Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

A Multi-Modal Phishing Detection System

Amrit Lal P Siva
Dept. of CSE
FISAT, Angamaly,Kerala, India

Ashin Shibu
Dept. of CSE
FISAT, Angamaly,Kerala, India

Dinil P S
Dept. of CSE
FISAT, Angamaly,Kerala, India

Hansa J Thattil
Dept. of CSE
FISAT, Angamaly,Kerala, India

Dhanraj Latish
Dept. of CSE
FISAT, Angamaly,Kerala, India

Abstract - Phishing attacks have evolved significantly, employing visual mimicry, semantic deception, and network-level manipulation to bypass traditional detection systems. Conventional approaches based on URL blacklists or single-modal feature analysis often fail against zero-day and dynamically generated phishing pages. This paper presents a multi-modal phishing detection framework that integrates URL lexical features, HTML/DOM structural attributes, visual cues, semantic content, and network-based indicators. Structured features are processed using a stacked ensemble model comprising Logistic Regression, LightGBM, and Linear SVM classifiers. Webpage screenshots are analyzed using a fine-tuned EfficientNet-B0 model to extract visual embeddings, while semantic representations are generated using DeBERTa-v3 Base to identify deceptive language patterns. These heterogeneous features are fused through dense neural layers to produce a final phishing probability score. The system incorporates cost-sensitive learning to address class imbalance and integrates explainability mechanisms, including Grad-CAM visualization and DOM-level feature highlighting. The proposed architecture aims to deliver a scalable, adaptive, and interpretable solution for detecting modern phishing attacks across multiple content modalities.

Keywords
Phishing Kits, HTML Analysis, Machine Learning, Deep Learning, Zero-Day Attacks, Continuous Adaptation.

1. INTRODUCTION

Phishing attacks represent one of the most pervasive and financially destructive threats in the contemporary digital landscape, responsible for substantial data breaches and financial losses across individuals and organizations globally [1], [2]. The sophistication of these campaigns has dramatically increased due to the widespread availability and efficiency of phishing kits [3]. These kits are pre-built, ready-to-deploy packages that allow attackers to generate and launch numerous fake websites targeting various brands within minutes, dramatically reducing the barrier to entry for cybercriminals. The core challenge in defense is that while these kits may be adapted to target different legitimate entities (brands), the underlying structure, or HTML code architecture, often remains consistent across pages generated by the same kit [3].

Traditional phishing detection methods, such as URL filtering, blacklist comparison, and visual similarity analysis, are struggling to keep pace with this evolution and are becoming permanently outdated [4]. These conventional techniques typically rely on previously known data or features that are easily altered by attackers through obfuscation, leading to the failure to identify new or zero-day phishing attacks [5]. There is a critical and unmet need for a self-learning, adaptive detection system capable of analyzing the persistent structural features of web pages, rather than easily modifiable content, to automatically identify and classify emerging phishing kits without human intervention. This necessity forms the foundation for this survey and the subsequent proposal of the Phish Eye framework.

The scope and objectives of this survey are as follows:

A. Develop a multi-modal phishing detection system integrating URL, HTML, visual, semantic, and network features.
B. Extract and analyze structured webpage features using machine learning techniques.
C. Detect visual phishing patterns using a fine-tuned EfficientNet-B0 model.
D. Identify deceptive webpage content using DeBERTa-v3 semantic analysis.
E. Improve detection performance using weighted decision-level fusion of multiple models.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

2. LITERATURE REVIEW

In this work, we propose a multi-modal phishing detection system that integrates URL, HTML/DOM, visual, semantic, and network-based features to accurately identify malicious websites [5]. The proposed architecture combines structured feature analysis with deep learning models, including EfficientNet-B0 for screenshot-based visual detection [6] and DeBERTa-v3 Base for semantic text analysis [7] along with a stacked ensemble model for structured data processing. A fusion mechanism aggregates these heterogeneous feature representations to generate a final phishing probability score. The system also incorporates explainability modules such as visual saliency maps and DOM-based highlighting to enhance transparency and interpretability [8]. This approach aims to improve robustness, detection accuracy, and adaptability against evolving phishing attacks.

Venturi et al. (2022) [3] proposed a three-phase framework designed for identifying and classifying phishing kits to enable early detection by platform providers. The authors used static code inspection to detect artifacts like .htaccess rules, suspicious PHP scripts, and obfuscated resource references that are typically excluded from visual rendering. These features were then summarized and used to train supervised classifiers, including Random Forest and SVM, achieving an F1-score of 0.9 for evasive kits. While the system highlights template-level commonalities as a practical signal, the method relies entirely on static patterns. Furthermore, static analysis cannot capture execution-time behaviors or novel dynamic evasion techniques, which limits its effectiveness against progressively adaptive adversaries who employ server-side randomization.

Orunsolu and Sodiya [9] introduced an Anti-Phishing Kit (APK) scheme utilizing a two-stage modular architecture designed to detect toolkits. The first stage employs a Sorter Module to isolate potentially malicious pages by detecting login-related fields and specific obfuscated content snippets that characterize kit deployed sites. The second stage applies a hybrid Naive Bayes and SVM classifier based on 18 heuristic features, including SSL validation and domain age. Although the system offers fast detection speeds of approximately 0.3 seconds per page, it is restricted by its reliance on predefined heuristics. This makes the system vulnerable to randomized or zero-day phishing kit structures that do not match the static signature set, necessitating frequent manual updates to the rule base.insufficient for fine grained kit attribution, leaving a gap for more advanced structural signature methods.

Lu et al. (2022) [10] proposed a homology-based approach that emphasizes structural and visual similarity analysis using Document Object Model (DOM) and visual layout comparison. The system incorporates three primary modules Data Collection, Homology Analysis, and Classification to measure similarity at both structural and visual levels. This dual layered approach is designed to identify "cloned" sites that look identical to the naked eye but share deep code-level similarities. While effective for forensics and brand impersonation detection, the extensive structural and visual comparisons increase computational complexity and latency significantly. Furthermore, the method struggles with dynamically generated content and structurally distinct interfaces used to mislead detectors, often requiring high-resource overhead for real time monitoring.

Purwanto et al. [11] presented PhishSim, a feature-free detection framework that leverages Normalized Compression Distance (NCD) to measure similarity without manual feature extraction. The system uses an incremental prototype-based classification strategy, allowing the model to adapt to new samples obtained from blacklists and user reports by comparing their compressed bitstreams. Although it achieved an AUC of 96.68\%, PhishSim encounters limitations when handling zero-day attacks with completely novel HTML structures that lack prior prototypes in the compression dictionary. Additionally, maintaining and optimizing the prototype database introduces extra computational overhead in large scale environments, particularly as the number of known phishing kit families grows.Deep learning was further explored by Opara et al.through the WebPhish framework, which accepts raw URL and HTML content simultaneously. By leveraging convolutional layers to extract hierarchical semantic patterns, the model eliminates the need for manual feature engineering and identifies correlations between URL naming conventions and page structures. While achieving 98.1\% accuracy, the architecture is computationally intensive, with training times averaging nearly 491 seconds per epoch during development. This lack of efficiency poses challenges for real time edge device deployment where computational resources are limited and low latency responses are required for user protection.

Uddin et al. (2020) [12] developed HTMLPhish, a deep learning framework designed to detect phishing web pages solely through HTML source code analysis. By utilizing a hybrid architecture combining Convolutional Neural Networks (CNN) and Bidirectional LSTM networks, the model captures both local tag-level dependencies and sequential long range relationships within the code. This embedding-based representation

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

allows the model to ignore textual content and focus on the hierarchical arrangement of HTML elements. The system demonstrated strong temporal stability with only a 4\% decline in accuracy over two months. However, the model remains sensitive to heavily obfuscated or minified HTML, and its dependence on static code analysis reduces its effectiveness against dynamic, script based phishing pages that render content client-side.

Deep learning was further explored by Opara et al. (2024) [13] through the WebPhish framework, which accepts raw URL and HTML content simultaneously. By leveraging convolutional layers to extract hierarchical semantic patterns, the model eliminates the need for manual feature engineering and identifies correlations between URL naming conventions and page structures. While achieving 98.1\% accuracy, the architecture is computationally intensive, with training times averaging nearly 491 seconds per epoch during development. This lack of efficiency poses challenges for real time edge device deployment where computational resources are limited and low latency responses are required for user protection.

Sameen et al. (2020) [14] introduced PhishHaven, an AI-driven system designed for real time URL analysis using a parallel ensemble of ten classifiers. The system expands shortened URLs and extracts seventeen lexical features, such as URL length and special character counts, for concurrent processing across multi-threaded models. A 67\% majority voting scheme is used to finalize detection, resulting in an accuracy of 98\%. Nevertheless, the system is limited to lexical feature analysis, which restricts its understanding of deeper structural relationships and requires constant retraining to adapt to new URL obfuscation methods used by kit developers.

Zara et al. (2023) [15]proposed a comprehensive phishing detection framework integrating ML, DL, and ensemble learning. The study emphasizes feature selection techniques like Information Gain (IG) and Principal Component Analysis (PCA) to reduce dimensionality and eliminate redundant data before the training phase. Among the evaluated models, Random Forest achieved the highest accuracy of 99.0\% due to its ability to handle non linear feature relationships. However, the system's reliance on static, predefined attributes poses limitations when confronting dynamically generated or behaviorally deceptive phishing pages that intentionally vary their attribute sets to bypass threshold based detectors.

Castaño et al. (2023) [16] introduced PhiKitA, a comprehensive dataset linking phishing websites with their generated kits. This dataset establishes a verified ground truth by associating each phishing website with its originating kit, supporting the evaluation of multi class classification and clustering algorithms. The study identified that many phishing kit families share identical structural frameworks while targeting multiple diverse brands. While providing a critical benchmark for forensics and campaign tracking, the performance of models trained on this data decreases in multi-class setups. This indicates that existing feature sets are often insufficient for fine grained kit attribution, leaving a gap for more advanced structural signature methods.

From the reviewed works, we derive several key insights that guide our proposed system design. Inspired by Venturi et al. and Uddin et al., we incorporate structured HTML and DOM-level analysis to capture template-based similarities. The modular detection strategy discussed by Orunsolu and Sodiya motivates our layered architecture design. The structural and visual similarity concepts from Lu et al. support our inclusion of visual feature learning through deep models. The adaptive and incremental learning perspective presented by Purwanto et al. highlights the importance of scalability and prototype awareness. Deep representation learning approaches from Opara et al. and ensemble-based optimization strategies from Zara et al. motivate our fusion-based architecture. Additionally, the dataset-level insights from Castaño et al. emphasize the significance of structural signatures for fine-grained attribution. Collectively, these ideas inform our decision to design a multi-modal, stacked ensemble-based detection framework that integrates structural, semantic, visual, and network-level analysis with enhanced explainability.

3. METHODOLOGY

3.1 Overview

The proposed methodology (fig:1) consists of **six major steps** : data preparation, data balancing, feature extraction, multi-modal model training, fusion and classification, and explainability with evaluation. The system performs **five types of feature extraction**:URL features, HTML/DOM structural features, visual features from webpage images, semantic text features, and network-level features. For classification, it employs **three primary model components**: (1) a stacked ensemble classifier (Logistic Regression, LightGBM, and Linear SVM) for structured URL/HTML and network features, (2) EfficientNet-B0 for visual feature learning, and (3) DeBERTa-v3 Base for semantic analysis. These outputs are fused through

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

dense neural layers to produce the final phishing probability score.

3.2 Dataset and Preprocessing

The phishing detection dataset was constructed using HTML source codes collected from the Mendeley Phishing Websites dataset along with additional URL, image, and network-related attributes obtained from publicly available repositories. During preprocessing, corrupted files, inaccessible pages, and incomplete records were removed to ensure data quality and reliability. The final dataset consists of labeled instances categorized as phishing and legitimate.

3.2.1 HTML Parsing and Structural Cleaning

Raw HTML files often contain redundant scripts, comments, and inconsistent formatting. To standardize structural analysis, the following preprocessing steps were applied:

- Parsing HTML to construct the Document Object Model (DOM) tree
- Extraction of forms, input fields, scripts, hyperlinks, and metadata.
- Removal of unnecessary comments and irrelevant tags.
- Normalization of extracted structural attributes

These steps ensure consistent structural representation for downstream feature extraction.

3.2.2 Text Content Extraction and Normalization

Visible textual content was extracted from rendered webpages while excluding hidden elements and script-based content. The normalization process included:

- Lowercasing of textual content
- Removal of extra whitespaces and special formatting characters
- Cleaning of non-informative symbols

This preprocessing improves semantic feature learning and reduces textual noise.

3.2.3 Image Pre-Processing

To prepare visual inputs for the deep learning model, screenshots of each webpage were first collected and then subjected to image pre-processing. The captured images were resized to a fixed resolution compatible with the convolutional neural network architecture. This step ensures uniform image dimensions, reduces computational complexity, and improves the efficiency of feature extraction during the visual analysis stage.

3.2.4 Feature Structuring

After preprocessing, each sample was organized into a unified structured format containing URL information, raw HTML, extracted DOM features, cleaned text content, webpage screenshot, network-related attributes, and the corresponding label. This structured representation enables effective multi-modal feature extraction across lexical, structural, semantic, visual, and network dimensions.

3.3 System Architecture

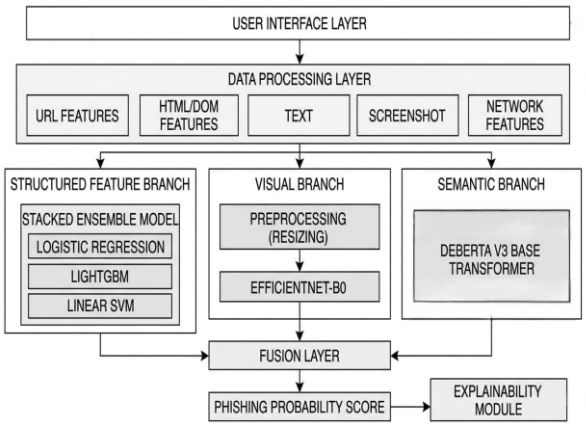

Figure 1 : System Architecture

The proposed multi-modal phishing detection system architecture (see Figure 1) consists of sequential processing stages:

- URL submission and webpage acquisition.
- Data preprocessing and structured feature construction.
- Multi-modal feature extraction (Structured, Visual, Semantic, Network).
- Model training and fusion-based classification.
- Performance evaluation and explainability generation.

The architecture branches into three parallel feature extraction pipelines:

Structured Feature Pipeline:URL lexical features, HTML/DOM structural features, and network-related attributes are extracted and processed using a stacked ensemble model comprising Logistic Regression, LightGBM, and Linear SVM. This branch captures statistical, structural, and rule-based phishing indicators and produces a structured feature vector.

Visual Pipeline : Webpage screenshots are provided by the user as input to the system. The input images are first

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

resized to a fixed resolution and then passed to a fine-tuned EfficientNet-B0 model. The network extracts visual embeddings representing phishing-specific design cues such as fake login forms, brand impersonation layouts, and suspicious UI elements.

Semantic Pipeline : Visible webpage text is extracted and tokenized before being fed into a DeBERTa-v3 Base transformer model. This branch learns contextual representations to detect deceptive language patterns, urgency-based messages, and credential-harvesting phrases.

The feature representations from all three pipelines are concatenated in a fusion layer, followed by fully connected dense layers with dropout regularization. A sigmoid output layer generates the final phishing probability score.

The system is trained using class-weighted binary cross-entropy loss with the Adam optimizer and evaluated using accuracy, precision, recall, F1-score, ROC-AUC, and confusion matrix metrics. Additionally, explainability modules such as Grad-CAM, DOM suspicious tag highlighting, and semantic keyword attribution provide interpretable insights into model predictions.

3.4 Feature Extraction Modules

The proposed system performs (see Figure 1) multi-modal feature extraction to capture diverse characteristics of phishing websites across structural, lexical, visual, semantic, and network dimensions. This comprehensive strategy ensures that both surface-level patterns and deep contextual relationships are effectively learned.

3.4.1 URL Feature Extraction

Lexical features are extracted from the URL to identify suspicious patterns commonly associated with phishing attacks. These include URL length, presence of special characters, token entropy, use of IP addresses instead of domain names, excessive subdomains, and suspicious keywords. Such features help detect obfuscation strategies frequently used by attackers.

3.4.2 HTML and DOM Feature Extraction

Structural features are derived from the parsed HTML and DOM tree. The system analyzes elements such as the number of forms, input fields, iframes, external resource links, hidden elements, and obfuscated JavaScript code. Additionally, abnormal form actions, mismatched domain references, and suspicious redirection behaviors are identified. These features capture template-level and structural similarities commonly found in phishing kits.

3.4.3 Visual Feature Extraction

To capture layout-level deception and brand impersonation cues, webpage screenshots provided by the user are used as visual inputs. The images are pre-processed and passed to a fine-tuned EfficientNet-B0 convolutional neural network. The model extracts high-level visual embeddings representing color distribution, logo placement, input field alignment, and overall page structure. These visual patterns help detect cloned or visually deceptive pages.

3.4.4 Semantic Feature Extraction

Visible textual content extracted from webpages is encoded using the DeBERTa-v3 Base transformer model. This model captures contextual meaning, deceptive language patterns, urgency cues, and impersonation-related phrases. Unlike traditional bag-of-words techniques, transformer-based embeddings preserve semantic relationships and long-range dependencies within the text.

The proposed multi-modal phishing detection framework provides a comprehensive and effective solution by integrating structural, lexical, visual, semantic, and network-level features within a unified architecture. By combining deep learning models with a stacked ensemble classifier, the system captures both surface-level anomalies and deeper contextual deception patterns. The fusion-based approach improves detection accuracy, strengthens generalization against zero-day attacks, and enhances robustness against evolving phishing strategies, while the inclusion of explainability mechanisms increases transparency and practical applicability.

3.5 Classification Models

The proposed framework adopts a hybrid classification strategy that integrates both traditional Machine Learning (ML) models and Deep Learning (DL) models to achieve robust multi-modal phishing detection.

1. ***Machine Learning Models (Structured Feature Classification)***:For structured features such as URL, HTML/DOM, and network attributes, a stacked ensemble of traditional ML cla)sifiers is employed.
   - *Logistic Regression (LR)**:*** A linear classifier that models the relationship between extracted numerical features and the phishing label. It provides fast inference and strong baseline performance.
   - *LightGBM*: A gradient boosting decision tree model capable of capturing complex non-linear

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

relationships within structured data. It is efficient and performs well on high-dimensional tabular datasets.

- *Linear Support Vector Machine (SVM):* A margin-based classifier that maximizes class separation and improves generalization, particularly effective in high-dimensional feature spaces.

These classifiers are combined using a stacking strategy to leverage their complementary strengths and improve overall structured feature classification performance.

2. **Deep Learning Models (Representation Learning):** Deep learning models are utilized for learning high-level representations from visual and textual data.
   - *EfficientNet-B0*: A convolutional neural network used for visual classification of webpage screenshots. It learns layout structures, logo placement, design similarity, and visual deception patterns.
   - *DeBERTa-v3 Base*: A transformer-based model used for semantic text classification. It captures contextual meaning, deceptive language patterns, and long-range dependencies within webpage content.
   - *Fusion Neural Network (Dense Layers):* The embeddings generated from ML and DL components are concatenated and passed through fully connected dense layers with dropout regularization. A final sigmoid activation function produces the phishing probability score.

The integration of ML and DL models enables efficient processing of structured data while leveraging deep neural networks for complex visual and semantic pattern recognition, resulting in a robust and adaptive phishing detection framework.

3.6 System Working Methodology

The proposed multi-modal phishing detection system (see Figure 1) operates through a layered architecture where each layer performs a specific task in the detection pipeline. The workflow ensures systematic data transformation from raw URL input to final classification with interpretability.

3.6.1 User Interface Layer

The system begins at the user interface layer, implemented using a Flask-based web application. This layer allows users to submit a target URL for phishing analysis. Upon submission, the request is forwarded to the backend processing engine. After inference, the interface displays the classification result, phishing probability score, and explainability outputs including visual and structural highlights. This layer ensures accessibility and real-time interaction with the detection system.

3.6.2 Data Acquisition and Processing Layer

Once a URL is received, the system initiates automated webpage acquisition. The HTML source code is downloaded and rendered using a headless browser to accurately capture dynamic content. During this stage:

- The HTML document is parsed to construct the DOM tree.
- Visible textual content is extracted from the rendered page.
- A full-page screenshot is generated for visual analysis.
- URL lexical and network-related attributes are computed.

This layer transforms raw webpage data into structured components suitable for multi-modal feature extraction.

3.6.3 Structured Feature Extraction Layer

This layer processes numerical and categorical features derived from URL, DOM, and network attributes. URL features include lexical patterns and suspicious substrings. DOM features capture structural anomalies such as hidden elements, unusual form actions, and embedded scripts. Network features analyze redirection chains and external resource behavior.

These features are fed into a stacked ensemble model consisting of Logistic Regression, LightGBM, and Linear SVM to generate a consolidated structured feature representation.

3.6.4 Visual Feature Extraction Layer

In this layer, the captured webpage image is resized and normalized before being passed to a fine-tuned EfficientNet-B0 model. The convolutional neural network extracts visual embeddings representing phishing-related design cues such as fake login forms, brand impersonation layouts, and misleading interface elements. This layer enables detection of visually deceptive webpages.

3.6.5 Semantic Feature Extraction Layer

The cleaned textual content extracted from the webpage is processed in this layer. Text is tokenized and provided to a DeBERTa-v3 Base transformer model. The model generates contextual embeddings that capture deceptive language patterns, urgency-based messages, credential-

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

harvesting phrases, and brand impersonation indicators. This layer strengthens detection by analyzing semantic intent.

3.6.6 Fusion and Classification Layer

The embeddings from the structured, visual, and semantic layers are concatenated to form a unified feature vector. This fused representation is passed through fully connected dense layers with dropout regularization to enhance generalization. The final output layer uses a sigmoid activation function to compute the phishing probability score. Based on a predefined threshold, the webpage is classified as either *phishing* or *legitimate*.

3.6.7 Explainability Layer

To enhance transparency, the system integrates an explainability module. Grad-CAM is applied to the visual model to highlight influential image regions. Suspicious HTML tags and form elements are marked in the structural analysis. Additionally, the semantic model provides keyword attribution to identify deceptive phrases. These explanations are displayed to the user through the interface, improving trust and interpretability.

4. RESULTS AND DISCUSSION

4.1 Phishing Detection Model Performance

The proposed phishing detection system was evaluated using multiple modalities including URL-based features, HTML semantic analysis, and visual webpage analysis. Model performance was assessed using Accuracy, Precision, Recall, Confusion Matrix, and ROC-AUC metrics.

Table 1 : Results

Model	Accuracy (%)	Precision (%)	Recall (%)
Logistic Regression	99.57	99.66	9948
LightGBM	99.67	99.82	99.52
Linear SVM	99.58	99.72	99.44
Network Features	99.0	-	-
DeBerta-v3	87.0	-	-
EfficientNet-B0	74.0	-	-

Table 1 : Results presents the comparative performance of the implemented classification models.

From the results, it can be observed that classical machine learning models trained on structured URL, HTML, and network features achieved very high classification accuracy. Logistic Regression and Linear SVM demonstrated strong performance due to their ability to effectively model structured phishing indicators such as suspicious URL tokens, abnormal form actions, and embedded scripts.

LightGBM achieved the highest accuracy among the evaluated models, demonstrating the effectiveness of gradient boosting in capturing complex relationships among engineered phishing features.

Deep learning models trained on raw webpage content showed comparatively lower performance. The transformer-based DeBERTa-v3 model achieved 87% accuracy by analyzing semantic content and identifying deceptive language patterns used in phishing websites.

Similarly, the EfficientNet-B0 model trained on webpage screenshots achieved 74% accuracy, indicating that visual phishing cues alone are insufficient for reliable detection but still provide useful signals when combined with other modalities.

4.2 Confusion Matrix Analysis

To further analyze classification performance, confusion matrices were generated for different detection modules including URL-based models, HTML semantic models, and image-based models.

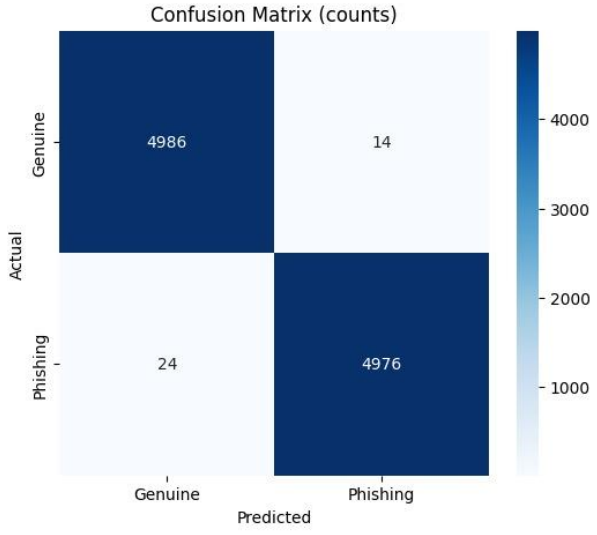

Figure 2 : Confusion matrix for the URL

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

The confusion matrix for the URL-based model (see Figure 2) shows strong diagonal dominance, with 4986 correctly classified legitimate webpages and 4976 correctly classified phishing webpages. Only a small number of misclassifications were observed, indicating high reliability of structured feature-based detection.

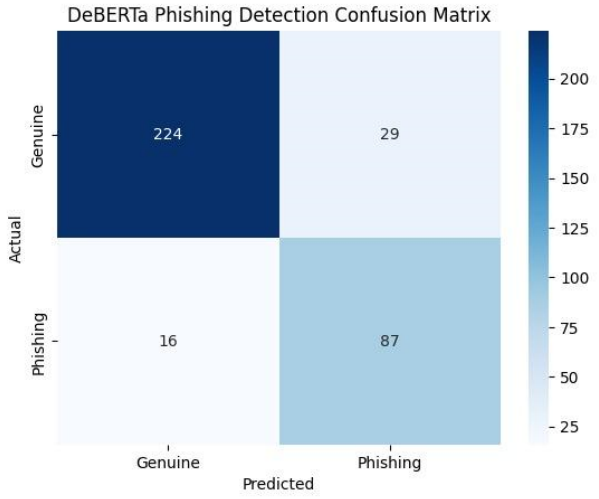

Figure 3 : Confusion matrix for the HTML

The HTML-based transformer model (see Figure 3) demonstrates good classification capability, correctly identifying 224 legitimate pages and 87 phishing pages. However, several false positives and false negatives are observed, highlighting the difficulty of detecting phishing solely from webpage textual content.

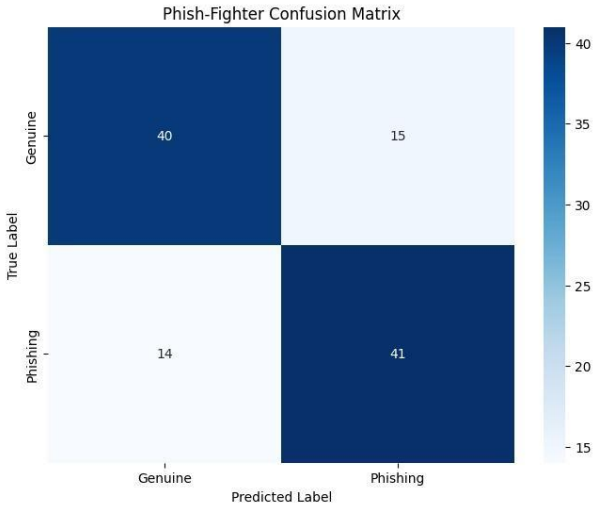

Figure 4 : Confusion matrix for the Image

The image-based model (see Figure 4) shows moderate performance. While it successfully detects several phishing interfaces, visual similarity between legitimate and phishing websites introduces classification challenges, leading to a higher number of misclassifications.

4.3 ROC Curve Analysis

Receiver Operating Characteristic (ROC) curves were used to evaluate the discriminative capability of the proposed phishing detection system.

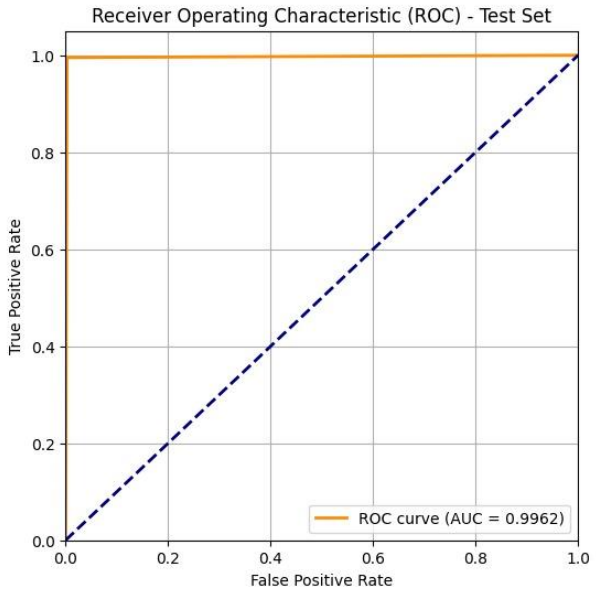

Figure 5 : ROC curve for the URL

The ROC curve for the URL-based detection system (see Figure 5) demonstrates strong classification performance with an Area Under the Curve (AUC) value of approximately 0.9962. This indicates that the model achieves excellent discrimination between phishing and legitimate webpages while maintaining a low false positive rate.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

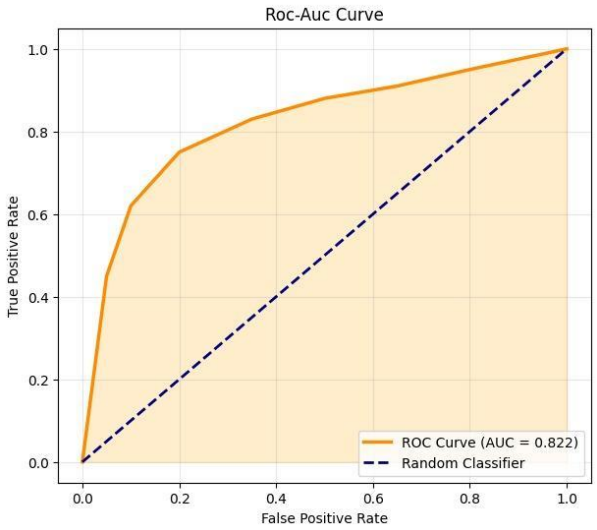

Figure 6 : ROC curve for the HTML

The ROC curve for the HTML semantic model (see Figure 6) illustrates the ability of the transformer-based DeBERTa-v3 architecture to discriminate between phishing and legitimate webpages using textual and structural HTML content.

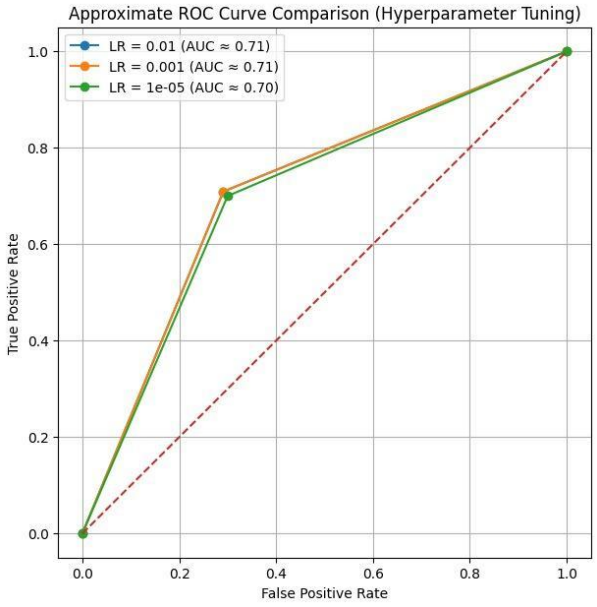

Figure 7 : ROC curve for the Image

The ROC curve for the image-based model (see Figure 7) demonstrates the performance of EfficientNet-B0 in identifying phishing websites from webpage screenshots. The model achieved an AUC value of approximately 0.82, which indicates a reasonable ability to capture visual phishing indicators such as fake login forms, cloned branding, and suspicious page layouts.

4.4 Comparative Discussion

The experimental results reveal several important insights:

- Structured feature-based machine learning models significantly outperform deep learning models when trained on phishing datasets containing strong engineered attributes.
- Gradient boosting methods such as LightGBM effectively capture complex interactions among URL and DOM-based features, resulting in the highest classification accuracy.
- Transformer-based semantic models improve contextual understanding but require larger datasets to fully exploit contextual embeddings.
- Image-based detection using webpage screenshots provides complementary information but performs poorly when used independently due to visual similarity between legitimate and phishing pages.
- The integration of URL features, HTML semantic analysis, network attributes, and visual information enables a robust multi-modal phishing detection system capable of achieving high detection accuracy.

Overall, the experimental results validate the effectiveness of the proposed multi-modal phishing detection framework for identifying malicious webpages with high accuracy and strong generalization capability.

REFERENCES

[1] A. Aleroud and L. Zhou, "Phishing environments, techniques, and countermeasures: A survey," *Computers & Security,* Vols. 160-196, p. 68, 2017.

[2] R. Verma and A. Das, "What's in a URL: Fast feature extraction and malicious URL detection," in *Proceedings of the 3rd ACM Workshop on Security and Artificial Intelligence*, 2015.

[3] A. Venturi, M. Colajanni, M. Ramilli and G. V. Santangelo, "Classification of Web Phishing Kits for Early Detection by Platform Providers," arXiv, 2022.

[4] S. Marchal, K. Saari, N. Singh and N. Asokan, "PhishStorm: Detecting phishing with streaming analytics," in *IEEE Conference on Communications and Network Security*, 458-466, 2014.

[5] N. Abdelhamid, A. Ayesh and F. Thabtah, "Phishing detection based on associative classification data mining," *Expert Systems with Applications,* vol. 41, no. 13, p. 41, 2014.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

[6] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning (ICML)*, 2019.

[7] P. He, X. Liu, J. Gao and W. Chen, "DeBERTa: Decoding-enhanced BERT with disentangled attention," in *International Conference on Learning Representations (ICLR)*, 2021.

[8] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[9] A. A. Orunsolu, A. S. Sodiya, A. T. Akinwale and B. I. Olajuwon, "An Anti-Phishing Kit Scheme for Secure Web Transactions," *International Journal of Electronics and Information Engineering,* vol. 6, no. 2, pp. 72-86, 2017.

[10] J. Feng, Y. Qiao, O. Ye and Y. Zhang, "Detecting phishing webpages via homology analysis of webpage structure," *PeerJ Computer Science,* vol. 8, p. e868, 2022.

[11] R. W. Purwanto, A. Pal, A. Blair and S. Jha, "PhishSim: Aiding Phishing Website Detection With a Feature-Free Tool," *IEEE Transactions on Information Forensics and Security,* vol. 17, pp. 1497-1512, 2022.

[12] U. e. al., "Phishing Website Detection Using Deep Learning Models," *IEEE Access,* 2023.

[13] 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.

[14] M. Sameen, K. Han and S. O. Hwang, "PhishHaven—An Efficient Real-Time AI Phishing URLs Detection System," *IEEE Access,* vol. 8, pp. 83425-83443, 2020.

[15] Z. e. al., "Phishing Website Detection Using Deep Learning Models," *IEEE Access,* 2023.

[16] F. Castaño, E. F. Fernández, R. Alaiz-Rodríguez and E. Alegre, "PhiKitA: Phishing Kit Attacks Dataset for Phishing Websites Identification," *IEEE Access,* vol. 11, pp. 40779-40789, 2023.

[17] C. Opara, B. Wei and Y. Chen, "HTMLPhish: Enabling Phishing Web Page Detection by Applying Deep Learning Techniques on HTML Analysis," *IEEE Access,* vol. 8, 2020.