INTERNATIONAL JOURNAL OF INNOVATIVE COMPUTING
ISSN 2180-4370
Journal Homepage : https://ijic.utm.my/

MultiPhishNet: A Multimodal Approach of QR Code Phishing Detection using Multi-Head Attention and Multilingual Embeddings

Omar Yasser Ibrahim Khalifa[1*] & Muhammad Zafran Muhammad Zaly Shah[2]
Faculty of Computing,
Universiti Teknologi Malaysia,
81310 UTM Johor Bahru, Johor, Malaysia
Email: yasser.ibrahim@graduate.utm.my[1]; m.zafran@utm.my[2]

Submitted: 17/2/2025. Revised edition: 20/4/2025. Accepted: 4/5/2025. Published online: 27/5/2025
DOI: https://doi.org/10.11113/ijic.v15n1.512

***Abstract*—Phishing attacks leveraging QR codes have become a significant threat due to their increasing use in contactless services. These attacks are challenging to detect since QR codes typically encode URLs leading to phishing websites designed to steal sensitive information. Existing detection methods often rely on blacklists or handcrafted features, which are inadequate for handling obfuscated URLs and multilingual content. This paper proposes MultiPhishNet, a multimodal phishing detection model that integrates advanced embedding techniques, Convolutional Neural Networks (CNNs), and multi-head attention mechanisms to automatically extract and learn key features from URLs and HTML content. The model leverages FastText embeddings for word-level representation, custom character embeddings for obfuscated URLs, and SBERT (Sentence-Bidirectional Encoder Representations from Transformers) embeddings for HTML content. To address class imbalance, ADASYN (Adaptive Synthetic Sampling) oversampling was applied, ensuring balanced training. The proposed method was evaluated on a moderately multilingual dataset, achieving an accuracy of 97.76% and an AUC of 0.9946. These results demonstrate that MultiPhishNet outperforms the baseline HTMLPhish model in phishing detection. Future research will focus on expanding the dataset to cover a broader range of languages and regional phishing tactics.**

***Keywords*—Phishing detection, QR codes, Multimodal deep learning, multilingual embeddings**

I. INTRODUCTION

Phishing attacks have evolved beyond traditional email scams, increasingly leveraging new technologies such as QR codes to deceive users. During the COVID-19 pandemic, the widespread adoption of QR codes for contactless interactions made them a prime target for malicious actors. Attackers can easily embed phishing URLs into QR codes, redirecting unsuspecting users to fraudulent websites designed to steal sensitive information or distribute malware. The inherent nature of QR codes, being visually unreadable by humans, exacerbates the difficulty of distinguishing malicious from legitimate codes [1]. Traditional detection methods, such as blacklisting, while effective for previously known phishing URLs, fail to identify newly generated or obfuscated phishing websites, underscoring the need for more advanced detection mechanisms [2].

Existing machine learning models for phishing detection often rely on handcrafted features that require extensive feature engineering, which is time-intensive and challenging to scale [3], or they focus solely on a single input modality, such as URLs or webpage content. However, phishing websites frequently employ obfuscation techniques, linguistic variations, and structural manipulations that cannot be effectively captured using conventional methods. Additionally, these models struggle with class imbalance, where legitimate samples significantly outnumber phishing samples, leading to biased models that perform poorly on minority classes [4].

To address these challenges, this paper presents MultiPhishNet, a multimodal phishing detection model that integrates both URLs and HTML content using advanced embedding techniques. The model leverages convolutional neural networks (CNNs) and multi-head attention mechanisms to automatically extract features from three types of embeddings: word-level FastText embeddings, character-level URL embeddings, and sentence-level SBERT embeddings for HTML content. The model was trained and evaluated on a moderate multilingual dataset, ensuring robust performance across different languages and phishing strategies.

Additionally, the application of ADASYN oversampling during training mitigates the issue of class imbalance by generating synthetic samples for underrepresented phishing cases.

The main contributions of this paper are summarized as follows:

- Effective detection of multilingual phishing websites by integrating multiple embedding types, including FastText for word-level representation, SBERT for sentence-level context, and character-level embeddings for obfuscated URLs.
- Improved handling of class imbalance through the application of ADASYN oversampling, leading to a more balanced training process and better detection of minority class phishing cases.
- Automated feature extraction using convolutional neural networks (CNNs) and multi-head attention mechanisms, reducing the need for handcrafted features and enhancing model generalization.

By combining multiple input modalities, MultiPhishNet offers a holistic approach to phishing detection, addressing critical gaps in existing models. Unlike traditional single-modality models that primarily analyze either URLs or HTML content, this multimodal approach captures structural nuances, linguistic variations, and obfuscation tactics comprehensively. The character-level embeddings enhance detection of obfuscated URLs by representing each character as an independent token, while FastText and SBERT embeddings capture word-level semantics and sentence-level context, respectively.

Moreover, the use of multi-head attention mechanisms allows the model to dynamically focus on key parts of the input data, improving its ability to identify phishing patterns in complex web structures. The combination of convolutional layers and attention mechanisms facilitates automated feature extraction, enhancing scalability while reducing the reliance on manual feature engineering.

II. RELATED WORK

Numerous research efforts have been dedicated to developing effective phishing detection techniques, particularly in the context of QR codes and multilingual phishing websites. This section reviews the evolution of phishing detection approaches, highlighting traditional blacklisting methods, machine learning solutions, and deep learning-based models with advanced architectures that address critical challenges in phishing detection.

A. Traditional Blacklisting Techniques

Traditional phishing detection predominantly relies on blacklisting, where known malicious URLs are stored in a centralized database. This method offers high precision for known threats but is inherently limited by its inability to detect newly generated or obfuscated phishing URLs. Moreover, blacklisting techniques struggle to cope with the dynamic nature of phishing attacks, as attackers frequently update their domains to evade detection [2]. Several popular blacklisting services, such as Google Safe Browsing and OpenPhish [5], [6], maintain large datasets of malicious URLs, but they require continuous updates and manual interventions to remain effective.

To overcome some of these limitations, hybrid approaches that combine blacklisting with other detection methods have been proposed. [2] introduced QsecR, a secure QR code scanning framework that integrates blacklisting with machine learning classifiers to enhance detection capabilities. Despite these improvements, blacklisting remains inadequate for addressing zero-day phishing attacks and multilingual phishing attempts.

B. Machine Learning-Based Solutions

Machine learning models for phishing detection leverage feature extraction techniques to classify URLs and webpage content. These models typically rely on handcrafted features, such as lexical properties, host-based information, and content-based attributes. [7] demonstrated the feasibility of using single-layer neural networks for phishing detection, achieving notable accuracy by manually engineering relevant features.

However, the reliance on feature engineering presents significant scalability issues. Manual feature extraction is time-intensive and requires domain expertise, making it challenging to adapt to evolving phishing strategies. More recent works, such as those by [8], explored the use of ensemble learning methods to enhance detection accuracy. While ensemble models improve performance, they still fall short in handling obfuscation and multilingual phishing attempts comprehensively.

C. Deep Learning-Based Approaches

Deep learning models have revolutionized phishing detection by automating feature extraction and enabling the processing of raw data, such as URLs and HTML content. HTMLPhish, proposed by [9], was one of the first models to apply convolutional neural networks (CNNs) for phishing detection by analysing raw HTML documents. This approach eliminated the need for manual feature engineering and achieved high detection accuracy.

Building on this foundation, [10] developed WebPhish, a multimodal deep learning framework that combines URL and HTML content embeddings using CNNs. By leveraging multiple data modalities, WebPhish demonstrated superior performance in detecting phishing websites across diverse datasets. This multimodal approach underscores the importance of integrating different types of inputs to enhance detection robustness.

Attention mechanisms have further improved phishing detection models by allowing dynamic focus on critical features within the input data. [11] incorporated multi-head attention mechanisms into CNN architectures, significantly enhancing the detection of obfuscated and complex phishing patterns. Additionally, they employed generative adversarial networks (GANs) to address class imbalance, further improving model generalization.

D. Multilingual Phishing Detection

Given the rise of global phishing campaigns targeting users across different languages, multilingual detection models have become crucial. Traditional approaches often fail to generalize well across languages due to language-specific feature dependencies. Recent research has explored the use of multilingual embeddings to address this challenge.

[12] demonstrated that transformer-based models, such as XLM-Roberta, can effectively capture semantic nuances in multilingual phishing content. By leveraging pre-trained multilingual embeddings, these models achieved high detection accuracy across diverse linguistic datasets.

Similarly, [13] highlighted the effectiveness of FastText and SBERT embeddings in multilingual phishing detection. FastText's subword-level representation proved particularly useful in capturing morphological variations, while SBERT provided contextual understanding at the sentence level. In this research, we adopt a combination of stsb-xlm-r-multilingual and FastText embeddings to enhance the detection of multilingual phishing websites, ensuring robust performance across various languages and phishing tactics.

E. Addressing Class Imbalance with ADASYN

One critical challenge in phishing detection is the inherent class imbalance in datasets, where legitimate samples vastly outnumber phishing samples. This imbalance can bias models towards predicting the majority class, reducing their ability to accurately detect phishing attempts. Oversampling techniques, such as Synthetic Minority Oversampling Technique (SMOTE) and Adaptive Synthetic Sampling (ADASYN), have been proposed to mitigate this issue.

While SMOTE generates synthetic samples by interpolating between existing minority class instances, ADASYN takes an adaptive approach by focusing more on difficult-to-classify instances [14]. Recent studies have shown that ADASYN outperforms SMOTE in phishing detection scenarios by generating more informative synthetic samples, particularly in regions with high-class overlap. In our research, we applied ADASYN to balance the dataset, ensuring that the model could effectively learn from underrepresented phishing cases without introducing noise or invalid samples.

By employing ADASYN, our approach enhances the robustness of the proposed phishing detection model, enabling better generalization across diverse datasets. The combination of ADASYN, multilingual embeddings, and multi-head attention mechanisms ensures a comprehensive solution for real-world phishing detection, addressing both data imbalance and linguistic diversity.

By adopting a holistic multimodal approach, our research aims to bridge the gap left by traditional and machine learning-based models, offering an accurate solution for phishing detection in multilingual environments.

III. METHODOLOGY

The methodology adopted in this study consists of three primary phases: Data Processing and Embedding, Class Imbalance Handling, and Model Development and Evaluation. Each phase is described in detail below, and the overall research framework is depicted in Fig. 1.

The proposed method adopts MultiPhishNet a multimodal approach to phishing detection by integrating advanced embedding techniques, Convolutional Neural Networks (CNNs), and a self-attention mechanism. This approach directly addresses key challenges in phishing detection, such as handling multilingual webpages and complex data formats, by focusing on the core content of a webpage URLs and HTML structures. Since phishing attempts can originate from various sources, including QR codes, the model prioritizes content analysis over the medium through which the webpage is accessed.

Given that a scanned QR code typically resolves to a URL, which then directs users to a webpage, the critical aspect for detection is the analysis of the resulting URL and HTML content. This ensures that phishing attempts can be detected effectively at the source, regardless of whether the webpage was accessed via a QR code scan or direct interaction. By focusing on these key components, the proposed approach offers robust phishing detection across different access methods, enhancing security in scenarios where QR codes are increasingly used for information sharing and transactions.

Fig. 1. Research Framework

A. Dataset Collection and Preparation

This research employs a publicly available dataset titled "Phishing Websites Dataset" from Mendeley data [15], The dataset comprises 80,000 instances, including 50,000 legitimate and 30,000 phishing websites. Each record contains essential elements such as the URL, corresponding HTML content, and metadata, including a record ID, creation date, and a binary label that indicates whether the entry is legitimate (0) or phishing (1). This dataset provides a comprehensive and moderately diverse representation of websites across various languages, URL structures, and domain categories making it suitable for multilingual phishing detection.

B. Data pre-processing

Data preprocessing was crucial to ensure the consistency and quality of the input data. This step involved cleaning, standardizing, and validating URLs and HTML content.

- Integrity Check: An integrity check ensured that each URL was accurately paired with its corresponding HTML content, eliminating missing or redundant entries.
- HTML Content Cleaning: Regular expressions were used to remove unnecessary tags and elements from the HTML content. Excess whitespace was normalized, retaining only essential textual information relevant for phishing detection.
- URL Standardization: URLs were converted to lowercase, and protocols (http://, https://) and subdomains (www.) were removed to focus on the core URL. Non-alphanumeric characters were replaced with spaces, except for Unicode ranges to handle multilingual URLs.
- Tokenization: URLs were tokenized at both word and character levels. Word-level tokenization captured high-level linguistic patterns, while character-level tokenization preserved fine-grained structural details, such as obfuscation techniques and irregular domain formats.
-

C. Multilingual Embedding

The cleaned and tokenized data were transformed into structured numerical representations using three types of embeddings:

- Word-Level Embeddings: FastText was used to generate 300-dimensional word embeddings. FastText's subword-level modeling capability enables the model to handle out-of-vocabulary tokens effectively, which is crucial for obfuscated or misspelled phishing URLs.
- Character-Level Embeddings: Each character token was represented as a 50-dimensional vector, capturing fine-grained structural patterns in URLs.
- Sentence-Level Embeddings: Sentence-BERT (SBERT) was employed to generate 768-dimensional embeddings for HTML content, providing contextual understanding across multiple languages. The SBERT variant used, stsb-xlm-r-multilingual, supports over 100 languages, ensuring the model's applicability in multilingual phishing detection.

Dynamic padding was applied to standardize the input dimensions. Word-level sequences were padded to 17 tokens, character-level sequences to 110 characters, and HTML embeddings retained their original 768-dimensional size, Fig. 2 illustrates the embedding workflow

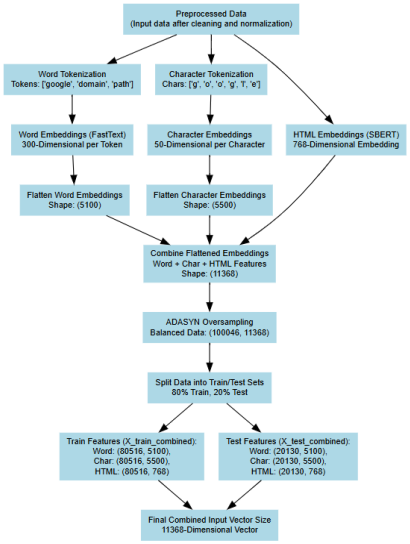

Fig. 2. Embedding Workflow

D. Class Imbalance Handling

Class imbalance is a significant issue in phishing detection, where legitimate websites vastly outnumber phishing ones. To address this, the Adaptive Synthetic Sampling (ADASYN) technique was used. ADASYN generates synthetic samples in difficult-to-classify regions of the feature space, ensuring a balanced dataset and improving the model's ability to generalize to new phishing patterns.

The dataset was oversampled from 79,987 to 100,046 samples, achieving parity between legitimate and phishing instances. After oversampling, the embeddings were reshaped to their original 3D forms to be compatible with the input layers of the model.

E. Model Development and Evaluation

The proposed model, named MultiPhishNet, integrates convolutional neural networks (CNNs) and a multi-head attention mechanism. CNNs are employed to extract spatial features from the input embeddings, enabling the model to capture critical patterns in URLs and HTML content. Multi-head attention is used to focus on the most relevant features across different branches, improving the model's ability to detect phishing patterns effectively.

The architecture consists of three parallel branches, each designed to process a different type of embedding essential for phishing detection. The HTML Embedding Branch handles 768-dimensional SBERT-generated HTML embeddings, where a dense layer initially reduces the dimensionality, followed by a convolutional layer and a multi-head attention mechanism to extract key contextual features. The URL Word Embedding Branch processes 300-dimensional word-level FastText embeddings using a 1D convolutional layer with 64 filters to extract features, followed by a multi-head attention mechanism. Lastly, the URL Character Embedding Branch processes 50-dimensional character-level embeddings, employing a 1D convolutional layer with 32 filters and a multi-

head attention mechanism to capture obfuscation patterns commonly found in URLs.

Global max pooling is applied after each branch to reduce dimensionality while preserving important features. The outputs from these branches are concatenated and passed through dense layers for final classification.

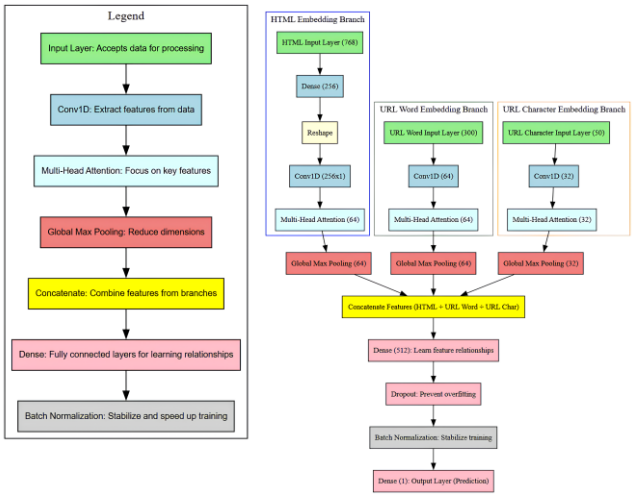

Fig. 3. MultiPhishNet Architecture

Fig. 3 illustrates the MultiPhishNet architecture, highlighting the input layers for different embeddings, convolutional layers, multi-head attention mechanisms, and the final dense layers for classification.

The model was trained using the Adam optimizer with a learning rate of 0.001 and evaluated using metrics such as accuracy, precision, recall, F1-score, and AUC. Early stopping was applied to prevent overfitting, and 80% of the data was allocated for training, with the remaining 20% used for testing.

By combining advanced embedding techniques, ADASYN oversampling, CNNs, and attention mechanisms, the proposed approach achieved robust performance in detecting phishing websites across multiple languages and varying obfuscation tactics.

IV. MULTIPHISHNET IMPLEMENTATION

The implementation of MultiPhishNet involves the practical realization of the proposed methodology into a robust phishing detection system. This section highlights key implementation details, including embedding generation, handling class imbalance, model construction, and training.

A. Embedding Generation

The core of MultiPhishNet's implementation is its multimodal embedding strategy, which processes URLs and HTML content through different representations:

- HTML Embedding: HTML content was preprocessed to remove unnecessary tags and elements, resulting in clean textual data. The stsb-xlm-r-multilingual model, a variant of SBERT supporting over 100 languages, was used to generate 768-dimensional embeddings for each HTML instance.
- Word-Level Embedding: FastText pre-trained word vectors were employed to capture semantic information at the word level. Tokenized URLs were padded to a sequence length of 17 tokens, with each token represented as a 300-dimensional vector.
- Character-Level Embedding: To capture structural anomalies and obfuscation in URLs, character-level embeddings were generated. Each character sequence was padded to a fixed length of 110 characters, with each character encoded as a 50-dimensional vector.

B. Class Balancing with ADASYN

Phishing datasets are typically imbalanced, with legitimate websites far outnumbering phishing ones. This imbalance poses a significant challenge, as models trained on such datasets tend to become biased towards the majority class, leading to poor recall for phishing cases. Initial experiments with Synthetic Minority Oversampling Technique (SMOTE) were conducted to address this issue. However, SMOTE generated invalid synthetic samples in some regions of the feature space, particularly for complex HTML embeddings, which adversely affected the model's performance.

To overcome this limitation, the Adaptive Synthetic Sampling (ADASYN) technique was employed. Unlike SMOTE, ADASYN adaptively focuses on generating synthetic samples in regions where classification is more difficult, ensuring a better-defined decision boundary and improving the model's ability to generalize to unseen phishing patterns.

The combined feature matrix, consisting of 11,368 dimensions (word, character, and HTML embeddings), was used as input for ADASYN. After applying the technique, the dataset size increased from 79,987 to 100,046 samples, achieving parity between legitimate and phishing instances. The balanced dataset was then split into 80% for training and 20% for testing.

C. Model Construction

The MultiPhishNet architecture comprises three branches, each designed to process a specific type of embedding:

- HTML Embedding Branch: Processes 768-dimensional SBERT embeddings using a dense layer followed by a convolutional layer and a multi-head attention mechanism.
- Word-Level Embedding Branch: Processes 300-dimensional FastText word embeddings using a 1D convolutional layer with 64 filters, followed by a multi-head attention mechanism.
- Character-Level Embedding Branch: Processes 50-dimensional character embeddings using a 1D convolutional layer with 32 filters, followed by a multi-head attention mechanism.

Global max pooling was applied in each branch to reduce dimensionality. The outputs from all branches were concatenated and passed through dense layers, with dropout and batch normalization applied to enhance generalization. The final layer used a sigmoid activation function to produce binary classifications.

Table I highlights the most important layers of the MultiPhishNet model, including their output shapes and parameter counts.

TABLE I. SUMMARY OF KEY LAYERS IN THE MULTIPHISHNET MODEL

Layers	**Output Shape**	**Param #**	Activation Function
html_input	(None, 768)	0	-
url_word_input	(None, 17, 300)	0	-
url_char_input	(None, 110, 50)	0	-
dense1	(None, 256)	196,864	ReLU
conv1d	(None, 17, 64)	57,664	-
multi_head_attention_layer	(None, 17, 64)	132,672	-
global_max_pooling1d_2	(None, 64)	0	-
dense2	(None, 512)	131,584	ReLU
dropout	(None, 512)	0	-
batch_normalization	(None, 512)	2,048	-
dense3	(None, 1)	513	Sigmoid
Total Trainable Parameters		625,345	-

This table summarizes critical layers, focusing on input processing, feature extraction, attention mechanisms, and the final classification layer. The total number of parameters in the model is 626,369, with 625,345 trainable parameters and 1,024 non-trainable parameters.

D. QR Code Scanner Integration

QR codes are increasingly being used as a phishing vector, directing users to potentially harmful webpages. Since QR codes essentially encode URLs, the phishing detection model was extended to handle QR code-based phishing attempts by scanning QR codes, retrieving the linked webpage, and analysing its content. This approach ensures that the detection system can generalize across different entry points, whether users manually type a URL or access it through a QR code.

The QR code scanning pipeline begins by reading QR code images using the pyzbar library, which extracts the corresponding URLs. For each URL, the HTML content of the linked webpage is fetched using the requests library, with retry logic to handle potential network issues. The retrieved HTML content is then pre-processed using a Sentence-BERT (SBERT) model to generate semantic embeddings. Simultaneously, the URLs are tokenized at both word (FastText) and character levels, ensuring compatibility with the trained phishing detection model. Finally, the processed inputs are fed into the phishing detection model, which classifies the webpage as either Phishing or Legitimate based on the prediction score. This QR code extension highlights the flexibility of the phishing detection solution in handling various real-world scenarios where phishing attacks can originate from different sources.

E. Model Training and Hyperparameter Configuration

The model was trained using the Adam optimizer with a learning rate of 0.001. The cross-entropy loss function was employed to minimize classification error. Training was conducted over 80 epochs with a batch size of 16, and early stopping with a patience of 10 epochs was used to prevent overfitting. Additionally, a cosine decay restart scheduler was applied to reset the learning rate periodically, aiding in better convergence.

The final model was selected based on the highest validation accuracy, ensuring optimal performance. Key hyperparameters are summarized in Table II.

By employing diverse embeddings, advanced feature extraction techniques, and effective oversampling ADASYN, MultiPhishNet achieved a robust solution for phishing detection.

TABLE II. KEY HYPERPARAMETERS FOR MULTIPHISHNET MODEL

Hyperparameter	**Value**
Batch size	16
Initial learning rate	0.001
Maximum epochs	80
Early stopping patience	10
Dropout rate	0.5
L2 regularization	0.01

V. RESULTS

This section presents a comprehensive analysis of the experimental results obtained from evaluating MultiPhishNet. The results include a comparison with the baseline HTMLPhish model, an examination of the effect of ADASYN oversampling on model performance, and a detailed analysis of individual model branches (HTML-only, URL-only), along with the Residual Attention Mechanism model, and Comparative Analysis with Other Phishing Detection Models.

A. MultiPhishNet Model Evaluation

The primary objectives were improving phishing detection across diverse languages, handling class imbalance effectively, and leveraging automated feature extraction through an advanced multimodal architecture. Dataset containing URLs and HTML content from websites in various languages were used to ensure multilingual compatibility.

The MultiPhishNet model's performance was evaluated using key metrics, including accuracy, AUC, precision, recall, and F1-score. Fig. 4 shows the overall results, where the model achieved an accuracy of 97.76%, AUC of 0.9946, precision of 97.53%, recall of 97.98%, and F1-score of 97.75%, indicating high classification effectiveness across phishing and legitimate samples.

Accuracy reflects the proportion of correctly classified websites, providing an overall measure of the model's correctness. AUC indicates the model's ability to differentiate between phishing and legitimate websites, with a near-perfect score of 0.9946 suggesting strong discrimination capability. Precision of 97.53% ensures that most websites predicted as phishing were indeed phishing, minimizing false positives. Recall of 97.98% highlights the model's capability to correctly identify most phishing websites, ensuring a high detection rate. The F1-score of 97.75% balances precision and recall, confirming the model's robustness in phishing detection.

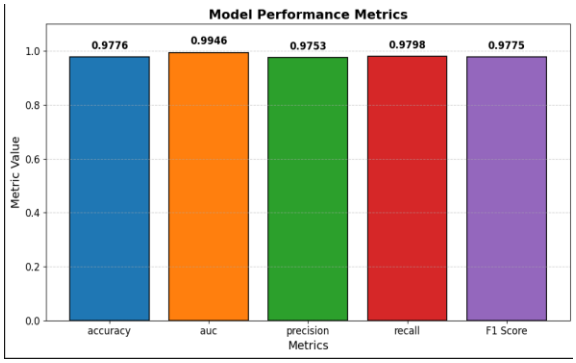

Fig. 4. Overall Metrics of the MultiPhishNet Model

The training-validation accuracy and loss plots Fig. 5 illustrate the model's learning behaviour over 30 epochs. Where The accuracy plot shows a consistent improvement in both training and validation accuracy, with validation accuracy stabilizing close to the training accuracy after the 10th epoch, reaching nearly 98%. This trend indicates that the model generalizes well to unseen data without significant overfitting. The loss plot demonstrates a steady decrease in both training and validation loss, with the validation loss stabilizing around 0.1 after the 10th epoch. This indicates effective convergence of the model during training.

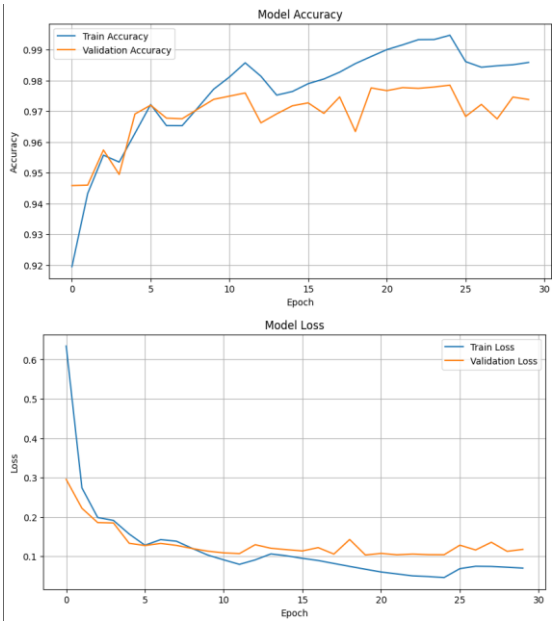

Fig. 5. Training-Validation Accuracy and Loss Plots

B. Comparison with Baseline Model

The baseline model a variant of HTMLPhish [9] was tested on the same dataset used for evaluating the proposed MultiPhishNet phishing detection model. The HTMLPhish model primarily focuses on processing HTML content through convolutional layers without incorporating URL information. In contrast, the proposed MultiPhishNet model combines both URL and HTML content using advanced embedding techniques and multi-head attention mechanisms, which allow the model to capture diverse linguistic patterns and structural nuances more effectively. A comparison of the result two models is presented in table 3, focusing on key performance metrics, including accuracy, AUC, precision, recall, and F1-score.

TABLE III. SUMMARY OF KEY LAYERS IN THE MULTIPHISHNET MODEL

Model	Accuracy	AUC	Precision	Recall	F1-Score
[9]	92.08%	97.08%	90.68%	93.42%	92.03%
MultiPhishNet	97.76%	99.46%	97.53%	97.98%	97.75%

F

C. Effect of ADASYN Oversampling on Model Performance

To address the inherent class imbalance in phishing datasets, ADASYN oversampling was employed. Table IV below presents the comparative results of the model with and without ADASYN.

TABLE IV. PERFORMANCE METRICS COMPARISON

Metric	Without ADASYN	With ADASYN
Accuracy	96.78%	97.76%
AUC	99.06%	99.46%
Precision	96.18%	97.53%
Recall	95.13%	97.98%
F1-Score	95.65%	97.75%

The application of ADASYN led to notable improvements across most performance metrics. Specifically, recall increased by 2.85%, reflecting the model's enhanced ability to identify phishing cases more effectively. Additionally, the F1-score rose by 2.1%, indicating a better balance between precision and recall. While accuracy and AUC exhibited smaller gains, the improvements in recall and F1-score demonstrate the effectiveness of ADASYN in mitigating class imbalance and enhancing overall detection performance.

D. Comparative Analysis of Model Variations

This section presents the comparative performance of four key model variations: the HTML-Only Model, the URL-Only Model, the Residual Attention Mechanism Model, and the final proposed model, MultiPhishNet. Each model was tested under identical conditions using the same dataset to ensure a fair

evaluation. The comparison was conducted using five key metrics: accuracy, AUC, precision, recall, and F1-score. Fig. 6 and Table V illustrates the performance across these metrics for all model variations.

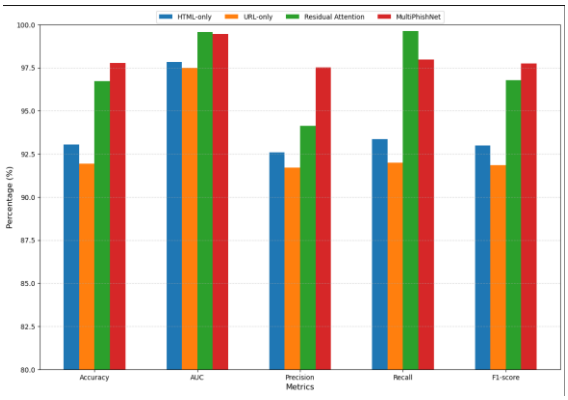

Fig. 6. Performance Comparison of Model Variations

TABLE V. PERFORMANCE METRICS FOR DIFFERENT MODEL VARIATIONS

Model	Accuracy (%)	AUC (%)	Precision (%)	Recall (%)	F1-score (%)
HTML-Only Model	93.03	97.83	92.59	93.37	92.98
URL-Only Model	91.93	97.49	91.69	92.00	91.84
Residual Attention Model	96.73	99.57	94.12	99.63	96.79
MultiPhishNet	97.76	99.46	97.53	97.98	97.75

The MultiPhishNet model achieved the highest performance across all metrics, demonstrating significant improvements compared to the other variations. The Residual Attention Mechanism Model also showed strong results, particularly in recall, due to its ability to capture complex patterns by integrating multi-head attention and residual connections. Meanwhile, the HTML-Only and URL-Only models, though effective, exhibited slightly lower scores, indicating the importance of combining both URL and HTML information for phishing detection.

VI. CONCLUSION

This research focused on developing a robust multimodal phishing detection model, MultiPhishNet, which integrates convolutional neural networks (CNNs) and multi-head attention mechanisms to analyse both URLs and HTML content. By adopting advanced multilingual embeddings, such as FastText and SBERT, the model successfully addressed key challenges associated with phishing websites targeting users in different languages. The outcomes demonstrated that MultiPhishNet consistently outperformed other model variations and the baseline HTMLPhish model, achieving an overall accuracy of 97.76%. This improvement was reflected across other key metrics, including precision, recall, F1-score, and AUC.

The experiments also highlighted the importance of addressing class imbalance using the ADASYN oversampling technique, which enhanced the model's ability to detect underrepresented phishing samples. Comparisons between models trained with and without ADASYN confirmed the effectiveness of this technique in boosting overall performance. These outcomes validate the hypothesis that combining multiple input modalities with automated feature extraction methods significantly improves phishing detection.

Despite the promising results achieved, this study has certain limitations. The dataset used for training and evaluation was moderately multilingual, encompassing a medium number of languages. While it demonstrates the feasibility of phishing detection in multilingual environments, its coverage of diverse linguistic patterns and region-specific phishing tactics remains limited.

Future research can enhance the proposed approach by expanding the dataset to include a broader range of languages and more diverse phishing tactics. This expansion would improve the model's generalization across various linguistic contexts and enhance its ability to detect region-specific phishing attacks.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to the anonymous reviewers for their valuable comments and constructive suggestions, which have helped improve the quality and clarity of this paper.

CONFLICTS OF INTEREST

The author(s) declare(s) that there is no conflict of interest regarding the publication of this paper.

REFERENCES

[1] Starikova, A. (2022). Phishing QR-code Attack on QQ Users. https://me-en.kaspersky.com/blog/phishing-qr-code-attack-on-qq-users/19794/

[2] Rafsanjani, A. S., Kamaruddin, N. B., Rusli, H. M., Dabbagh, M. (2023). QsecR: Secure QR Code Scanner According to a Novel Malicious URL Detection Framework. *IEEE Access*, *11*, 92523–39. Doi:10.1109/ACCESS.2023.3291811.

[3] Gong, Y., Liu, G., Xue, Y., Li, R., Meng, L. (2023). A Survey on Dataset Quality in Machine Learning. *Inf Softw Technol.*, *162*, 107268. Doi: 10.1016/j.infsof.2023.107268.

[4] Brandqvist, J., John, L. N. (2023). Phishing Detection Challenges for Private and organizational Users: A Comparative Study. *DIVA Portal.* https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1778402.

[5] Google Safe Browsing. https://safebrowsing.google.com/

[6] OpenPhish – Phishing Intelligence. https://openphish.com/

[7] Nguyen, L. A. T., To, B. L., Nguyen, H. K., Nguyen, M. H. (2014). An Efficient Approach for Phishing Detection using Single-layer Neural Network. *Proc Int Conf Adv Technol Commun (ATC).* Hanoi, Vietnam: IEEE; 435–40. Doi:10.1109/ATC.2014.

[8] Pawar, A., Fatnani, C., Sonavane, R., Waghmare, R., Saoji, S. (2022). Secure QR Code Scanner to Detect Malicious URL using Machine Learning. *2022 2nd Asian Conference on*

Innovation in Technology (ASIANCON). IEEE; Doi:10.1109/asiancon55314.2022.9908759.

[9] Opara, C., Wei, B., Chen, Y. (2020). HTMLPhish: Enabling Phishing Web Page Detection by Applying Deep Learning Techniques on HTML Analysis. IEEE.

[10] Opara, C. Chen, Y., Wei, B. (2024). Look Before You Leap: Detecting Phishing Web Pages by Exploiting Raw URL and HTML Characteristics. *Expert Syst Appl.*, *236*, 121183. Doi:10.1016/j.eswa.2023.121183.

[11] Said, Y., Alsheikhy, A. A., Lahza, H., Shawly, T. (2024). Detecting Phishing Websites through Improving Convolutional Neural Networks with Self-attention Mechanism. *Ain Shams Eng J., 15*(4), 102643. Doi:10.1016/j.asej.2024.102643.

[12] Staples, D., Hakak, S., Cook, P. (2023). A Comparison of Machine Learning Algorithms for Multilingual Phishing Detection. IEEE. Doi:10.1109/pst58708.2023.10320177.

[13] Miguel, M. G. A., Faris, H. (2024). Spam Reviews Detection Models in Multilingual Contexts Applying Sentiment Analysis, Metaheuristics, and Advanced Word Embedding. Universidad de Granada; https://digibug.ugr.es/handle/10481/91051.

[14] Zakariah, M., AlQahtani, S. A., Al-Rakhami, M. S. (2023). Machine Learning-based Adaptive Synthetic Sampling Technique for Intrusion Detection. *Appl Sci*., *13*(11), 6504. Doi:10.3390/app13116504.

[15] Ariyadasa, S., Fernando, S., Fernando, S. (2021). Phishing Websites Dataset. *Mendeley Data.* V1. Doi:10.17632/n96ncsr5g4.1.

[16] Yazhmozhi, M., Janet, B., Reddy. (2020). US. Anti-phishing System using LSTM and CNN. *2020 IEEE Int Conf on Innovative Research in Engineering and Technology (INOCON).* IEEE. 1–5. Doi:10.1109/INOCON50539.2020.9298298

[17] Adebowale, M., Lwin, K., Hossain, A. (2020). Intelligent Phishing Detection Scheme Algorithms using Deep Learning. *J Enterp Inf Manag*. Doi:10.1108/JEIM-01-2020-0036.

[18] Do, N. Q., Selamat, A., Krejcar, O., Yokoi, T., Fujita, H. (2021). Phishing Webpage Classification via Deep Learning-based Algorithms: An Empirical Study. *Appl Sci., 11*(19), 9210. Doi:10.3390/app11199210.