Comparative Evaluation of Phishing Detection Models: Structured vs. Raw Multimodal Datasets

Ishraq Kamal Adib
Dept. of Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
ishraq.kamal.adib@g.bracu.ac.bd

Ishan Khan
Dept. of Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
ishan.khan@g.bracu.ac.bd

Nehal Mahfuz
Dept. of Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
nehal.mahfuz@g.bracu.ac.bd

***Abstract*—Phishing attacks continue to evolve, requiring sophisticated detection mechanisms that can adapt to diverse data sources and processing constraints. This paper presents a comparative evaluation of machine learning models for phishing detection across two contrasting datasets: a structured, preprocessed benchmark dataset and a raw multimodal dataset containing HTML, CSS, and image data. We evaluate classical ML, ensemble, and deep learning approaches on these datasets to understand how data quality, preprocessing effort, and modality diversity influence detection performance. Our results show that XGBoost achieves 96.41% accuracy on the structured dataset, while LightGBM reaches 93.46% on the raw multimodal dataset. Feature importance analysis reveals external service-based features as key discriminators in structured data, while multimodal integration provides complementary signals despite increased complexity. These findings offer practical insights for deploying phishing detection systems under varying resource constraints.**

***Index Terms*—phishing detection, multimodal analysis, logistic regression, biLSTM, XGBoost, randomforest, lightGBM, VGG16**

I. Introduction

Phishing is a growing threat whereby attackers trick users into revealing sensitive information by impersonating legitimate websites. Despite various detection measures, sophisticated phishing schemes continue to bypass defenses, causing significant financial and reputational damages globally. Recent advancements in machine learning and multimodal feature integration offer promising detection capabilities, combining URL lexical patterns, webpage content analysis, and external verification services to better discern phishing attacks.

This paper focuses on reviewing these multimodal approaches and evaluating their performance on two curated datasets with contrasting characteristics:

1) A structured, preprocessed benchmark dataset requiring minimal cleaning.
2) A large-scale, raw multimodal dataset containing HTML, CSS, and image data.

By comparing these scenarios, we aim to understand how dataset quality, preprocessing effort, and modality diversity influence phishing detection performance and model selection.

The main contributions of this paper are:

- Comparative evaluation of classical ML, ensemble, and deep learning models on two phishing datasets with differing preprocessing levels and modalities.
- Feature importance analysis to identify key discriminative signals across URL, HTML, and external service features.
- Discussion of trade-offs between dataset complexity, preprocessing effort, and achievable performance. Practical recommendations for deploying phishing detection systems under varying resource constraints.

II. Literature Review

Phishing website detection has seen significant advancements through diverse methodologies. Shafin (2025) [1] introduced the SLA-FS (SHAP and LIME–Aggregated Feature Selection) framework, designed to enhance detection by combining SHapley Additive exPlanations (SHAP) for global feature importance with Local Interpretable Model-Agnostic Explanations (LIME) for local, instance-specific validation. Applied to the Web Page Phishing Detection dataset (11,430 balanced samples, 87 features across URL, HTML, and external services), SLA-FS retained features consistently important across both perspectives. After dimensionality reduction to 56 features, Random Forest (RF), XGBoost (XGB), and KNN all improved in accuracy, with RF reaching 97.41%. However, the framework incurred high computational cost and showed marginal gains for KNN. Future extensions aim to integrate SLA-FS into deep learning and ensemble models, targeting larger, more diverse datasets.

Building on interpretability-driven feature selection, Sánchez-Paniagua et al. (2022) [2] developed the Phishing Index Login Websites Dataset (PILWD-134K) and validated SLA-FS on this large-scale resource (134,000 verified samples, 54 handcrafted features). Evaluations across RF, XGB, LightGBM, and KNN showed LightGBM achieving the highest accuracy (97.95%). Dimensionality reduction enhanced efficiency, though KNN still benefited little. The authors highlighted the computational expense of SLA-FS

on massive datasets and dependency-related false positives. Future work suggests hybrid ensemble integration and cross-institutional benchmarking.

The absence of standardized benchmarks was tackled by Hannousse and Yahiouche (2020) [3], who proposed a reproducible dataset construction framework guided by six principles, including URL diversity, DOM preservation, and temporal traceability. Their balanced dataset (11,430 samples, 87 features) enabled comparisons across Decision Tree, RF, Logistic Regression, Naïve Bayes, and SVM with various feature selection strategies. RF with Chi-square feature selection achieved the highest accuracy (96.83%), with external service features proving most informative. Limitations involved heavy reliance on external services, costly feature extraction, and limited dataset size. They recommend improving extraction efficiency, supporting real-time updates, and validating on multiple benchmarks.

Unlike feature-engineering approaches, Alam et al. (2020) [4] introduced WebPhish, a deep learning model that directly processes raw URLs and HTML through a dual-input CNN-based architecture. By embedding and fusing URL and HTML characters, WebPhish achieved 98.1% accuracy, surpassing traditional ML models and single-modality baselines. Despite its effectiveness, the method depended on full HTML access and required significant computational resources. The authors suggested incorporating visual and temporal features, exploring transformers, and optimizing for lightweight deployment.

To exploit complementary modalities, Zhou et al. (2022) [5] combined lexical URL features with HTML textual and structural attributes, forming composite vectors for XGBoost classification. Trained on PhishTank (phishing) and Alexa (legitimate) data, the model achieved 99.31% accuracy, outperforming URL-only or HTML-only baselines. While highly effective, the approach remained vulnerable to zero-day attacks and required HTML access. Suggested improvements included deep learning replacements for handcrafted features and integration of visual or reputation-based attributes.

Focusing exclusively on URLs, Rahman et al. (2024) [6] proposed the Optimal Feature Vectorization Algorithm (OFVA), extracting 41 intra-URL features (10 novel) from a dataset of 247,950 URLs, later reduced to 34 features via RF importance ranking. Across 15 ML classifiers, RF achieved the best results (97.52% accuracy, balanced recall 0.97–0.98). However, the absence of HTML or behavioral features limited robustness against sophisticated phishing. Future directions include feature integration, deep learning adoption, and testing on adversarial scenarios.

Parallelization was the key theme in Nagy et al. (2023) [7], who compared sequential and parallel implementations of ML and DL models using Singh and Kumar's dataset (66,000 samples with lexical, content-based, and network-level features). Naïve Bayes achieved 96.01% accuracy, while RF, CNN, and LSTM models reached perfect recall (100%). Parallelization reduced training time by up to 71.5% without performance loss, though improvements varied by hardware. Future research should focus on GPU-optimized scaling for real-time applications.

On smaller datasets, Shaik et al. (2022) [8] assessed Decision Tree, RF, LightGBM, Logistic Regression, and SVM using 3,000 URLs from PhishTank and UNB. LightGBM attained the highest testing accuracy (86.0%), narrowly outperforming RF and Decision Tree. Performance was constrained by limited data volume, missing WHOIS records, and feature diversity. Suggested enhancements include expanding datasets, incorporating HTML features, and exploring DL approaches.

Finally, Kumar and Singh (2022) [9] explored deep learning architectures (LSTM, Bi-LSTM, GRU) for phishing URL detection on 450,176 samples. Bi-LSTM delivered the best accuracy (99.0%), followed by GRU (97.5%) and LSTM (96.9%), demonstrating the strength of sequential modeling. Despite this, dataset imbalance and the omission of domain or behavioral features constrained generalizability. Future efforts may incorporate multimodal data, low-latency optimization, and transformer-based architectures.

A. Synthesis

In summary, prior research demonstrates that both classical and deep learning models can achieve high accuracy on curated phishing datasets, especially when leveraging multimodal features and advanced feature selection methods. However, most studies focus on either highly preprocessed datasets or large-scale raw datasets in isolation. Few works directly compare model performance across datasets with differing preprocessing levels and modalities. This paper addresses this gap by evaluating multiple models on both a structured benchmark dataset and a raw multimodal dataset, highlighting the implications for real-world deployment.

III. Methodology

A. Overview of Approach

In this study, we adopt a structured experimental workflow to evaluate phishing detection performance across two datasets with contrasting characteristics. The process begins with dataset acquisition and exploratory analysis, followed by dataset-specific preprocessing and feature engineering. For the structured benchmark dataset (Dataset 1), minimal preprocessing was required due to its curated nature, while the raw multimodal dataset (Dataset 2) demanded extensive integration of HTML, CSS, and visual features.

Feature sets were then used to train a range of machine learning models, including classical algorithms, ensemble methods, and deep learning architectures. Model performance was assessed using accuracy, precision, recall and F1-score, with additional interpretability provided through feature importance analysis. The workflow is summarised in Fig. 1, and detailed methodologies for each dataset are presented in the following subsections.

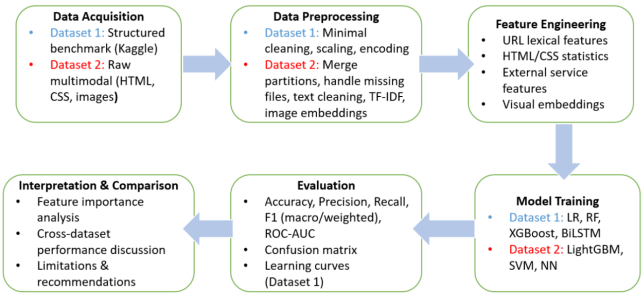

Fig. 1. Overview of datasets and experimental workflow.

B. Dataset Description

Initially, we began our experiments with a dataset sourced from the Kaggle Web Page Phishing Detection benchmark, originally constructed by Hannousse and Yahiouche (2020) [10] . This dataset contains 11,430 unique samples, evenly split between phishing (5,715) and legitimate (5,715) webpages. Each instance is represented by 89 pre-engineered features spanning three categories:

- URL-based features (e.g., number of question marks, number of dots, ratio of digits in host).
- HTML/content-based features (e.g., number of hyperlinks, presence of empty title, phishing hints in text).
- External service-based features (e.g., Google index status, PageRank, web traffic rank).

The dataset was already clean, balanced, and well-structured, with no missing values and consistent feature formatting. Only minimal preprocessing was required — removal of non-predictive identifiers, scaling of numerical features, and one-hot encoding of rare categorical variables. While this dataset allowed us to quickly achieve high accuracy with multiple models, its preprocessed nature limited opportunities to explore complex data cleaning, feature extraction, and integration steps.

To evaluate our approach under more challenging, real-world conditions, we subsequently selected a large-scale, raw multimodal dataset containing HTML, CSS, and screenshot image files alongside metadata. The original dataset size exceeded 35 GB, making it impractical to process in full within our time and resource constraints. We therefore extracted a balanced subset of 1,000 samples (500 phishing, 500 legitimate) for experimentation. Unlike the first dataset, this collection required substantial preprocessing: merging multiple data partitions, handling missing files, extracting basic HTML/CSS statistics, generating TF-IDF features from text content, and deriving visual embeddings from screenshots using VGG16. The resulting feature set integrated multiple modalities into a single tabular representation, enabling evaluation of models on heterogeneous, high-dimensional data.

C. Dataset Preprocessing

The preprocessing stage was tailored to the characteristics of each dataset.

Structured benchmark dataset: Given its curated nature, this dataset required only minimal preprocessing. Non-predictive identifiers (e.g., URL strings) were removed to prevent data leakage. All numerical features were scaled to a uniform range using standard scaling, and rare categorical variables were one-hot encoded. No missing values were present, and feature names were already standardized.

Raw multimodal dataset: In contrast, the raw multimodal dataset required extensive preprocessing to integrate heterogeneous data sources. Multiple data partitions containing HTML, CSS, image, and metadata files were merged into a unified structure. Missing or inaccessible files were identified and excluded. HTML and CSS content underwent basic cleaning (removal of extraneous whitespace, non-UTF-8 characters) before feature extraction. Textual features were generated using TF-IDF vectorization, while visual features were obtained by passing screenshots through a pre-trained VGG16 network and extracting the penultimate layer embeddings. All feature groups were concatenated into a single tabular representation, with missing values imputed where necessary.

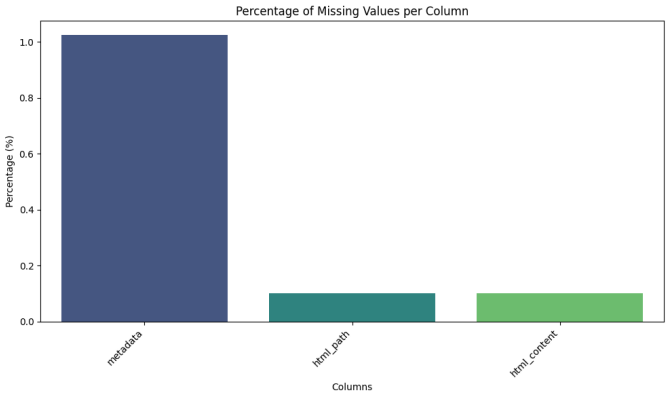

Fig. 2. Data preprocessing pipeline with missing value handling.

This preprocessing pipeline ensured that both datasets were in a consistent, model-ready format, while preserving the multimodal nature of the second dataset.

D. Feature Engineering and Selection

The feature engineering process differed substantially between the two datasets due to their inherent structure and modality diversity.

Structured benchmark dataset: As the dataset already contained 89 well-defined features spanning URL lexical, HTML/content, and external service categories, no additional feature extraction was required. However, to assess the relative contribution of each feature, we computed feature importance scores using the trained XGBoost model. This analysis revealed that the most influential predictors included `google_index`, `page_rank`, and `web_traffic` (external service features), as well as lexical indicators such as `nb_qm`, `nb_www`, and `ratio_digits_host`. Content-based features like `nb_hyperlinks` and

`empty_title` also ranked highly, confirming the value of multimodal signals even in a preprocessed dataset.

Raw multimodal dataset: Feature engineering was essential to transform heterogeneous raw data into a unified representation. HTML and CSS files were parsed to extract structural statistics (e.g., tag counts, attribute frequencies), while textual content was vectorized using TF-IDF to capture discriminative terms. Screenshot images were processed through a pre-trained VGG16 network to obtain 4,096-dimensional visual embeddings. These feature groups — HTML statistics, CSS statistics, TF-IDF vectors, and visual embeddings — were concatenated, and missing values were imputed. No explicit dimensionality reduction was applied due to the relatively small subset size, but the high dimensionality of the visual and TF-IDF features was noted as a potential challenge for scalability.

In both datasets, the final feature sets preserved modality diversity, enabling the evaluation of models capable of handling heterogeneous inputs.

E. Model Selection and Training

The choice of models for each dataset was guided by their size, feature composition, and modality diversity.

Structured benchmark dataset: We evaluated a mix of classical machine learning algorithms, ensemble methods, and a deep learning baseline. Specifically, Logistic Regression (LR), Random Forest (RF), and XGBoost (XGB) were selected for their proven performance on tabular phishing datasets, while a Bidirectional Long Short-Term Memory (BiLSTM) network was included to assess the potential of sequence-based deep learning on pre-engineered features.

Raw multimodal dataset: Given the high dimensionality and heterogeneous nature of the features, we selected LightGBM, XGBoost, and a feed-forward Neural Network (NN). LightGBM was chosen for its efficiency and ability to handle sparse, high-dimensional data; XGBoost for its robustness and proven performance on tabular datasets with complex feature interactions; and NN as a flexible baseline for capturing nonlinear relationships across modalities.

For both datasets, an 80/20 train-test split was used, ensuring class balance in both sets. Model performance was evaluated on the held-out test set using accuracy, precision, recall, F1-score (macro and weighted). Confusion matrices were generated to analyse misclassification patterns, and learning curves were plotted (for the structured dataset) to assess generalisation behaviour.

F. Evaluation Metrics

Model performance was assessed using a combination of threshold-based and ranking-based metrics to provide a comprehensive evaluation of classification effectiveness.

Accuracy measures the proportion of correctly classified instances over the total number of instances. While useful for balanced datasets, it can be misleading in imbalanced scenarios.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Precision quantifies the proportion of predicted phishing sites that are truly phishing, reflecting the model's ability to avoid false positives. This is critical in operational contexts where incorrectly flagging legitimate sites may disrupt user trust.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

Recall (or sensitivity) measures the proportion of actual phishing sites correctly identified, capturing the model's ability to minimise false negatives. High recall is particularly important in phishing detection, as missed phishing attempts can lead to severe security breaches.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

F1-score is the harmonic mean of precision and recall, providing a balanced measure when both false positives and false negatives are important. Macro-averaged F1-score treats all classes equally, while weighted F1-score accounts for class imbalance.

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

For interpretability, confusion matrices were generated to visualise the distribution of true positives, false positives, true negatives, and false negatives. Learning curves were plotted (for the structured dataset) to assess generalisation behaviour as a function of training set size.

G. Implementation Details

All experiments were executed in Google Colab using a GPU runtime (Python 3.x) with scikit-learn, XGBoost, LightGBM, TensorFlow/Keras, Pandas, and NumPy. Feature extraction used scikit-learn (TF-IDF) and Keras Applications (VGG16) for visual embeddings. We fixed random seeds and saved intermediate outputs (feature importance, confusion matrices, learning curves) for reproducibility.

IV. Results and Discussion

A. Overview of Experimental Results

The experiments were conducted on two datasets with contrasting characteristics: a clean, structured benchmark dataset and a raw, multimodal dataset requiring extensive preprocessing. Across both datasets, ensemble methods consistently outperformed other approaches, though the absolute performance levels differed markedly due to dataset quality, feature composition, and preprocessing complexity.

B. Structured Benchmark Dataset Results

Table I summarises the performance of all evaluated models on the structured benchmark dataset. XGBoost achieved the highest overall performance, with an accuracy of 97.21%, macro-averaged F1-score of 0.972, and balanced precision and recall. Random Forest followed closely, while Logistic Regression and BiLSTM delivered slightly lower but still competitive results.

TABLE I
PERFORMANCE OF MODELS ON STRUCTURED BENCHMARK DATASET

Model	Accuracy	Precision	Recall	F1-score
Logistic Regression	95.12	0.951	0.952	0.951
Random Forest	96.85	0.968	0.968	0.968
XGBoost	**97.21**	0.972	0.972	0.972
BiLSTM	94.89	0.949	0.949	0.949

The confusion matrix for XGBoost (Fig. 3) shows near-perfect classification, with only a small number of phishing sites misclassified as legitimate and vice versa. This indicates strong generalisation, further supported by the learning curve in Fig. 4, which demonstrates stable convergence between training and validation scores.

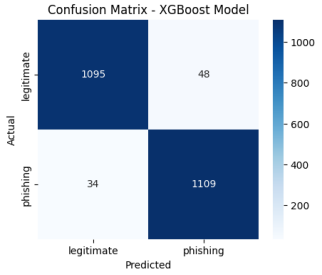

Fig. 3. *
(a) Confusion matrix for XGBoost

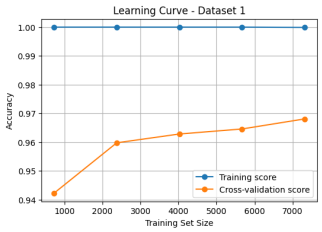

Fig. 4. *
(b) Learning curve

Feature importance analysis (Fig. 5) revealed that external service-based features such as `google_index`, `page_rank`, and `web_traffic` were among the most discriminative, aligning with prior literature that highlights the predictive power of reputation and indexing signals. Lexical features like `nb_qm` (number of question marks) and `nb_www` (occurrences of "www") also ranked highly, suggesting that URL structure remains a strong indicator of phishing attempts. Content-based features such as `nb_hyperlinks` and `empty_title` contributed meaningfully, reinforcing the value of multimodal feature sets even in curated datasets.

C. Raw Multimodal Dataset Results

Table II presents the performance of three models evaluated on the raw multimodal dataset: Neural Network, LightGBM, and XGBoost. Among these, XGBoost achieved the highest overall accuracy of 93.60%, followed closely by LightGBM (93.47%) and the Neural Network (92.24%). In terms of phishing-specific metrics, LightGBM yielded the highest F1-score (0.933) and Precision (0.949), while XGBoost matched its Recall (0.918) and delivered a slightly lower F1-score (0.931). The Neural Network performed competitively, with an F1-score of 0.921.

The confusion matrix for LightGBM (Fig. 6) shows a higher rate of misclassification compared to the structured dataset,

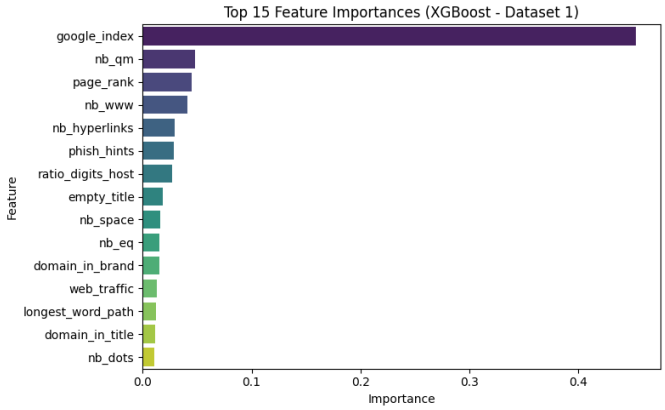

Fig. 5. Feature importance ranking for XGBoost on the structured dataset.

TABLE II
MODEL PERFORMANCE SUMMARY ON RAW MULTIMODAL DATASET

Model	Accuracy	Precision	Recall	F1-score
Neural Network	0.922449	0.932773	0.909836	0.921162
LightGBM	0.934694	0.949153	0.918033	0.933333
XGBoost	0.930612	0.941176	0.918033	0.929461

particularly with legitimate sites being incorrectly flagged as phishing. This reflects the increased difficulty of the task due to the dataset's smaller size, high dimensionality, and heterogeneous feature types.

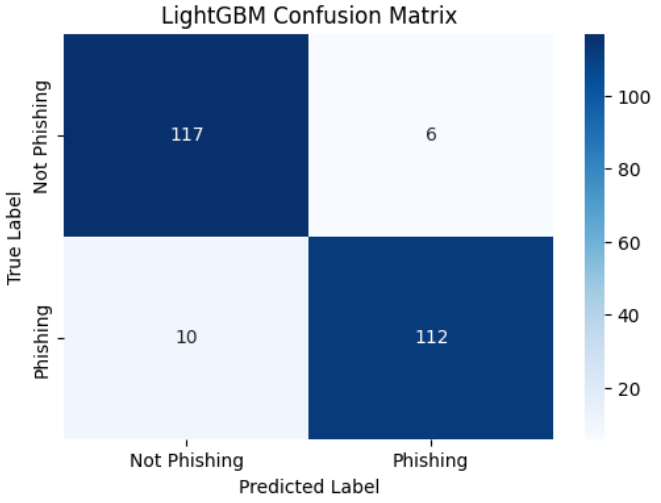

Fig. 6. Confusion matrix for LightGBM on the raw multimodal dataset (Fig. 9b).

Unlike the structured dataset, no single feature group dominated performance. Instead, the combination of HTML/CSS statistics, TF-IDF text features, and visual embeddings contributed collectively. However, the high dimensionality of

TF-IDF and visual features, combined with the limited sample size, likely constrained model generalisation.

D. Cross-Dataset Comparison

Performance was consistently higher on the structured benchmark dataset than on the raw multimodal dataset. This can be attributed to:

- **Data quality** – The benchmark dataset was clean, balanced, and pre-engineered, while the multimodal dataset required extensive preprocessing and still contained noise.
- **Feature composition** – The benchmark dataset's engineered features were highly discriminative, whereas the multimodal dataset's features were more diffuse and high-dimensional.
- **Sample size** – The multimodal subset was much smaller, limiting the ability of models to learn complex patterns.

These results highlight a trade-off: curated datasets enable high accuracy with relatively simple models, while raw, real-world datasets demand more complex preprocessing and may yield lower performance despite richer modalities.

V. Conclusion

This study provides a comprehensive comparison of phishing detection models across structured and raw multimodal datasets. External service based features proved most discriminative in structured data, while multimodal integration offered complementary signals despite increased complexity. The results demonstrate that dataset quality and preprocessing significantly impact model performance, with a clear trade off between data curation effort and achievable accuracy. For practical deployment, practitioners should consider resource constraints, data availability, and performance requirements when selecting between structured feature engineering approaches and raw multimodal processing pipelines.

Future research should focus on developing more efficient multimodal architectures (using transformers). Furthermore, future work could:

- Scale the multimodal experiments to the full dataset.
- Explore dimensionality reduction techniques to mitigate high-dimensional feature challenges.
- Investigate cross-dataset generalisation to assess model robustness.

References

[1] S. S. Shafin, "An explainable feature selection framework for web phishing detection with machine learning," Data Science and Management, vol. 8, pp. 127–136, 2025, doi: 10.1016/j.dsm.2024.08.004.

[2] M. Sánchez-Paniagua, E. Fidalgo, E. Alegre, and R. Alaiz-Rodríguez, "Phishing websites detection using a novel multipurpose dataset and web technologies features," Expert Systems with Applications, vol. 207, art. 118010, 2022, doi: 10.1016/j.eswa.2022.118010.

[3] A. Hannousse and S. Yahiouche, "Towards benchmark datasets for machine learning based website phishing detection: An experimental study," arXiv preprint, arXiv:2010.12847, 2020.

[4] T. Alam, A. Khan, M. Z. Asghar, and S. Khan, "Look Before You Leap: Detecting phishing web pages by exploiting raw URL and HTML characteristics," arXiv preprint, arXiv:2011.04412, 2020.

[5] X. Zhou, Y. Li, and H. Zhang, "An effective detection approach for phishing websites using URL and HTML features," Scientific Reports, vol. 12, art. 8310, 2022, doi: 10.1038/s41598-022-10841-5.

[6] M. Rahman, S. S. Islam, and K. Hossain, "Unveiling suspicious phishing attacks: Enhancing detection with an optimal feature vectorization algorithm and supervised machine learning," Frontiers in Computer Science, vol. 4, art. 1428013, 2024, doi: 10.3389/fcomp.2024.1428013.

[7] N. Nagy, M. Aljabri, A. Shaahid, M. H. Alsharif, and B. A. Alzahrani, "Phishing URLs detection using sequential and parallel ML techniques: Comparative analysis," Sensors, vol. 23, no. 7, art. 3467, 2023, doi: 10.3390/s23073467.

[8] H. A. Shaik, S. D. Kaleb, G. D. Upadhye, and A. A. Shaikh, "Phishing URL detection using machine learning methods," Advances in Engineering Software, vol. 173, art. 103288, 2022, doi: 10.1016/j.advengsoft.2022.103288.

[9] A. Kumar and S. Singh, "Multimodel phishing URL detection using LSTM, Bi-LSTM, and GRU models," Future Internet, vol. 14, no. 11, art. 340, 2022, doi: 10.3390/fi14110340.

[10] Shashwat Work, "Web Page Phishing Detection Dataset," Kaggle, [Online]. Available: https://www.kaggle.com/datasets/shashwatwork/web-page-phishing-detection-dataset. [Accessed: Sep. 10, 2025]

[11] A. AlEroud, "Phishing Website Dataset," Zenodo, Jun. 2023. [Online]. Available: https://zenodo.org/record/8041387. [Accessed: Sep. 10, 2025]