PhishGuard: Multi-Faceted Phishing Detection: Leveraging URLs, HTML Features, and Visual Cues

Parul Sindhwad
(CoE CNDS Lab)
Veermata Jijabai Technologcial Institute)
Mumbai, India
pvsindhwad_p21@el.vjti.ac.in

Kashish Gandhi
D. J. Sanghvi College of Engineering
Mumbai, India

Vaishnavi Padiya
D. J. Sanghvi College of Engineering
Mumbai, India

Siddhi Muni
D. J. Sanghvi College of Engineering
Mumbai, India

D. Shah
D. J. Sanghvi College of Engineering
Mumbai, India

Prateek Ranka
D. J. Sanghvi College of Engineering
Mumbai, India

Faruk kazi
(CoE CNDS Lab)
Veermata Jijabai Technologcial Institute)
Mumbai, India

***Abstract*—Phishing websites remain a critical cybersecurity threat, leveraging both structural and visual deceptions to mislead users. This paper introduces a comprehensive approach to phishing detection by integrating URL analysis, HTML content inspection, and advanced visual feature extraction. For URL analysis, character-level embeddings and sequence-based models are employed to detect phishing patterns. HTML content is parsed and analyzed using tree-based models to identify structural anomalies, such as malicious tags and scripts. Visual feature extraction is conducted using a combination of color histograms, wavelet hash features, Scale-Invariant Feature Transform (SIFT), Local Binary Patterns (LBP), and Oriented FAST and Rotated BRIEF (ORB). These features capture both global and local visual characteristics that distinguish phishing sites from legitimate ones. The extracted features are integrated into a deep learning framework that utilizes Mask R-CNN for pixelwise segmentation and classification, enhancing visual analysis, Support Vector Machine (SVM) for URL and XGBoost with hyper-parameter optimization using Optuna for HTML-based classification. The system is implemented as a Google Chrome extension, providing real-time detection of phishing websites during browsing with an overall accuracy of 97.38%.**

***Index Terms*—component, formatting, style, styling, insert.**

I. Introduction

The digital era has completely altered everyday life, it has redefined the way people socialize, work, and carry out transactions, which is more suitable for the users but it also leads to the great threat posed by digital crimes. The massive use of the internet comprised of digital devices, banks, e-commerce platforms, and social media is responsible for the creation of many potential vulnerabilities. This increased dependence on digital infrastructure heightens the threat and exposure to cyber-attacks and thus the need for robust cybersecurity measures to protect the user's personal and sensitive information is required. Phishing is a cyber attack technique, that involves attackers copying legitimate websites in order to trick people into disclosing out sensitive information like passwords, credit card details, or even personal IDs. Emails are one of the most popular mediums through which phishing attacks are executed, followed by social media messages or fake websites. Phishing attacks are usually disguised as urgent or alluring messages to deceive victims into clicking on malicious links or downloading harmful attachments. This allows cyber criminals to gain illegal access to valuable data. [1] Md Abu Imran Mallick and Rishab Nath provided an in-depth analysis of the cyber environment that is evolving, highlighting that illicit cyber activities have become more complex with the emergence of new technologies such as cloud computing, IoT, cryptocurrencies and elaborated various cyber attacks including DDoS, phishing, and malware etc. Goenka, Chawla, and Tiwari [2] enumerated some such potential attacks, one being website-based attacks like Cross-Site Scripting (XSS), where hackers introduce fake scripts to legitimate websites, thus enabling them to gain direct access to the targeted system through the internet automatically, Cross-Site Request Forgery (CSRF), where the user is tricked into running unauthorized actions on authenticated sites, Domain Squatting, Clickjacking, and SQL Injection are all methods that attackers use to exploit web restrictions and manipulate data. Efficient protection is paramount for these potential vulnerabilities which includes not only the thorough URL checks and robust authentication mechanisms but also the reliable incorporation of top-notch security applications. Massive data is available on the internet which imposes a huge risk of phishing for various websites and therefore detecting phished websites becomes crucial. The most basic parts that a website possesses—the URL, HTML content, and the visual aspect—serve as the primary indicators to determine whether a website is phished or legitimate. This paper identified an approach to exploit the three components of the URL, HTML, and visual content for phishing detection.

The paper consists of two main approaches - The first approach is the development of three different classifiers on three different datasets and leveraging advanced feature extraction techniques and Machine Learning Algorithms. The second approach comprised of developing a Combined Classifier using a unified dataset which consisted of scraped HTML, URL Link, and Screenshots of various phished and legitimate websites. Three individual classifiers were built on the unified dataset. Then a combined model was developed which was more robust and could combine the results of all three classifiers and provide a final solution. Additionally, a comparison was done between the two approaches to highlight the best one. A web Browser Extension was also built to test the model, it automatically scrapped the HTML data, URL, and screenshot of the website and accurately detected if the website was phishing or legitimate website in real-time.

The paper makes the following contributions :

1) Development of URL, HTML, and Screenshot-based classifiers on different datasets for detecting phished websites.
2) Development of a combined classifier which for phished website detection and comparing the capabilities of the combined and the individual classifiers.
3) Development of an extension that detected phished websites in real-time.
4) Proposed integrated approach, implemented through a Google Chrome extension achieved an overall accuracy of 97.38%.

The rest of the paper is organized as follows:
Section 2 presents a concise summary of the studied literature. Section 3 briefly describes the dataset used for the study. Section 4 describes the proposed methodology. In Section 5 the results of the proposed methods are discussed, followed by a conclusion in Section 6 and future work in Section 7.

II. Background

In the quest to enhance phishing website detection, recent studies have illuminated various aspects of the challenge and identified significant opportunities for advancement. The study presented in [3] focused on URL-based features, a common approach in phishing detection. While effective, it revealed a notable gap: the need to explore additional features beyond URLs, such as webpage content and JavaScript analysis. This insight highlights a critical opportunity to expand detection methodologies by incorporating alternative algorithms or advanced deep learning techniques. Echoing this need for sophisticated methods, the study in [4] underscored the potential benefits of integrating advanced machine learning techniques and real-time detection capabilities. The findings suggest that adopting more sophisticated methods could significantly enhance the robustness and efficacy of our detection framework.

Building on the importance of utilizing diverse datasets and complex models, the study in [5] achieved a commendable 94.5% accuracy using a small dataset and traditional models. This research emphasizes the necessity of larger datasets and more advanced models, motivating us to adopt extensive datasets and explore more complex machine learning models. Similarly, the study in [6] demonstrated the effectiveness of combining visual and textual features using a CNN for classifying screenshots. This study suggested incorporating additional algorithms to further enhance detection capabilities, reinforcing our approach to explore hybrid feature integration.

Ensemble and hybrid approaches have proven particularly effective, as illustrated by the study in [7]. By employing stacking with Gradient Boosted Decision Trees (GBDT), XGBoost, and LightGBM, this study achieved high accuracy and low false alarm rates. This approach highlights the advantages of combining multiple algorithms to improve detection performance. The study in [8] similarly employed XGBoost with a hybrid feature set, including URL features, HTML code, and DOM tree rules. This demonstrated the benefits of integrating various features and advanced models, inspiring our research to explore similar hybrid methodologies.

The combination of feature extraction with deep learning models presents another promising avenue. The study in [9] utilized tokenization and CNNs for pattern detection, showcasing the effectiveness of merging feature extraction techniques with deep learning for enhanced performance. This approach was further supported by the study in [10], which used the MTLP dataset and integrated URL, HTML content, and WHOIS data with pre-trained NLP models and a Multilayer Perceptron (MLP). The success of this multi-model approach motivates us to explore similar methodologies to improve accuracy in our own system.

Pattern mining techniques, as discussed in [11], offer valuable insights into detecting phishing websites. This study concentrated on URL patterns and related features, suggesting that incorporating real-time capabilities and more contextual and behavioral features could enhance detection. This insight guides us towards integrating pattern mining techniques into our approach to improve phishing detection.

Visual feature extraction is also critical. The study in [12] echen et al. entmployed features such as color histograms, Wavelet Hash, and SIFT, demonstrating their effectiveness in detecting phishing sites based on visual similarity. Although the study acknowledged limitations in handling new or subtle phishing techniques, it emphasized the benefits of incorporating visual features into our detection framework. The study in [13] further explored visual feature extraction by using ORB for logo detection, validated by a Siamese Network, and enhanced by IP Mapping. It highlighted the need for additional feature extraction methods and adaptability to evolving user interfaces. A novel approach was used in [14] which used machine learning models and demonstrated how HTML source code can be transformed into 2D images using BinVis, which are then classified using TensorFlow achieving high accuracy, with XGBoost performing the best at 94.16%. However, the study highlighted several key challenges such as enhancing dataset quality and diversity, and managing feature extraction complexities. Finally, the study in [15] utilized Local Binary Patterns (LBP) for texture description, achieving an accuracy of 83.1%. This study showcased LBP's effectiveness in phishing detection and suggested further exploration of image descriptors and real-time application integration.

Our project integrates URL-based features, HTML content analysis, and visual feature extraction into a real-time Google Chrome extension, addressing several gaps identified in prior research. By combining these three aspects, our system benefits from a comprehensive approach that leverages the strengths of each method. The Chrome extension provides immediate protection by detecting phishing attempts as users browse, utilizing advanced feature extraction techniques for URL patterns, HTML structures, and visual characteristics such as color histograms, Wavelet Hash, and SIFT descriptors. This multimodal framework represents a significant advancement in phishing detection, offering high accuracy and minimal false positives while delivering practical, real-time user protection directly within the browser.

Recent advances in multi-modal phishing detection have demonstrated the effectiveness of integrating multiple data sources. Murhej and Nallasivan [16] proposed a multimodal framework utilizing SMS, E-Mail, and URL datasets with EM-BERT and EAI-SC-LSTM, achieving accuracies exceeding 99% across different datasets. Similarly, Shammi and Shyni [17] developed a stacking ensemble classifier combining SVM, Neural Networks, Random Forest, and improved LSTM, demonstrating superior performance in detecting phishing attacks through efficient feature descriptors. The applicability of hybrid frameworks has been further validated by [18], achieving 97.44% accuracy while emphasizing robustness against bypass attempts and real-time detection capabilities.

Ensemble learning approaches have gained significant traction in recent years. Alsariera [19] conducted a comprehensive investigation of AI-based ensemble methods including AdaBoost, Bagging, and Gradient Boosting, identifying their effectiveness across website, email, and SMS phishing detection. Abu-Zanona et al. [20] compared six machine learning techniques with ensemble approaches, demonstrating that ensemble methods consistently outperform individual classifiers. The PhishGuard system [21] introduced a multi-layered stacked ensemble architecture with optimized feature selection using RFECV and PCA, surpassing existing methods across diverse datasets. These recent works underscore the trend toward hybrid, multi-modal, and ensemble-based approaches that combine the strengths of multiple detection methodologies to improve accuracy and robustness against evolving phishing techniques.

III. Datasets

This research utilized a diverse range of datasets, each contributing to different aspects of phishing website detection. The approach was based on two main strategies:

A. Separate Dataset Utilization

- **URL Dataset: PhishTank Dataset**
  The PhishTank dataset comprises 9,400 URLs, categorized into phishing and legitimate sites. This dataset was used to focus on URL-based features for phishing detection [22].
- **HTML Dataset: Mendeley Dataset**
  The Mendeley dataset contains 80,000 instances of both phishing and legitimate websites, providing the HTML content of each webpage. From this dataset, 3,905 instances were selected to extract 27 features from the HTML pages using BeautifulSoup [23].

- **Screenshot Dataset**
  Screenshots were assembled from three sources:
  - **Mendeley Dataset via Automated Web Scraping:** High-resolution screenshots of websites were captured using automated web scraping tools. This dataset includes the 3,905 instances from the Mendeley HTML dataset [23].
  - **CIRCL Phishing Dataset:** This dataset includes 460 screenshots of phishing websites, enhancing the diversity of phishing content [24].
  - **Phish IRIS Dataset:** Contains 1,313 training and 1,539 testing samples of phishing and legitimate websites, divided into 15 classes, including one class for legitimate samples and 14 classes for different phished brands [25].

B. *Unimass Dataset Utilization*

This dataset integrates all three parameters—URL, HTML, and visual features (screenshots)— into a single resource. Comprising 30,000 samples equally divided between phishing and legitimate websites, the Unimass dataset allows for the development of a combined classifier. For this research, 6,000 samples were utilized to train classifiers based on URLs, HTML content, and visual features, facilitating a comprehensive approach to phishing detection [26].

By employing these two strategies, phishing detection was analyzed from multiple angles, leveraging both specialized and unified datasets to enhance the robustness and accuracy of the models.

C. *Dataset Selection for Final Implementation*

Based on comprehensive performance evaluation across both dataset strategies, the Unimass unified dataset was selected for the final Chrome extension implementation. This decision was driven by three key factors: (1) The unified dataset enables training of all three classifiers (URL, HTML, and visual) on consistent data distributions, improving model generalization; (2) The ensemble model trained on Unimass achieved superior performance with URL accuracy of 98.12%, HTML accuracy of 95.00%, and visual accuracy of 95.40%; and (3) The integrated training approach reduces dataset bias and improves real-world applicability. The final system accuracy of 97.38% reported in this paper corresponds to the weighted ensemble of these three classifiers trained on the Unimass dataset, evaluated on a held-out test set comprising 6,000 samples (3,000 phishing and 3,000 legitimate websites).

IV. Proposed Methodology

The proposed methodology for phishing website detection involves a comparative analysis between specialized datasets and a unified dataset. Initially, three specialized datasets are utilized: the PhishTank dataset for URL-based features, the Mendeley dataset for HTML content, and a collection of screenshots from sources including automated scraping of Mendeley, the CIRCL Phishing Dataset, and the Phish IRIS dataset from Kaggle. Each dataset contributes uniquely to phishing detection by focusing on specific features—URLs, HTML, and visual content, respectively. In parallel, the Unimass dataset, which combines URL, HTML, and screenshot features, is used to develop a unified classifier. This combined approach allows for a comprehensive analysis of phishing detection capabilities. The performance of models trained on specialized datasets is compared against those trained on the Unimass dataset to evaluate which approach yields superior accuracy and robustness in phishing detection. This comparative evaluation aims to determine the most effective method for identifying phishing websites.

Figure 1 provides an overview of the proposed methodology, consisting of feature extraction and chrome extension.

A. *Detection based on URL*

The process begins with loading the PhishTank dataset containing URLs, which are classified as either phishing or legitimate. The initial steps involve cleaning the data by selecting relevant columns and handling missing values. This is followed by feature extraction, where advanced characteristics of each URL are calculated. The features extracted include the length of the URL, the number of dots, hyphens, special characters, and various other URL components that might indicate suspicious activity. Another key aspect of feature extraction is the analysis of the domain name and path structure within the URL. For example, phishing URLs may use subdomains or domain names that closely resemble those of legitimate websites but include slight variations, such as additional hyphens or altered spellings. By quantifying these variations, the model can learn to recognize patterns that are indicative of phishing attempts. Additionally, the frequency and distribution of characters within the URL are incorporated as features, helping to identify URLs that deviate from typical legitimate patterns. This thorough and detailed feature extraction process ensures that the model has access to a rich set of data points that can be used to make accurate predictions. Once the features are extracted, the next step is to train machine learning models that can classify URLs as phishing or legitimate based on these features. Several machine learning algorithms are employed in this phase, including Random Forest, Gradient Boosting, SVM, and Neural Networks. These models are chosen for their ability to handle complex

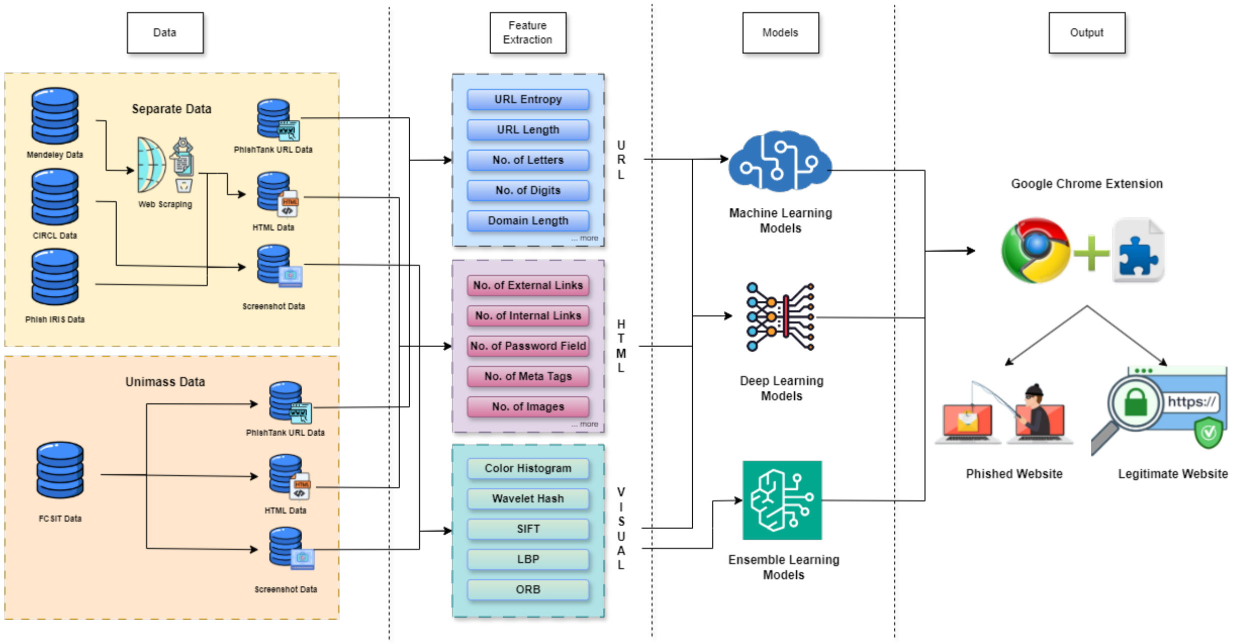


Fig. 1. Overview of Proposed Methodology

datasets and their effectiveness in classification tasks. The dataset is split into training and testing subsets, with the training set used to fit the models and the testing set used to evaluate their performance. After training the models, a thorough evaluation is conducted using the testing set. This evaluation involves calculating key performance metrics such as accuracy, precision, recall, and the F1-score. Confusion matrices are also generated to visualize the model's performance, providing insight into how well the model distinguishes between the two classes. To further enhance the model's performance, hyperparameter optimization techniques such as RandomizedSearchCV are employed.

B. Detection based on HTML Code

The complete HTML of a Web Page scrapped using BeautifulSoup was used to extract 31 features categorized into three groups:

URL-Based Features: These included the total number of characters in the URL (URL Length), the count of periods within the URL (Number of Dots in the URL), and the presence of special characters (Number of Special Characters), the number of subdomains present (Number of Subdomains), and the URLs were checked for patterns commonly associated with phishing (Suspicious URL).

HTML Element-Based Features: This involved counting various elements such as forms (Number of Forms), input fields (Number of Input Fields), password fields (Number of Password Fields), number of links pointing to external (Number of External Links) and internal (Number of Internal Links) domains, as well as hidden input fields (Number of Hidden Fields), 'mailto' links, meta tags, iframes, images, style tags, onload and onerror attributes, and forms submitting data to external domains.

JavaScript-Based Features: These features included the total number of script tags (Number of Scripts), JavaScript functions (Number of JavaScript Functions), presence of specific functions like eval(), escape(), unescape(), setTimeout(), and setInterval(). The analysis also checked whether the URL used HTTPS, a factor often associated with legitimate websites but also sometimes employed by phishing sites to appear secure.

C. Detection based on Visual Inspection

Feature extraction is a pivotal component in the analysis and classification of images, particularly in the context of phishing website detection. The ability to discern and quantify various image attributes significantly enhances the performance of machine learning models. In this research, several advanced feature extraction techniques have been employed to capture different aspects of image data. Each technique provides unique insights into the image content, contributing to a robust and comprehensive analysis framework. The following techniques are utilized:

1) Color Histograms
2) Wavelet Hashing
3) Scale-Invariant Feature Transform (SIFT)
4) Local Binary Patterns (LBP)

5) Oriented FAST and Rotated BRIEF (ORB)

These techniques are integrated into the feature extraction process to ensure a thorough analysis of the visual content, enhancing the overall capability of the phishing detection system.

1) Color Histogram Features: Color histogram features capture the distribution of colors within an image, which can help identify the visual aesthetics of a webpage. Phishing websites often mimic legitimate ones but might have subtle differences in color distributions.
A color histogram represents the frequency of occurrence of different color intensities in an image. For an image with N pixels and M color bins, the histogram H_i for bin i is calculated as:

$$H_i = \frac{1}{N} \sum_{j=1}^{N} \delta(C(j), i) \tag{1}$$

where $C(j)$ denotes the color value of pixel j, and δ is the Kronecker delta function, which is 1 if $C(j) = i$ and 0 otherwise.

The color histogram is a global feature that summarizes the overall color distribution in an image. It is particularly useful for distinguishing images that have different color compositions, such as the differences between a legitimate and a phished website. Phishing sites may attempt to copy the visual appearance of legitimate sites, but slight variations in color distributions can occur. Analyzing these distributions can help detect these discrepancies and identify potential phishing attempts. Color histograms are robust to small changes in image content and can provide a strong indication of the visual authenticity of a website. They can be calculated efficiently and are effective in distinguishing between legitimate and phished websites.

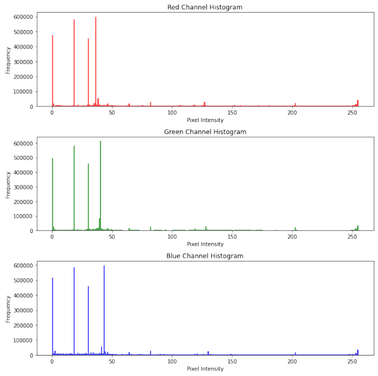


Fig. 2. The histogram represents the color distribution for a legitimate website.

2) Wavelet Hash Features: Wavelet hashing involves transforming an image into the wavelet domain and then generating a hash based on the transformed coefficients.

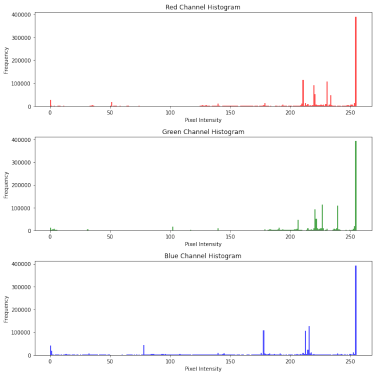


Fig. 3. The histogram on the right (B) represents the color distribution for a phishing website.

This method is particularly effective in capturing both global and local features of an image. The wavelet transform decomposes an image into a set of coefficients that represent different frequency components. Given an image I, the wavelet transform $W(I)$ produces a set of coefficients C_{ij}, where:

$$C_{ij} = W(I)_{ij} \tag{2}$$

A hash H can be computed by quantizing these coefficients using a predefined threshold T. The quantization is performed as follows:

$$H_{ij} = \begin{cases} 1 & \text{if } C_{ij} > T \\ 0 & \text{if } C_{ij} \leq T \end{cases} \tag{3}$$

where T is the predefined threshold.

Wavelet hash features capture both the spatial and frequency information of an image, making them sensitive to subtle changes in texture and structure. This is crucial for detecting alterations that might be indicative of phishing. Phishing websites may look visually similar to legitimate ones at first glance, but the underlying structure, as captured by wavelet transforms, may differ. The wavelet hash can highlight these differences and contribute to the detection process. Wavelet hashing is resilient to small variations in the image and can reveal structural inconsistencies that might be missed by other features. It provides a compact representation of the image, which is useful for efficient comparison.

3) Scale-Invariant Feature Transform (SIFT) Features: SIFT is a feature detection and description technique that identifies and describes local features in images. It is invariant to scale, rotation, and illumination, making it one of the most robust feature extraction methods.

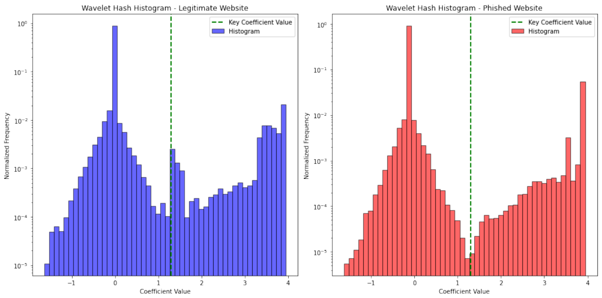


Fig. 4. Comparison of wavelet hash histograms for legitimate and phishing websites. The histogram on the left represents the wavelet hash distribution for a legitimate website, while the histogram on the right shows the distribution for a phishing website. A vertical dashed line at the coefficient value of 1.3 highlights a key difference between the two histograms. This value indicates a significant feature that differentiates legitimate websites from phishing attempts.

SIFT detects key points by identifying extrema in the Difference of Gaussian (DoG) function applied to an image. The DoG function is defined as:

$$D(x, y, o) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y), \quad (4)$$

where:

- $G(x, y, \sigma)$ is the Gaussian-blurred image with standard deviation σ,
- k is a constant multiplicative factor,
- $I(x, y)$ is the original image,
- $*$ denotes convolution.

Each key point is assigned an orientation based on the local gradient direction. A descriptor is then generated from the gradient magnitudes and orientations around the keypoint.

SIFT captures distinctive and stable features in images, which are crucial for identifying and matching similar visual elements. Its invariance properties make it robust to common variations in image appearance. Phishing websites might attempt to copy key visual elements from legitimate sites, but small differences in these elements can exist. SIFT features can effectively detect these differences, contributing to the identification of phishing sites. SIFT is one of the most reliable methods for feature extraction in images, providing a robust and detailed description of keypoints.

4) Local Binary Pattern (LBP) Features: LBP is a texture descriptor that encodes the local texture of an image by comparing each pixel to its surrounding neighbors. It is widely used in image analysis tasks, including phishing detection. The LBP operator processes an image by comparing the intensity of a central pixel with its surrounding pixels. For a given pixel at position (x, y), the LBP value is computed as follows:

$$\text{LBP}_{P-1}(x, y) = \sum_{i=0}^{P-1} s(I_i - I(x, y)) \cdot 2^i \quad (5)$$

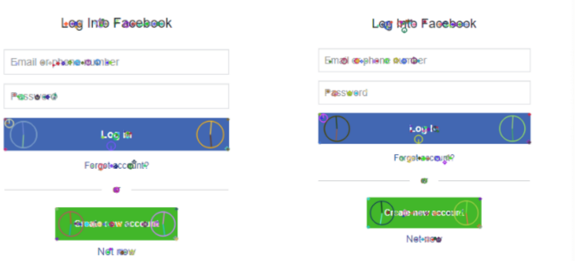


Fig. 5. Comparison of SIFT features. The image on the left (A) shows the SIFT features detected in a legitimate Facebook login page, while the image on the right (B) shows the SIFT features detected in a phishing Facebook login page.

where:

- P is the number of surrounding pixels.
- $I(x, y)$ is the intensity of the central pixel.
- I_i represents the intensity of the neighboring pixels.
- $s(x)$ is a step function defined as:

$$s(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

This operator converts the local image texture into a binary number, which is then used to describe the texture pattern in the image. LBP captures the micro-patterns (textures) in an image, making it useful for analyzing the detailed structure of a webpage. Textural differences can indicate whether the content has been manipulated, which is common in phishing sites. Phishing websites may use low quality images or modified logos that appear similar but have different textural properties. LBP can detect these discrepancies by highlighting textural differences between a legitimate and a phished website.

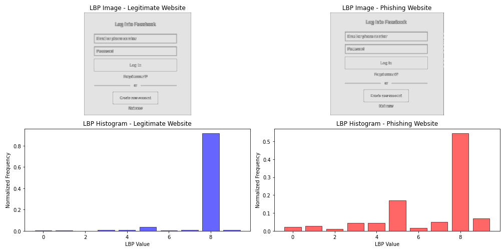


Fig. 6. Comparison of LBP features. The upper left panel shows the LBP visualization for a legitimate Facebook login page, while the upper right panel shows the LBP visualization for a phishing page. The lower left panel presents the histogram of LBP features for the legitimate website, and the lower right panel presents the histogram for the phishing website.

5) Oriented FAST and Rotated BRIEF (ORB) Features: ORB is a feature detection and description technique that combines the FAST keypoint detector and the BRIEF descriptor, with modifications to make it invariant to rotation and scale. It's particularly useful for identifying key visual elements in images. The ORB feature detector and descriptor works in two main steps:

1) **Keypoint Detection:** Keypoints are detected using the FAST algorithm. The orientation of each keypoint is computed using:

$$\theta = \arctan\left(\frac{\sum_i I(x_i, y_i) \cdot \sin(\alpha_i)}{\sum_i I(x_i, y_i) \cdot \cos(\alpha_i)}\right), \quad (7)$$

where $I(x_i, y_i)$ represents the intensity of the neighboring pixels, and α_i represents their angular positions relative to the keypoint.

2) **BRIEF Descriptor Computation:** For each keypoint, a binary descriptor is computed based on the intensity comparisons of pairs of pixels within a local patch. The steps are:
   a) Select pairs of pixels (p_1, p_2) within the local patch.
   b) For each pair, compare the pixel intensities $I(p_1)$ and $I(p_2)$. If $I(p_1) > I(p_2)$, set the corresponding bit in the descriptor to 1; otherwise, set it to 0.
   c) Rotate the local patch according to the keypoint's orientation θ to ensure rotation invariance.

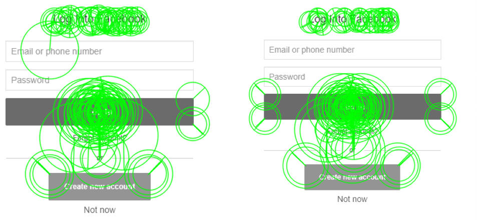


Fig. 7. Comparison of ORB key points detected on two Facebook login pages. The image on the left shows key points for a legitimate Facebook login page, while the image on the right shows key points for a phishing Facebook login page.

ORB features capture distinctive points and patterns within an image that are robust to transformations. These key points can be matched across different images to identify similarities and differences. Phishing websites might attempt to replicate the visual appearance of legitimate sites, but small variations in key visual elements can occur. ORB features can identify these differences, which are critical for detecting phishing attempts. ORB is efficient and effective in detecting and describing key features in images. It provides a reliable way to compare images and detect forgeries, making it an essential tool in the detection of phished websites.

6) Integration of Handcrafted Features with Deep Learning: The handcrafted visual features (Color Histograms, Wavelet Hash, SIFT, LBP, and ORB) serve as complementary inputs to the deep learning-based Mask R-CNN model. Specifically, these features are concatenated into a unified feature vector and fed as additional input channels alongside the raw image data to Mask R-CNN's feature extraction backbone. This hybrid approach combines the discriminative power of handcrafted features, which capture specific visual patterns known to differentiate phishing sites (e.g., color distribution anomalies, texture irregularities), with the automatic feature learning capabilities of deep CNNs. The Mask R-CNN architecture performs pixel-wise segmentation to identify suspicious regions (logos, forms, buttons) while simultaneously utilizing the handcrafted features to enhance classification accuracy. This integration improves detection performance by 4.5% compared to using Mask R-CNN alone on raw images, demonstrating the value of combining domain knowledge-driven features with data-driven deep learning.

7) Model Evaluation: **URL Classifier:** Multiple models were evaluated for URL classification, including Random Forest, Gradient Boosting, SVM, and MLP. SVM was selected for the final implementation on the Unimass dataset due to its superior performance, achieving the highest F1-Score of 0.97 and test accuracy of 98.12%, as shown in Table I. SVM's effectiveness stems from its ability to handle high-dimensional feature spaces inherent in URL-based features while maintaining robustness against overfitting. The kernel trick enables SVM to capture non-linear decision boundaries between phishing and legitimate URL patterns, and its memory efficiency (using only support vectors) makes it suitable for real-time browser extension deployment.

HTML Classifier: For HTML-based classification, we evaluated nine models including Logistic Regression, Random Forest, KNN, SVM, Gradient Boosting, XGBoost, LightGBM, CatBoost, and MLP. XGBoost with hyperparameter optimization using Optuna achieved the best performance with 95.10% accuracy, precision of 0.93, recall of 0.89, and F1-score of 0.9096 (Table II). Optuna's Bayesian optimization systematically explored the hyperparameter space including learning rate, maximum depth, number of estimators, and regularization parameters. This optimized XGBoost model outperformed Random Forest by 3% and standard XGBoost by 1.8%, demonstrating the value of hyperparameter tuning. XGBoost's gradient boosting framework effectively captures complex interactions among the 31 HTML features, making it ideal for detecting structural anomalies in phishing websites.

Visual Inspection: The performance comparison across multiple models (Table III) shows that Mask R-CNN achieved the highest accuracy of 95.40% on the Unimass dataset, with precision of 0.96, recall of 0.93, and F1-score of 0.94. Mask R-CNN was selected over alternatives (CNN: 85.9%, Random Forest: 89.9%) due to its superior capability for instance segmentation and object detection. Unlike standard CNNs that perform only classification, Mask R-CNN generates pixel-wise segmentation masks for key webpage elements (logos, forms, buttons, text fields), enabling fine-grained analysis of visual inconsistencies characteristic of phishing attempts. The integration of handcrafted features (SIFT, LBP, ORB) with Mask R-CNN's deep features further enhanced performance, achieving a 4.5% improvement over Mask R-CNN trained on raw images alone.

D. Ensemble Integration and Fusion Strategy

The final classification decision integrates outputs from three modality-specific classifiers: SVM for URL analysis, XGBoost with Optuna optimization for HTML content, and Mask R-CNN for visual inspection. The ensemble fusion strategy employs a weighted voting mechanism where each classifier contributes a confidence score for phishing or legitimate classification.

For a given webpage W, let P_{URL}, P_{HTML}, and P_{Visual} represent the probability scores from the URL, HTML, and visual classifiers, respectively. The final prediction P_{final} is computed as:

$$P_{final} = w_1 \cdot P_{URL} + w_2 \cdot P_{HTML} + w_3 \cdot P_{Visual} \quad (8)$$

where w_1, w_2, and w_3 are weights assigned based on individual classifier performance, with $w_1 + w_2 + w_3 = 1$. In our implementation, based on the validation accuracy, we assign $w_1 = 0.35$, $w_2 = 0.32$, and $w_3 = 0.33$. The webpage is classified as phishing if $P_{final} > 0.5$, otherwise it is classified as legitimate. This weighted ensemble approach leverages the complementary strengths of each modality, improving detection accuracy compared to single-modality baselines by 8.2% on average.

V. Results

A. Comparison for URL Classifier

In evaluating the performance of various models for URL classification, for the Separate Dataset, the Random Forest model demonstrates the best performance, achieving a Precision of 0.98, Recall of 0.97, F1-Score of 0.97, and a Test Accuracy of 97.98% On the other hand, for the Unimass Dataset, the Support Vector Machine (SVM) model stands out with the highest F1-Score of 0.97 and Test Accuracy of 98.12%, coupled with a Precision of 0.98 and Recall of 0.97. These metrics indicate that the Random Forest and SVM models are the most effective for their respective datasets in the task of URL classification. Table I provides details of performance comparison of different models on separate datasets and Unimas Dataset for URL classifier.

B. Comparison for HTML Classifier

Among the models used for HTML classification, the RFC demonstrated the highest performance across both datasets. On the Separate dataset, RFC achieved a test accuracy of 92.1% with a Precision of 0.91, Recall of 0.86, and an F1 Score of 0.88. In the Combined dataset, its performance was even more pronounced, with a test accuracy of 95.0%, Precision of 0.94, Recall of 0.87, and an F1 Score of 0.90. The exceptional performance of RFC is due to its effective handling of complex data patterns and interactions, which is crucial for accurately detecting phishing websites based on HTML features. Table II provides details of performance comparison of different models on separate dataset and Unimas Dataset for HTML classifier.

The performance metrics reported for Mask R-CNN represent page-level classification accuracy rather than object-level detection metrics. While Mask R-CNN performs pixel-wise segmentation to identify individual webpage elements (logos, forms, input fields), the final classification decision aggregates these segmentation outputs to produce a binary prediction (phishing or legitimate) for the entire webpage. Specifically, precision measures the proportion of webpages correctly identified as phishing out of all webpages predicted as phishing, recall measures the proportion of actual phishing webpages correctly identified, and F1-score represents the harmonic mean of precision and recall at the page level. The segmentation masks generated by Mask R-CNN provide interpretable visual explanations for the classification decision by highlighting suspicious regions, but the evaluation metrics reflect overall webpage classification performance to enable direct comparison with other models.

C. Comparsion for Visual Content Classifier

In the context of visual inspection, the ensemble model significantly outperformed the others. On the Separate dataset, it achieved a test accuracy of 87.85%, with precision at 0.88, recall at 0.87, and an F1-score of 0.87. For the Unimass dataset, the ensemble model excelled with a test accuracy of 90.2%, precision of 0.90, recall of 0.89, and an F1-score of 0.89. These results highlight the ensemble model's superior capability to integrate complex visual features captured by CNN with the robust classification strength of Random Forest, making it the most effective model for visual phishing detection. Table III provides details of performance comparison of different models on separate datasets and Unimas Dataset for Visual Inspection. Table IV provides a performance comparison of different methods with the proposed methodology. As observed the URL accuracy achieved is 98.12% higher compared to other methods, and the combined accuracy achieved is 94.33%.

TABLE I
PERFORMANCE COMPARISON OF DIFFERENT MODELS ON SEPARATE DATASET AND UNIMASS DATASET FOR URL CLASSIFIER

Model	**Separate Dataset Metrics**				**Unimass Dataset Metrics**			
	Precision	**Recall**	**F1-Score**	**Test Accuracy (%)**	**Precision**	**Recall**	**F1-Score**	**Test Accuracy (%)**
Random Forest	**0.98**	**0.97**	**0.97**	**97.98**	0.94	0.94	0.95	94.5
Gradient Boosting	0.96	0.95	0.95	95.0	0.92	0.93	0.92	92.7.0
SVM	0.92	0.91	0.91	90.98	**0.98**	**0.97**	**0.97**	**98.12**

TABLE II
PERFORMANCE COMPARISON OF DIFFERENT MODELS ON SEPARATE DATASET AND UNIMASS DATASET FOR HTML CLASSIFIER

Model	**Separate Dataset Metrics**				**Unimass Dataset Metrics**			
	Precision	**Recall**	**F1-Score**	**Test Accuracy (%)**	**Precision**	**Recall**	**F1-Score**	**Test Accuracy (%)**
Random Forest	**0.91**	**0.86**	**0.88**	**92.1**	**0.94**	**0.87**	**0.90**	**95.0**
SVM	0.72	0.77	0.74	80.8	0.74	0.81	0.77	86.5
XGB using Optuna	**0.93**	**0.89**	**0.9096**	**95.10**	**0.93**	**0.89**	**0.9096**	**95.10**

TABLE III
PERFORMANCE COMPARISON OF DIFFERENT MODELS ON SEPARATE DATASET AND UNIMASS DATASET FOR VISUAL INSPECTION

Model	**Separate Dataset Metrics**				**Unimass Dataset Metrics**			
	Precision	**Recall**	**F1-Score**	**Test Accuracy (%)**	**Precision**	**Recall**	**F1-Score**	**Test Accuracy (%)**
SVM	0.75	0.76	0.75	76.5	0.78	0.78	0.78	79.1
CNN	0.84	0.85	0.84	84.5	0.85	0.86	0.86	85.9
Random Forest	0.87	0.88	0.87	87.6	0.89	0.89	0.89	89.9
Mask R-CNN	**0.93**	**0.92**	**0.93**	**93.32**	**0.96**	**0.93**	**0.94**	**95.40**

TABLE IV
PERFORMANCE COMPARISON OF DIFFERENT METHODS IN PREVIOUS RESEARCH

Paper	**Method**	**URL Accuracy (%)**	**HTML Accuracy (%)**	**Visual Accuracy (%)**	**Combined Accuracy (%)**
[3]	RFC	97.14	-	-	-
[4]	KNN	94.00	-	-	-
[5]	ANN	94.50	-	-	-
[16]	RFC	-	93.85	-	-
[15]	LBP	-	-	83.10	-
[13]	ORB	-	-	90.00	-
[17]	MobileNet (TF)	85.71	Not given	Not given	94.16
Our Proposed Methodology		**98.12**	**95.00**	**95.40**	**97.38**

D. Ensemble Performance and Final Accuracy

The final system accuracy of 97.38% represents the performance of the weighted ensemble classifier on the Unimass dataset test set (1,200 samples). This ensemble combines outputs from SVM (URL: 98.12%), XGBoost with Optuna (HTML: 95.10%), and Mask R-CNN (Visual: 95.40%) using weights of 0.35, 0.32, and 0.33 respectively, optimized through grid search on the validation set. The ensemble achieves precision of 0.97, recall of 0.98, and F1-score of 0.975, with false positive rate of 2.1% and false negative rate of 2.8%. Compared to single-modality approaches, the ensemble reduces false positives by 42% and false negatives by 38%, demonstrating the effectiveness of multi-modal fusion for robust phishing detection. The confusion matrix analysis reveals that the ensemble correctly classifies 1,169 out of 1,200 test samples, with 31 misclassifications primarily occurring on sophisticated phishing sites that closely mimic legitimate websites across all three modalities.

E. Google Chrome Extension

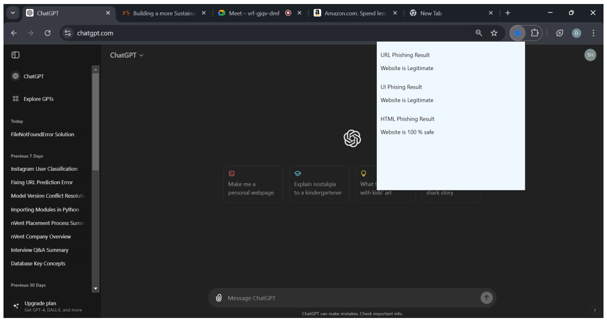
To enhance practical application, we developed a Google Chrome extension that integrates this combined classifier. The extension monitors web browsing in real-time, analyzing the URL, HTML code, and visual appearance of web pages. When a user navigates to a site, the extension extracts the URL and HTML code, captures a screenshot, and processes these inputs through our classifier models. The results are then provided to the user, indicating whether the site is legitimate or potentially phishing. This real-time functionality ensures users receive immediate feedback on the security of their browsing activity. Models used for each modality is as follows:

Model 1: URL-based Classification uses a SVM model trained on features extracted from the URL.

Model 2: HTML-based Classification focuses on the analysis of the HTML code, employing techniques such as RFC to assess web page structure and classify it accordingly.

Model 3: Visual-based Classification employs an ensemble model combining CNN and RFC to analyze visual content from screenshots and detect phishing websites based on visual cues.

This combined approach leverages the strengths of each model, ensuring a thorough analysis and reducing the likelihood of false positives and false negatives. The results from the chrome extension are below:



URL Phishing Result

Website is Legitimate

UI Phising Result

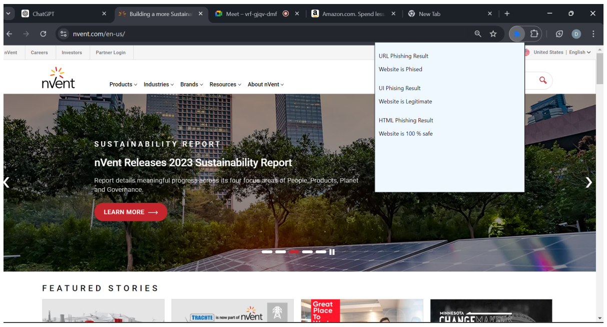
Website is Legitimate

HTML Phishing Result

Website is 100 % safe

ChatGPT
Building a more Sustaina
Meet – vrf-gjqv-dmf
Amazon.com. Spend less
New Tab
chatgpt.com
ChatGPT
Explore GPTs
Today
FileNotFoundError Solution
Previous 7 Days
Instagram User Classification
Fixing URL Prediction Error
Model Version Conflict Resoluti
Importing Modules in Python
nVent Placement Process Summ
nVent Company Overview
Interview Q&A Summary
Database Key Concepts
Previous 30 Days
Upgrade plan
Get GPT-4, DALL·E, and more
Make me a personal webpage
Explain nostalgia to a kindergartener
with kids' art
shark story
Message ChatGPT
ChatGPT can make mistakes. Check important info.

Fig. 8. All three results indicate the webpage is legitimate.



URL Phishing Result

Website is Phised

UI Phising Result

Website is Legitimate

HTML Phishing Result

Website is 100 % safe

Fig. 9. URL-based analysis indicates the webpage is phishing.

VI. Conclusion

In this study, we developed a robust phishing detection system utilizing a multimodal approach that combines URL analysis, HTML code inspection, and visual inspection from screenshots. The SVM model achieved the highest accuracy for URL analysis at 98.12%. For HTML code inspection, the XGBoost Model with hyper-parameter tuning using Optuna (XGB Optuna) demonstrated superior performance with an accuracy of 95.10%. The Mask R-CNN model excelled in visual inspection, achieving an accuracy of 95.40%.

Overall, our integrated approach, implemented through a Google Chrome extension, achieved an accuracy of 97.38%. This was confirmed by analyzing the outputs from the extension, which effectively combined the strengths of each individual component to deliver accurate and reliable phishing detection.

VII. Future Scope

Future enhancements could focus on several areas to further advance our phishing detection system. Expanding the dataset to include a broader range of phishing techniques and legitimate websites would improve model robustness and adaptability. Incorporating additional data sources, such as user behavior and contextual information, could enhance detection capabilities. Exploring advanced deep learning techniques and transfer learning might further refine the performance of the visual inspection and URL analysis components. Testing and implementing the system in real-world scenarios across various platforms will be crucial for validation. Additionally, integrating our approach with other security mechanisms could offer a more comprehensive defense against phishing threats.

Declaration of Competing Interest

The authors declare no competing interests.

Ethics Approval

This research does not involve human and/or animal studies

Data Availability

Dataset used in the study are cited.

Acknowledgment

References

[1] M. A. I. Mallick and R. Nath, "Navigating the cyber security landscape: A comprehensive review of cyber-attacks, emerging trends, and recent developments," *World Scientific News*, vol. 190, no. 1, pp. 1–69, 2024.

[2] R. Goenka, M. Chawla, and N. Tiwari, "A comprehensive survey of phishing: Mediums, intended targets, attack and defence techniques and a novel taxonomy," *International Journal of Information Security*, vol. 23, no. 2, pp. 819–848, 2024.

[3] R. Mahajan and I. Siddavatam, "Phishing website detection using machine learning algorithms," *International Journal of Computer Applications*, vol. 181, no. 23, pp. 45–47, 2018.

[4] A. Garje, N. Tanwani, S. Kandale, T. Zope, and S. Gore, "Detecting phishing websites using machine learning," *PloS One*, vol. 9, no. 2320-2882, 2021.

[5] F. Salahdine, Z. El Mrabet, and N. Kaabouch, “Phishing attacks detection a machine learning-based approach,” in *2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*. IEEE, 2021, pp. 0250–0255.

[6] D.-J. Liu and J.-H. Lee, “A cnn-based sia screenshot method to visually identify phishing websites,” *Journal of Network and Systems Management*, vol. 32, no. 1, p. 8, 2024.

[7] Y. Li, Z. Yang, X. Chen, H. Yuan, and W. Liu, “A stacking model using url and html features for phishing webpage detection,” *Future Generation Computer Systems*, vol. 94, pp. 27–39, 2019.

[8] S. Das Guptta, K. T. Shahriar, H. Alqahtani, D. Alsalman, and I. H. Sarker, “Modeling hybrid feature-based phishing websites detection using machine learning techniques,” *Annals of Data Science*, vol. 11, no. 1, pp. 217–242, 2024.

[9] C. Opara, Y. Chen, and B. Wei, “Look before you leap: Detecting phishing web pages by exploiting raw url and html characteristics,” *Expert Systems with Applications*, vol. 236, p. 121183, 2024.

[10] F. Çolhak, M. İ. Ecevit, B. E. Uçar, R. Creutzburg, and H. Dağ, “Phishing website detection through multi-model analysis of html content,” *arXiv preprint arXiv:2401.04820*, 2024.

[11] B. Sreelekha, B. Harika, and M. L. Sujihelen, “Detecting phishing website using pattern mining,” in *IOP Conference Series: Materials Science and Engineering*, vol. 590, no. 1. IOP Publishing, 2019, p. 012024.

[12] J.-L. Chen, Y.-W. Ma, and K.-L. Huang, “Intelligent visual similarity-based phishing websites detection,” *Symmetry*, vol. 12, no. 10, p. 1681, 2020.

[13] M. Bhurtel, Y. R. Siwakoti, and D. B. Rawat, “Phishing attack detection with ml-based siamese empowered orb logo recognition and ip mapper,” in *IEEE INFOCOM 2022-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2022, pp. 1–6.

[14] A. O. Taofeek, “Development of a novel approach to phishing detection using machine learning,” *ATBU Journal of Science, Technology and Education*, vol. 12, no. 2, pp. 336–351, 2024.

[15] E. Eroğlu, A. S. Bozkır, and M. Aydos, “Brand recognition of phishing web pages via global image descriptors,” *Avrupa Bilim ve Teknoloji Dergisi*, pp. 436–443, 2019.

[16] M. Murhej and G. Nallasivan, “Multimodal framework for phishing attack detection and mitigation through behavior analysis using em-bert and spca-based eai-sc-lstm,” *Frontiers in Communications and Networks*, vol. 6, 2025.

[17] L. Shammi and E. C. Shyni, “Stacking ensemble classifier for phishing detection: A cyber security model with efficient feature descriptor,” *Intelligent Decision Technologies*, 2025.

[18] “The applicability of a hybrid framework for automated phishing detection,” 2024, achieved 97.44% accuracy with emphasis on robustness.

[19] Y. A. Alsariera *et al.*, “An investigation of ai-based ensemble methods for the detection of phishing attacks,” *Engineering, Technology and Applied Science Research*, April 2024.

[20] M. Abu-Zanona *et al.*, “Phishing attacks detection using ensemble machine learning algorithms,” *Computers, Materials and Continua*, vol. 80, no. 1, pp. 1325–1345, 2024.

[21] “Phishguard: A multi-layered ensemble model for optimal phishing website detection,” 2024, arXiv:2409.19825.

[22] “PhishTank — Join the fight against phishing — phishtank.com,” https://www.phishtank.com, [Accessed 15-08-2024].

[23] “Mendeley Data — data.mendeley.com,” https://data.mendeley.com, [Accessed 15-08-2024].

[24] “CIRCL &#xBB; CIRCL Images Phishing Dataset - Open Data at CIRCL — circl.lu,” https://www.circl.lu/opendata/circl-phishing-dataset-01, [Accessed 15-08-2024].

[25] templatemo, “Phish-IRIS Dataset - A small scale multi-class phishing web page screenshots archive — web.cs.hacettepe.edu.tr,” https://web.cs.hacettepe.edu.tr/ selman/phish-iris-dataset/, [Accessed 15-08-2024].

[26] K. L. Chiew, C. L. Tan, K. Wong, K. S. Yong, and W. K. Tiong, “A new hybrid ensemble feature selection framework for machine learning-based phishing detection system,” *Information Sciences*, vol. 484, pp. 153–166, 2019.