier, a fully connected classification head is added to the model so that it can classify if a given HTML page is phishing or legitimate (Marchal et al., 2021) [15].

The model is trained on two output labels, which are the two classes of the dataset.Training on a pre-trained RoBERTa model significantly improves performance by leveraging its rich contextual knowledge of text forms, which is useful in detecting phishing patterns in HTML code [16]. The classification output is calculated as:

$$\hat{y} = \sigma(W_h h + b_h) \tag{9}$$

where h represents the final hidden state, W_h and b_h are learnable parameters, and σ is the sigmoid activation function for binary classification.

*4) **Training Configuration and Optimization**:* To fine-tune RoBERTa for phishing detection, we particularly developed a training setting to balance between performance and avoiding overfitting We train for five epochs with a checkpoint for every epoch We use the Adam optimizer with weight decay of 0.01 to regulate updates and prevent too much parameter tuning [18]. The optimization follows:

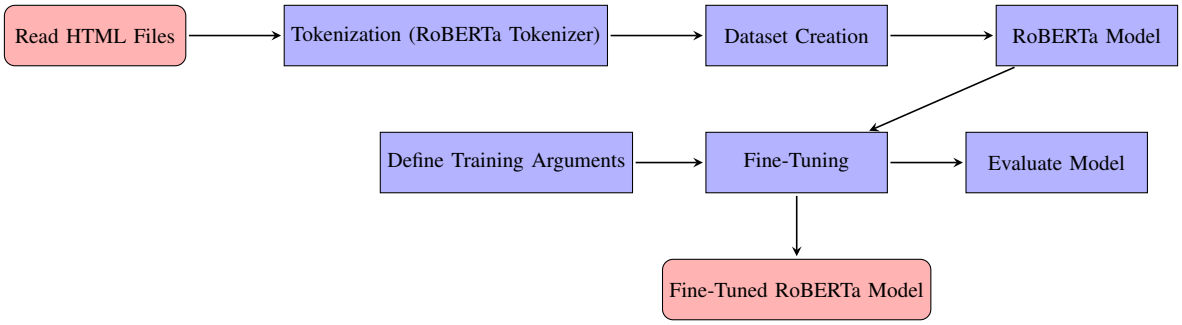

Fig. 3: Fine-Tuning RoBERTa for Phishing Detection

$$\theta_{t+1} = \theta_t - \alpha \frac{m_t}{\sqrt{v_t} + \epsilon} \tag{10}$$

where α is the learning rate and m_t and v_t represent moment estimates, ensuring stable updates.

The batch size is set to 32 balancing computational efficiency with gradient stability. Mixed precision training (fp16) speeds up the computation with no accuracy loss, and gradient accumulation (four steps) gives smooth updates [19].

For model checkpointing we save only the top-performing model based on validation loss, keeping only two key checkpoints for retraining. Training logs are recorded every five steps, providing detailed insights into the learning process [20].

*5) **Model Training and Evaluation**:* The Trainer class from Hugging Face's Transformers library is used to take care of the training. Trainer takes care of data loading, gradient updates, and evaluation on its own. Fine-tuning task involves passing the training data through the RoBERTa model, calculation of loss, and updating the model parameters accordingly. Binary cross-entropy loss function is employed, which is defined as:

$$L = -\frac{1}{N} \sum_{i=1}^{N} [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \tag{11}$$

where y_i is the true label, $\hat{y}_i$ is the predicted probability, and N is the number of samples.

The validation set is utilized to estimate the model's generalization capability, i.e., the ability to well differentiate between phishing and normal pages. The top-performing model is initialized upon the completion of training based on validation loss and is now ready to perform inference on unseen HTML files. This completely fine-tuned RoBERTa model is now ready to label new HTML webpages as phishing or normal based on acquired patterns, which can be utilized for an efficient phishing detection system.

This inference step allows for real-time phishing detection through effective HTML content analysis, which makes it a good candidate for integration into security software and browser plugins.

IV. Result

A. *URL based detection*

we use a combination of GPT-2 and RAG to classify URLs more accurately. Using contextual information and retrieving relevant external data, the model improves phishing detection. Evaluating its performance through accuracy, the ROC curve, and the confusion matrix shows its strong ability to distinguish between phishing and legitimate websites.

1) Accuracy: We got 98.3% accuracy by the combination of GPT-2 and RAG model. The model's accuracy is more helpful for finding whether website is phished or not. The connection between LLM and RAG results in a significant performance boost that improves URL classification by obtaining access to contextually relevant external information for more accurate predictions.

2) Receiver Operating Characteristic (ROC) Curve: The ROC curve shows that the model is strongly capable of discrimination separate phishing from legitimate sites, with low False High Positive Rate (FPR) and high True Positive Rate (TPR). Its a large Area Under the Curve (AUC) value of 0.9973 in addition confirms its robustness.

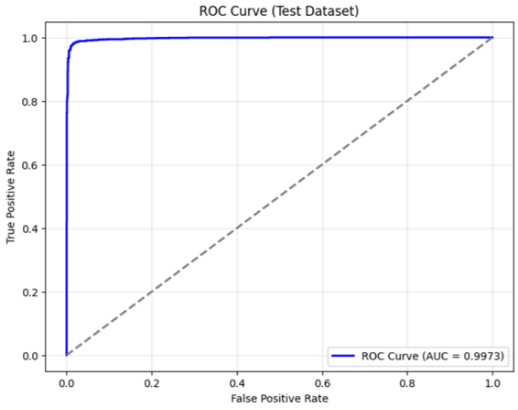

Fig. 4: **ROC Curve for URL detection**

3) Confusion Matrix: The confusion matrix below shows how the model is classifying phishing and normal URLs, indicating correct and incorrect predictions. With fewer in-

correct positives and negatives, the model is highly reliable in distinguishing between genuine and phishing websites.

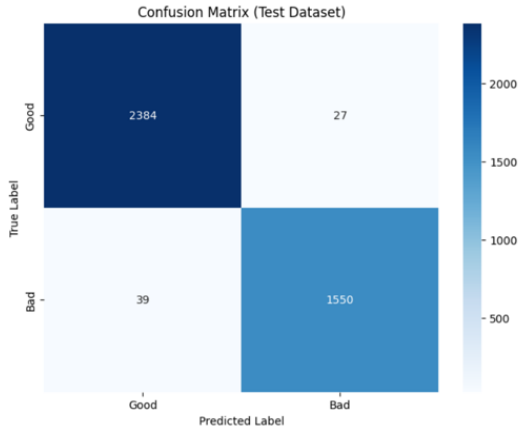

Fig. 5: **Confusion Matrix for URL detection**

4) Discussion: The RAG and GPT-2 model is efficient for phishing detection with a 98.3% accuracy. With Incorporating knowledge retrieval mechanisms into big language models. the accuracy of URL categorization and alleviates false positives. Its high performance is also witnessed by the confusion matrix and ROC curve. These findings suggest that the model is a scalable and dependable solution that can be used in real-world phishing detection tasks.

B. Screenshot based detection

Screenshot-based phishing detection was conducted using the BLIP (Bootstrapped Language-Image Pretraining) model. The fusion of vision and language models allowed for more accurate detection by taking advantage of both visual features and contextual cues from website screenshots.The performance of the model was gauged by several measures, such as accuracy, ROC curve, and confusion matrix.

1) Accuracy: We achieved an accuracy level of 86.4% with The BLIP model is the combination of a vision-language approach to feature extraction and world knowledge were found to be good at identifying phishing sites based on screenshots.

2) Receiver Operating Characteristic (ROC) Curve: To assess the model's capability to differentiate phishing attacks and reliable sources, we bring forward the Receiver Operating Characteristic (ROC) curve. The model is well performance in terms of an Area Under the Curve (AUC) measure of 0.98, which is a high discrimination capability between phishing and legitimate websites.

3) Confusion Matrix: The confusion matrix gives an summary of how accurately the model classifies. It shows HTML code patterns contribute immensely to phishing detection ration. The number of true positives, true negatives, false positives, and false negatives, which show the model's capacity to accurately categorize valid and phishing sites from screenshots.

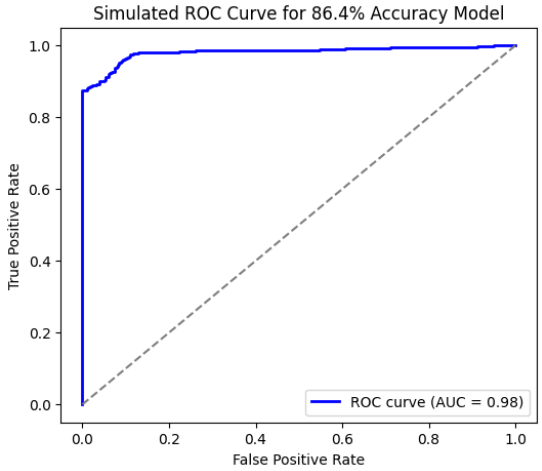

Fig. 6: **ROC Curve for screenshot detection**

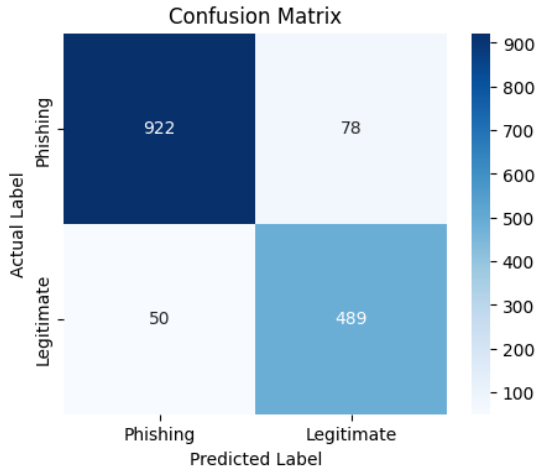

Fig. 7: **Confusion Matrix for screenshot detection**

4) Discussion: The BLIP model turns out to be a power effective tool to identify phishing websites based on screenshots, to achieve 86.4% accuracy with an AUC of 0.98. Its capability Its potential to detect visual phishing signs makes it both reliable and scalable solution to be used practically, with both precision and security.

C. HTML based detection

HTML-based phishing detection was conducted using the RoBERTa model. This model effectively analyzes HTML structures and textual content to identify phishing patterns. Its performance was measured through a number of metrics, such as accuracy, ROC curve, and confusion matrix.

1) Accuracy: We maintained an accuracy of 97.4% with The RoBERTa model shows an ability to understand contextual HTML code patterns significantly improve phishing detection adoption.

2) Receiver Operating Characteristic (ROC) Curve: For purposes evaluate the model's ability to distinguish between phishing attempts Besides real HTML pages, we find the

Receiver Operator. Receiver Operating Characteristic (ROC) curve. It illustrates robust performance with an Area Under the Curve (AUC) a value of 0.99, indicating a high skill in discriminating phishing on legitimate websites.

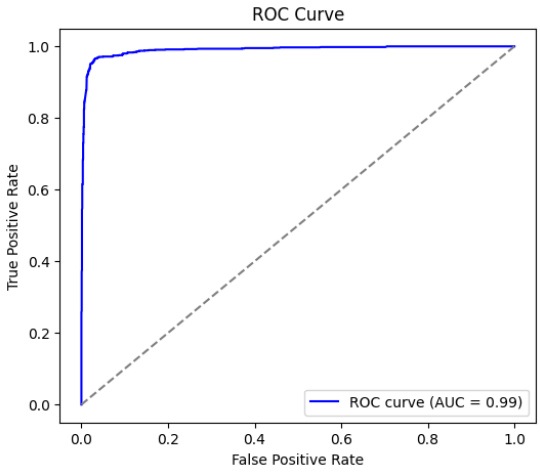

Fig. 8: **ROC Curve for HTML detection**

3) Confusion Matrix: The confusion matrix provides us summary of the model's performance in classification. It shows the number of true positives, true negatives, false positives, and false negatives, indicating that the model can correctly recognize phishing and authentic HTML pages.

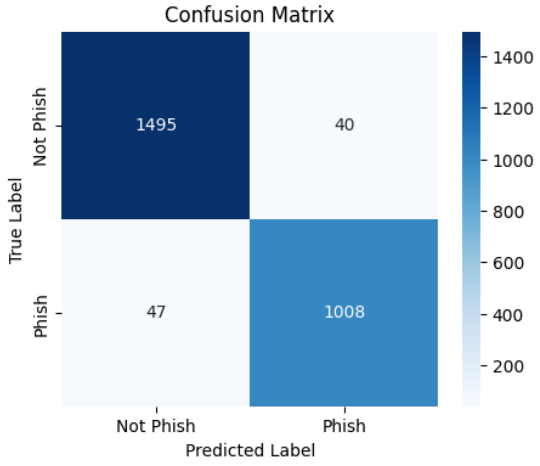

Fig. 9: **Confusion Matrix for HTML detection**

4) Discussion: RoBERTa is proficient at phishing detection via HTML analysis with 97.4% accuracy. and a high AUC of 0.99. With the understanding of HTML structures and text patterns, it effectively identifies phishing attempts. Its Strong performance makes it a reliable and scalable solution. for real-world cybersecurity.

D. Comparision with existing models

V. Conclusion

Phishing is a serious cyber attack that requires advanced methods of threat identification to facilitate efficient remediation. This book presents a proposal for a multimodal language. model, computer vision, and HTML structure understanding phishing detection system based to enhance effectiveness of detection and accuracy.

TABLE III: Comparison of URL-Based Models

Model	Accuracy	Precision	Recall	F1 Score
GPT-2 + RAG (Ours)	**98.3**	**98.2**	**97.5**	**97.9**
Bi-LSTM	97.0	95.1	94.8	95.0
Transformer	97.3	96.2	95.4	95.8
CNN (UCI)	98.2	97.0	97.2	97.6
Random Forest	97.0	96.8	97.8	97.3
Hybrid Ensemble	85.4	87.0	84.0	85.5

TABLE IV: Comparison of Image-Based Models

Model	Accuracy	Precision	Recall	F1 Score
BLIP + OCR (Ours)	**86.4**	**91.0**	**90.9**	**90.7**
CNN-LSTM	92.0	90.1	89.5	89.8
ResNet50 + SVM	89.5	87.4	85.9	86.6
InceptionV3 + CNN	88.7	85.5	84.2	84.8
CNN (Pretrained)	84.6	82.2	80.5	81.3
Hybrid (CNN + ResNet)	83.2	80.0	78.9	79.4

a) URL-based detection: RAG and GPT-2 combination results to obtained our model 98.3% accuracy, and it worked effectively of the retrieval-augmented language models in the detection of phishing URL.

b) Screenshot-based detection: With BLIP, this procedure had a success rate of 86.4%, which is the ability of vision-language models to traverse webpage images.

c) HTML-based detection: Implemented with RoBERTa, our method reached an accuracy of 97.4%, proving its utility in the study of web architectures and the content of text elements.

The precision reported through the ROC curve and confusion matrix assures that the system possesses high detectability precision with hardly any false positives and false negatives. With the application of multiple detection techniques, this solution has enhanced flexibility to new phishing techniques, offering A secure framework for cybersecurity guidelines. This method can be utilized in web-based phishing detection security software, browser add-ons, and security solutions to actively detect and prevent phishing attacks. By providing URL, screenshot, and HTML-based security, it enhances user secures and threats. Enhancement to cyber threats on the Internet the security of the Internet for individuals and organizations.

TABLE V: Comparison of HTML-Based Models

Model	Accuracy	Precision	Recall	F1 Score
ROBERTa (Ours)	**97.4**	**96.8**	**96.0**	**96.4**
XGBoost	96.5	94.9	93.8	94.3
Random Forest	95.0	94.2	92.5	93.3
BERT	97.2	96.1	95.8	95.8
GNN	94.5	93.0	91.7	92.3
GAT	91.5	88.9	87.6	88.2

REFERENCES

[1] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). "Language Models are Unsupervised Multitask Learners."

[2] Lewis, P., Perez, E., Piktus, A., et al. (2020). "Retrieval-augmented generation for knowledge-intensive NLP tasks." In *Proceedings of the 34th International Conference on Neural Information Processing Systems (NeurIPS '20)*.

[3] Koide, T., Fukushi, N., Nakano, H., & Chiba, D. (2023). "Detecting Phishing Sites Using ChatGPT." arXiv preprint arXiv:2306.05816.

[4] Wang, H., & Hooi, B. (2024). "Automated Phishing Detection Using URLs and Webpages." arXiv preprint arXiv:2408.01667.

[5] Rao, R.S., & Pais, A.R. (2017). "An Enhanced Blacklist Method to Detect Phishing Websites." In *Information Systems Security. ICISS 2017. Lecture Notes in Computer Science*, vol. 10717, Springer.

[6] AlEroud, A., & Karabatis, G. (2022). "An Effective Detection Approach for Phishing Websites Using URL and HTML Features." *Scientific Reports*. Available at: https://www.nature.com/articles/s41598-022-10841-5.

[7] Omrani, P., Hosseini, A., Hooshanfar, K., Ebrahimian, Z., Toosi, R., and Akhaee, M. A. (2024). "Hybrid Retrieval-Augmented Generation Approach for LLMs Query Response Enhancement." *2024 10th International Conference on Web Research (ICWR)*, Tehran, Iran, 22-26. Available at: https://doi.org/10.1109/ICWR61162.2024.10533345.

[8] Yu, Jeffy. (2024). "Retrieval Augmented Generation Integrated Large Language Models in Smart Contract Vulnerability Detection." Available at: https://arxiv.org/abs/2407.14838.

[9] Li, J., Li, D., Xiong, C., Hoi, S. (2022). "BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation." *arXiv preprint arXiv:2201.12086*. Available at: https://arxiv.org/abs/2201.12086.

[10] Lin, Y., Wang, S., Du, X., et al. (2021). "Phishpedia: A hybrid deep learning-based approach to visually identify phishing webpages." *Proceedings of the ACM Internet Measurement Conference*, 511–525.

[11] He, Y., Zhang, H., Liu, S., et al. (2022). "Contrastive learning for phishing website detection based on visual similarity." *Expert Systems with Applications, 202*, 117140.

[12] Afroz, S., & Greenstadt, R. (2011). "PhishZoo: Detecting phishing websites by looking at them." *IEEE International Conference on Semantic Computing (ICSC)*, 368–375.

[13] Dalgic, F. C., Bozkir, A. S., and Aydos, M. (2018). "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, 1-8.

[14] Wang, Y. and Duncan, I. (2019). "A Novel Method to Prevent Phishing by using OCR Technology." *2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*, Oxford, UK, 1-5. Available at: https://doi.org/10.1109/CyberSecPODS.2019.8885101.

[15] Marchal, S., Saidi, H., & Francillon, A. (2021). "Phishing Website Detection through Multi-Model Analysis of HTML Content." *arXiv*. Available at: https://arxiv.org/html/2401.04820v1.

[16] Zhang, X., et al. (2023). "Phishing Webpage Detection via Multi-Modal Integration of HTML DOM Graph and URL Features." *MDPI Electronics*. Available at: https://www.mdpi.com/2079-9292/13/16/3344.

[17] A. Vazhayil, R. Vinayakumar and K. P. Soman, "Comparative Study of the Detection of Malicious URLs Using Shallow and Deep Networks," *2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Bengaluru, India, 2018, pp. 1-6.

[18] Bahnsen, A.C., Torroledo, D., Camacho, B., & Villegas, S. (2021). "A Survey of Intelligent Detection Designs of HTML and URL Phishing Attacks." *IEEE Xplore*. Available at: https://ieeexplore.ieee.org/document/10019269.

[19] Liu, X., et al. (2021). "Phishing Web Page Detection with HTML-Level Graph Neural Network." *IEEE Xplore*. Available at: https://ieeexplore.ieee.org/document/9724318.

[20] Zhao, Y., & Liu, W. (2023). "Look Before You Leap: Detecting Phishing Web Pages by Exploiting Raw URL and HTML Characteristics." *arXiv*. Available at: https://arxiv.org/abs/2011.04412.

[21] Xie, T., et al. (2022). "HTMLPhish: Enabling Phishing Web Page Detection by Applying Deep Learning Techniques on HTML Analysis." *ResearchGate*. Available at: https://www.researchgate.net/publication/347020476.

[22] Feroz, N., & Meng, W. (2022). "Towards Web Phishing Detection: Limitations and Mitigation." *arXiv*. Available at: https://arxiv.org/abs/2204.00985.

[23] J. V. Jawade, & S. N. Ghosh. (2021). "Phishing Website Detection Using Fast.ai Library." *2021 International Conference on Communication, Information, and Computing Technology (ICCICT)*, IEEE, pp. 1-5.

[24] Islam, M., Sajjad, M., Hasan, M. M., & Mazumder, M. (2023). "Phishing Attack Detecting System Using DNS and IP Filtering." *Asian Journal of Computer Science and Technology*, 12, 16-20. Available at: https://doi.org/10.51983/ajcst-2023.12.1.3552.

[25] A. A. A. and P. K., "Towards the Detection of Phishing Attacks," *2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)*, Tirunelveli, India, 2020, pp. 337-343. Available at: https://doi.org/10.1109/ICOEI48184.2020.9142967.

[26] Jakobsson, M., & Myers, S. (2006). "Phishing and Counter-Measures: Understanding the Increasing Problem of Electronic Identity Theft." Available at: https://doi.org/10.1002/9780470086100.

[27] Fouss, B., Ross, D. M., Wollaber, A. B., & Gomez, S. R. (2019). "PunyVis: A Visual Analytics Approach for Identifying Homograph Phishing Attacks." *2019 IEEE Symposium on Visualization for Cyber Security (VizSec)*, Vancouver, BC, Canada, pp. 1-10.

[28] National Institute of Standards and Technology (NIST), "Technical Guide to Information Security Testing and Assessment," 2018.

[29] T. Bikku, J. Jarugula, L. Kongala, N. D. Tummala, and N. V. Donthiboina, "Exploring the Effectiveness of BERT for Sentiment Analysis on Large-Scale Social Media Data," *2023 3rd International Conference on Intelligent Technologies (CONIT)*, Hubli, India, 2023, pp. 1-4.