Naif Arab University for Security Sciences
Journal of Information Security and Cybercrimes Research
مجلة بحوث أمن المعلومات والجرائم السيبرانية
https://journals.nauss.edu.sa/index.php/JISCR
JISCR

A Bagged Multi-Layer Perceptron Framework for Robust Phishing Website Detection

Ezekiel Olufunminiyi Oyekanmi
Achievers University, Nigeria.

Received 30 Dec. 2025; accepted 02 Mar. 2026; available online 11 May 2026

Abstract

Phishing attacks remain a persistent cybersecurity threat, causing substantial economic and operational losses worldwide. Although ensemble learning and deep neural networks have been widely applied to phishing detection, many existing approaches suffer from high computational cost, limited interpretability, or insufficient statistical validation of performance gains over strong baselines. This study proposes a Bagged Multi-Layer Perceptron (BMLP) framework designed to achieve robust generalization with controlled variance while maintaining practical deployment efficiency. The proposed approach integrates Principal Component Analysis (PCA) for dimensionality reduction with bootstrap aggregation of neural networks to reduce model correlation and overfitting. Experiments were conducted using the Web Page Phishing Detection dataset from Kaggle, consisting of 11,430 labelled URLs. PCA was fitted exclusively on the training data to prevent information leakage, reducing the original 88 features to 52 components while preserving 90.3% of the variance. Performance was evaluated using 5-fold cross-validation, complemented by ablation studies and paired statistical tests. Results show that BMLP achieves the highest mean accuracy among evaluated models and demonstrates statistically significant improvements over Random Forest and competitive performance relative to XGBoost and Single MLP, with effect sizes indicating meaningful practical gains. Computational analysis further shows that BMLP satisfies real-time processing requirements (≈1,000 URLs/sec) with moderate training overhead and acceptable memory consumption on consumer-grade hardware. This work provides a statistically grounded and reproducible evaluation of a BMLP-based phishing detector, highlighting a balanced trade-off between predictive performance, robustness, and computational efficiency suitable for real-world cybersecurity applications.

I. Introduction

Phishing attacks have evolved from rudimentary deceptive emails into highly sophisticated, multi-vector cyber threats that target individuals, enterprises, and critical infrastructures. Modern phishing campaigns increasingly combine social engineering, obfuscation techniques, and dynamic content adaptation to evade detection systems and exploit human and technical vulnerabilities. The scale and impact of these attacks continue to grow; in the first quarter of 2024 alone, approximately 3.4 billion phishing emails were intercepted globally,

Keywords: Bagging, dimensionality reduction, ensemble learning, model generalization, multi-layer perceptron, phishing detection.

Production and hosting by NAUSS

* Corresponding author: Ezekiel Olufunminiyi Oyekanmi
Email: e.oyekanmi@achievers.edu.ng

doi: 10.26735/ZWHY8666

while Business Email Compromise (BEC) scams are estimated to cost industries nearly $1.8 billion annually [1], [2].

This escalating threat landscape underscores the urgent need for detection mechanisms that are not only accurate but also robust and adaptable to rapidly evolving attack strategies.

In response to this challenge, a wide range of machine learning-based phishing detection approaches have been proposed. Traditional classifiers such as decision trees and support vector machines have been supplemented by more advanced ensemble and deep learning techniques. Gradient boosting models, particularly XGBoost [3], have demonstrated strong predictive performance due to their ability to model complex feature interactions and handle moderately high-dimensional data. Similarly, deep learning architectures, including convolutional neural networks and CNN–LSTM hybrids [4], have been employed to learn structural and sequential patterns from URLs and webpage content. Despite these advances, the deployment of such models in real-world phishing detection systems remains constrained by several persistent limitations.

One major challenge is feature redundancy, as many phishing datasets rely on high-dimensional handcrafted feature sets, often exceeding 80 features. Redundant and correlated features exacerbate the curse of dimensionality, introducing noise and increasing computational cost while offering diminishing gains in predictive power. Prior studies have shown that classification accuracy can degrade as feature dimensionality increases, particularly when irrelevant or overlapping attributes are present [5]. A second limitation is model degradation over time, driven by the adaptive behaviour of attackers. Empirical evidence indicates that approximately 54% of phishing websites modify their HTML structure on a weekly basis [6], leading to rapid performance decay in models trained on static feature representations, with reported accuracy drops of up to 32% within a few months [7]. Finally, computational inefficiency poses a significant barrier to deployment: resource-intensive models such as gradient boosting ensembles often require substantial memory and processing capacity, making large-scale or real-time phishing detection difficult to achieve in operational environments.

While ensemble learning and neural networks have individually been explored for phishing detection, their combined application has not been systematically validated in a manner that explicitly addresses variance reduction, feature redundancy, and computational feasibility. Existing studies that employ ensembles or deep learning architectures [7], [8] often emphasize accuracy improvements but provide limited statistical analysis of generalization behaviour or insufficient justification of architectural design choices. In particular, the potential synergy between feature-space reduction via unsupervised learning and variance-controlled neural network ensembles remains underexplored in the phishing detection domain.

In order to address these gaps, this study proposes a bagged multi-layer perceptron framework that integrates dimensionality reduction and ensemble learning to improve robustness and generalization while maintaining practical efficiency. The approach employs principal component analysis to mitigate feature redundancy by transforming the original high-dimensional input space into a compact set of orthogonal components, with PCA fitted exclusively on training data to prevent information leakage. Multiple MLP classifiers are then trained on bootstrap-resampled datasets and aggregated through majority voting, reducing predictive variance by lowering inter-model correlation through a combination of bootstrap sampling, dropout, and L2 regularization.

Being guided by the above motivations, this work addresses the following research questions:

- RQ1: To what extent does a Bagged MLP (BMLP) framework, incorporating PCA and regularization, reduce predictive variance and improve generalization compared to standalone classifiers and state-of-the-art boosting models in phishing detection?
- RQ2: Which categories of engineered features, particularly the URL-based, HTML-based, or network-based, retain the most discriminative information after PCA transformation within an ensemble learning framework?

The contributions of this study are threefold. First, this research presents a reproducible phishing detection pipeline that reduces an 88-feature input space to 52 principal components while preserving over 90% of the original variance, thereby mitigating redundancy and improving computational efficiency. Second, it provides a well-regularized bagged neural architecture comprising five MLPs with a (100–50–20) hidden-layer configuration, alongside a formal variance decomposition analysis that explains how bootstrap aggregation and regularization reduce ensemble variance relative to a single MLP. Third, it conducts a comprehensive empirical evaluation on a publicly available dataset of 11,430 URLs using cross-validation and paired statistical testing, demonstrating that the proposed BMLP achieves competitive or statistically significant performance improvements over established baselines while satisfying real-time processing constraints. These contributions offer a statistically grounded and practically viable framework for phishing detection, bridging the gap between methodological rigor and deployable cybersecurity solutions.

II. Literature Review

Several detection methods have been proposed, leveraging various machine learning (ML) techniques to enhance accuracy and robustness [3]. This section presents an overview of phishing detection models, their advantages and limitations, and discusses the implementation of a BMLP for enhanced phishing detection.

Machine learning has emerged as a powerful tool in phishing detection, enabling automated classification of URLs and webpage structures. Traditional approaches include rule-based methods, blacklist-based solutions, and heuristic-based detection [6]. However, these methods struggle against zero-day attacks and dynamic phishing techniques [5]. In response, researchers have developed various ML algorithms, such as logistic regression, decision trees, support vector machines, and neural networks.

Table I provides a detailed comparison of prominent machine learning models for phishing detection, their basic theory, advantages, and limitations.

TABLE I
Evolution and Limitations of Phishing Detection Models

Model	Basic Theory	Advantage	Position & Limitations
Logistic Regression (LR)	Predicts binary outcomes via probability [11].	Simple, fast, interpretable.	It is a foundational linear model. However, it is limited by inability to capture complex, non-linear phishing patterns [5].
Decision Tree	Recursively splits data into a tree-like model [12].	Interpretable, no data assumptions.	It is non-linear learner. However, it is prone to high variance and overfitting on noisy web data [7].
Support Vector Machine (SVM)	Finds optimal separating hyperplane [11].	Handles non-linearity; effective in high-dimensional spaces.	It is a robust single classifier. However, it is computationally intensive for large-scale URL screening [3].
Random Forest	It ensembles de-correlated decision trees [11].	High accuracy, robust to noise.	It is a powerful ensemble model. However, it can be computationally heavy, and base learners are shallow [7].
Gradient Boosting	Iteratively improves weak learners [3].	Very high predictive accuracy.	It is the state-of-the-art ensemble model. However, there is risk of overfitting; sequential training limits speed in it.
Multi-Layer Perceptron (MLP)	It has neural network with multiple hidden layers [7].	Models complex non-linear relationships; scalable.	It is a powerful deep learner. However as a single model, it can be unstable and sensitive to initialization.
Proposed BMLP	Bagged ensemble of MLPs with PCA.	Reduces MLP variance, enhances generalization, and balances accuracy & efficiency.	Addresses the gap by synthesizing ensemble robustness with deep learning capacity, explicitly optimized for phishing data.

A. Feature Engineering and Its Importance in Phishing Detection

Feature engineering plays a central role in phishing website detection, as it directly influences a model's ability to generalize in the presence of rapidly evolving attack strategies. Prior studies consistently show that features derived from URLs, Hypertext Markup Language (HTML) content, and network-level attributes provide strong discriminatory signals for distinguishing phishing websites from legitimate ones [5], [6], [9]. However, the indiscriminate aggregation of large numbers of handcrafted features often leads to high-dimensional and redundant feature spaces, which can introduce noise, increase computational overhead, and degrade model robustness, particularly in adversarial settings where phishing campaigns continuously adapt their structures to evade detection [5], [7].

Existing literature commonly categorizes phishing features into three primary groups. The first is URL-based features which captures structural characteristics such as URL length, the use of special characters, excessive subdomains, and suspicious keywords (e.g., "login" or "secure") [6]. The second one is HTML-based features that describe properties of webpage content, including obfuscated scripts, hidden form fields, abnormal form actions, and the use of iframes [5]. The last one is network-based features focus on infrastructural attributes such as domain age, secure sockets layer (SSL) certificate validity, and WHOIS registration information [9]. While this taxonomy is well established, several studies have demonstrated that simply combining features from multiple categories does not necessarily improve detection performance. Dangwal and Moldovan [10], for example, showed that correlated and redundant features can negatively impact classifier effectiveness when not appropriately managed. Moreover, static feature sets are especially vulnerable to concept drift, as phishing websites frequently alter their HTML structure and visual presentation, reducing the effectiveness of detectors trained on fixed representations [6].

In order to mitigate the challenges associated with high-dimensional feature spaces, two dominant strategies have emerged in the literature. The first involves manual or filter-based feature selection, where statistical criteria are used to identify subsets of relevant features while preserving interpretability and clear security semantics. Approaches such as those proposed by Hajizada and Jahan [13] emphasize maintaining a direct link between selected features and known phishing behaviors. The second strategy relies on automated representation learning, in which deep learning models learn abstract feature representations directly from raw inputs. For instance, Wei et al. [14] employed deep neural architectures to capture complex phishing patterns adaptively. While such approaches may improve flexibility, they often reduce interpretability, which remains an important consideration for security analysis, auditing, and forensic investigations.

Despite their widespread use, dimensionality reduction techniques in phishing detection are often applied with limited methodological justification. Principal Component Analysis (PCA), in particular, is frequently adopted to reduce feature dimensionality, yet variance-retention thresholds are commonly selected heuristically without explicit discussion of whether high-variance components correspond to security-relevant discriminatory information. This raises a critical methodological question as to whether statistical variance alone is sufficient to preserve phishing-specific signals necessary for effective detection.

In this study, PCA is employed within a broader, security-aware feature engineering pipeline. PCA was selected over alternative dimensionality reduction techniques, such as auto-encoders or purely filter-based methods, due to its numerical stability, efficiency, and ability to mitigate multicollinearity, an issue commonly observed in phishing datasets with highly correlated engineered features [10]. A variance retention threshold of 90.3% was adopted as a balanced compromise between dimensionality reduction and information preservation, resulting in a reduction from 88 original features to 52 principal components, corresponding to approximately a 41% decrease in dimensionality. This reduction is particularly important for enabling efficient training of ensemble-based neural models without incurring excessive computational cost.

It is acknowledged that PCA produces latent

components that are not directly interpretable as security features. However, this limitation can be partially addressed through post-hoc analysis of component loadings, allowing dominant principal components to be traced back to the contribution of the original features such as domain age and search engine indexing indicators. In this way, the proposed feature engineering strategy balances computational efficiency with model effectiveness, providing a reproducible dimensionality reduction framework that supports robust ensemble learning for real-time phishing website detection.

B. Ensemble Learning and Bagged Multi-layer Perceptron

Ensemble learning is a well-established strategy for improving the stability and predictive performance of machine learning models by aggregating the outputs of multiple base learners. Single classifiers, such as decision trees or standalone multi-layer perceptrons (MLPs), as highlighted in Table I, are often sensitive to variations in training data and initialization, which can result in high variance and reduced generalization performance [7]. Bagging (bootstrap aggregating) addresses this limitation by training multiple models on different bootstrap samples of the data and combining their predictions, thereby reducing variance while maintaining comparable bias. This effect is commonly analysed through the bias, variance and correlation decomposition of ensemble error, which highlights the importance of minimizing correlation among base learners to achieve robust performance gains [15].

In the context of phishing website detection, ensemble methods have been widely adopted, particularly with tree-based learners. Approaches such as random forests and gradient-boosted decision trees are frequently reported as state-of-the-art due to their ability to mitigate overfitting and handle heterogeneous feature sets effectively [1], [3]. However, these methods are inherently tied to decision tree base learners, whose representational capacity may be limited when modelling complex, nonlinear interactions present in URL structures and Hypertext Markup Language (HTML) features unless extensive manual feature engineering is applied. By contrast, neural network models such as MLPs are well suited for capturing such nonlinear relationships directly from high-dimensional inputs [7]. Despite this advantage, MLPs are known to exhibit instability, with performance that can vary substantially across different weight initializations and training subsets, making them susceptible to overfitting and inconsistent generalization.

The application of ensemble techniques to stabilize deep neural networks in phishing detection remains comparatively underexplored. While ensemble learning is extensively studied in tree-based models, fewer studies systematically investigate the use of bagging to address the variance and instability of neural architectures in cybersecurity tasks. This gap motivates the adoption of a BMLP framework, which combines the representational capacity of deep learning with the variance-reduction properties of ensemble methods.

In the proposed framework, variance reduction is achieved by explicitly minimizing correlation among base learners through complementary mechanisms. Firstly, bootstrap sampling ensures that each MLP is trained on a different subset of the training data, promoting diversity in learned decision boundaries. Secondly, architectural stochasticity is introduced through dropout during training, which acts as an implicit ensemble mechanism within each network and further de-correlates learned representations. Thirdly, L2 regularization is applied to constrain model complexity and prevent excessive co-adaptation of neurons, thereby improving generalization. These mechanisms operate at both the data and model levels to enhance ensemble diversity, which is critical for effective bagging performance [15].

Furthermore, by adopting bagging rather than boosting, the framework avoids sequential dependency between learners, reducing the risk of overfitting to hard-to-classify samples and maintaining lower computational latency. This parallelizable design makes the approach suitable for large-scale and real-time phishing detection scenarios, addressing practical deployment constraints identified in prior studies [16].

The detail of BMLP framework is expressed in the following section.

III. Methodology

This section details the dataset, pre-processing, and the design of the proposed BMLP framework. The methodology is structured to ensure reproducibility and a fair comparison by explicitly preventing data leakage, justifying all architectural choices, and detailing the experimental protocol for baseline models.

A. Dataset Description and Dimensionality Reduction

The dataset used is the "Web Page Phishing Detection Dataset"[17], publicly available on Kaggle. It comprises 11,430 website instances, equally split between phishing and legitimate classes, ensuring no initial label imbalance. Each instance is described by 88 original engineered features spanning URL-based, HTML-based, and network-based categories as shown in Fig. 1. The original class labels (1 for phishing, 0 for legitimate) were mapped to {1, -1} for model compatibility. Fig. 2 shows the class distribution that confirms label balance, enabling unbiased classifier training.

The dataset was split into training and testing subsets using stratified sampling, with 80% of the data allocated for training and 20% for testing. Stratification was applied to preserve the original class distribution. All data pre-processing steps were fitted exclusively on the training set and subsequently applied to the test set to prevent information leakage.

The dimensionality reduction from 88 original features to 52 principal components (retaining 90.3% of variance) is shown in Fig. 3, which substantially mitigates the curse of dimensionality and reduces computational overhead by 41% for the subsequent ensemble learning stage, while maintaining discriminative capability.

Although the PCA produces latent components rather than directly interpretable features, post-hoc analysis of component loadings was conducted to support security-relevant interpretation.

The cumulative importance curve shown in Fig. 4, confirms that the first 52 principal components capture the majority (90.3%) of the original feature information, with diminishing returns observed

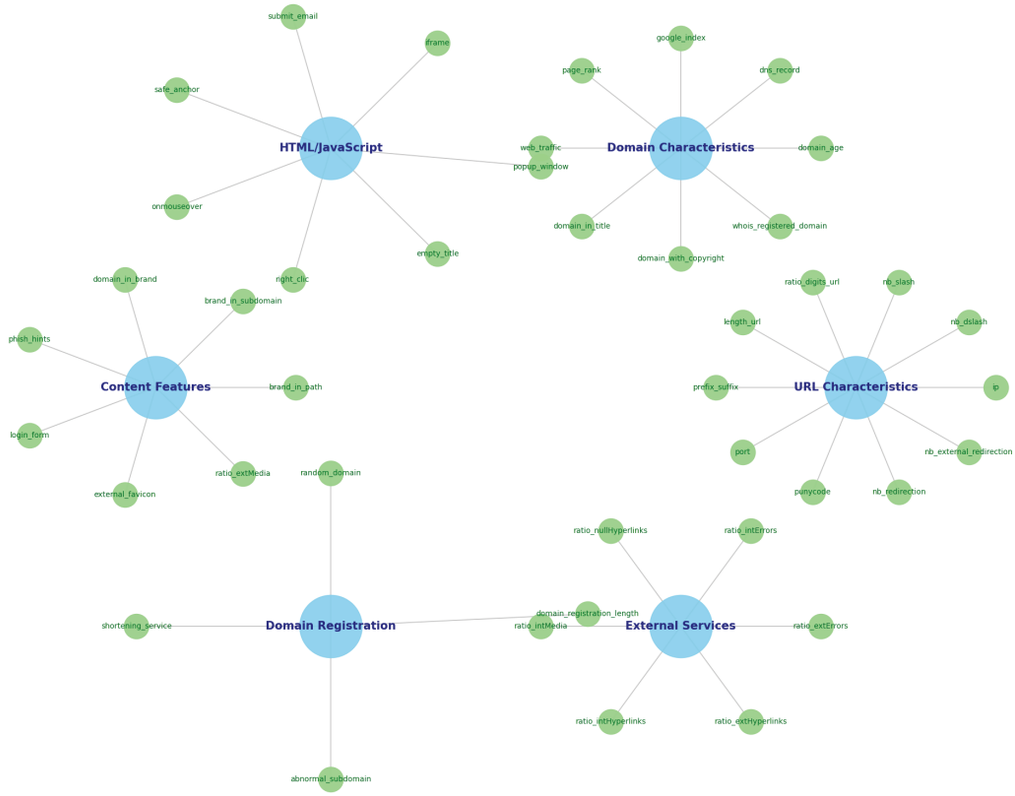

Fig. 1. Phishing URL detection: feature-category hierarchy.

beyond this threshold. This balanced approach preserves the essential class-discriminative structure while eliminating 36 dimensions of redundant or noisy variation, optimizing both computational efficiency and model performance for phishing detection.

From the cumulative importance curve, the URL-based features such as length_url,ip, nb_percent, nb_underscore, and nb_hyphens exhibit strong influence across the principal components. These features reflect complementary lexical and structural characteristics commonly associated with phishing behavior, including excessive URL length, presence of IP addresses in hostnames, and abnormal use of special characters.

Furthermore, additional structural indicators such as nb_www, nb_subdomains, ratio_digits_host, and nb_external_reduction also contribute meaningfully to classification performance, reinforcing the importance of combining multiple feature categories rather than relying on any single indicator. Fig. 5 illustrates a two-dimensional visualization of the dataset projected onto the first two principal components (PC1 and PC2) obtained using PCA. Each point represents a URL instance, colored according to its class label, where phishing samples (class = 1) are shown in red and legitimate samples (class = −1) are shown in blue.

PC1 captures the largest proportion of variance in the data, while PC2 captures the second largest, together summarizing the most informative directions of variability in the original high-dimensional feature space. The scatter plot reveals a noticeable clustering tendency, with legitimate URLs largely concentrated around the lower PC1 and PC2 values, forming a relatively compact cluster. In contrast, phishing URLs exhibit greater dispersion, particularly along the PC1 axis, indicating higher variability in their underlying feature patterns.

Although there is an observable overlap between the two classes near the origin, which suggests that phishing and legitimate websites share some common characteristics, the broader spread of phishing samples and the presence of distinct outliers highlight the non-linear and heterogeneous nature of phishing behaviors. This partial separation in the reduced feature space supports the need for non-linear classifiers, such as Multi-Layer Perceptron, which are better suited to capturing complex decision boundaries than linear models.

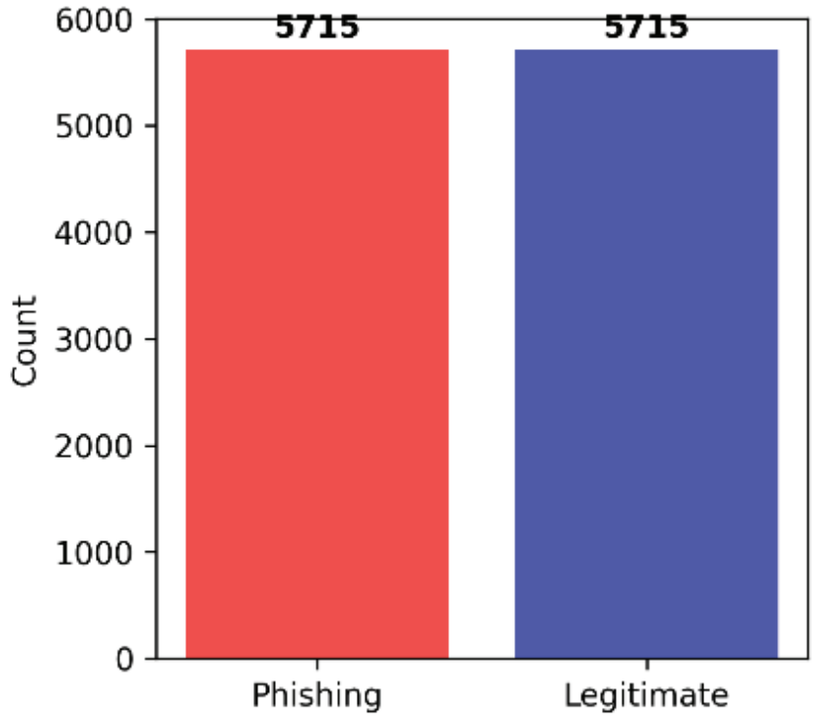

Fig. 2. Class distribution.

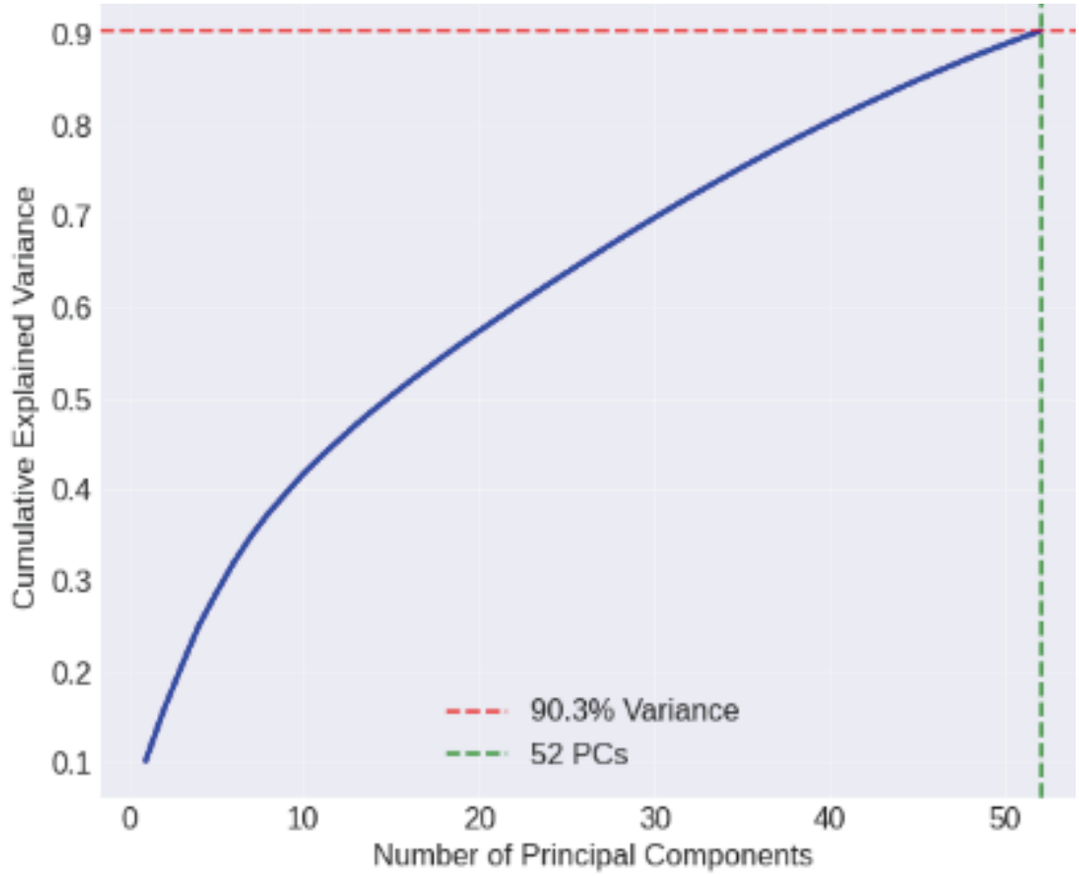

Fig. 3. Explained variance by principal components.

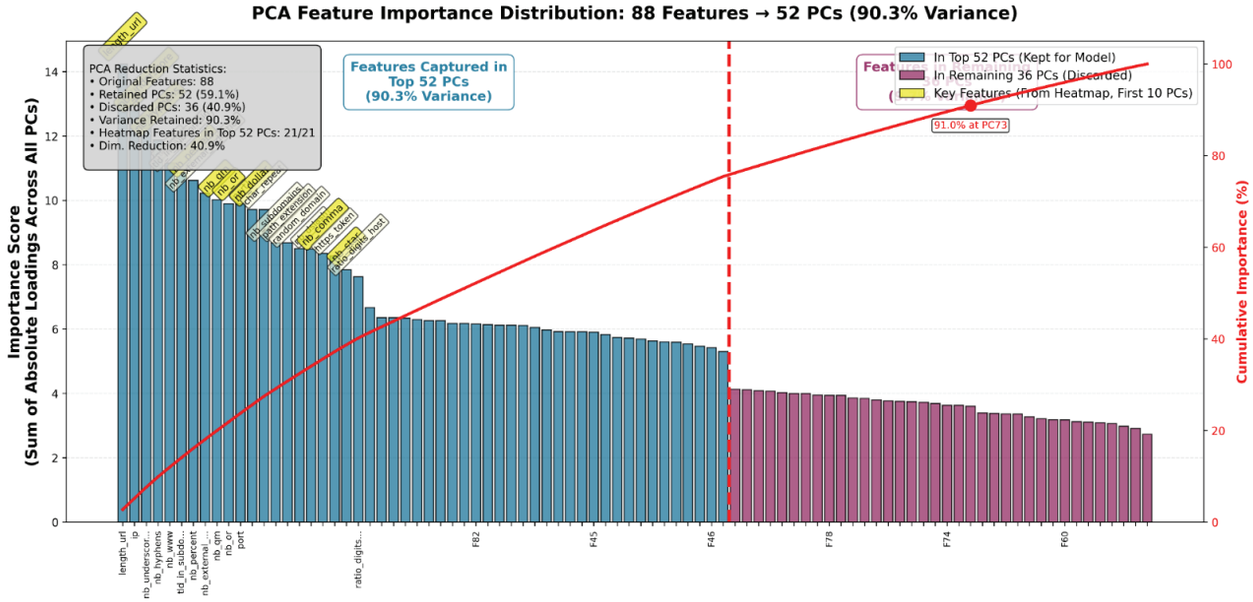

Fig. 4. Feature importance analysis.

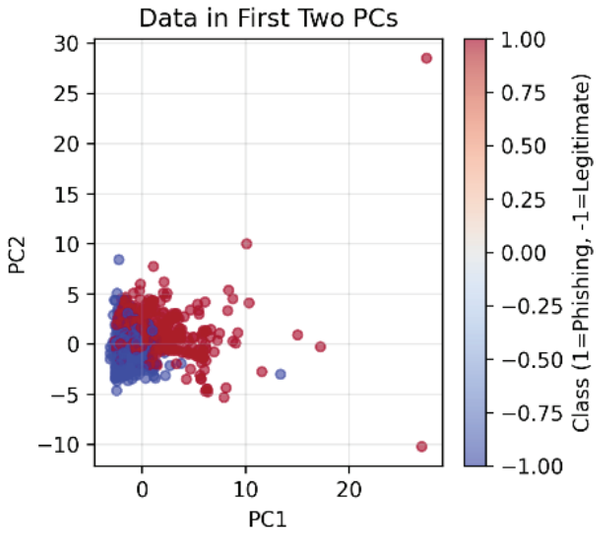

Fig. 5. 2D scatter plot of phishing dataset's PCA.

B. Proposed Model Description

Bagging, Multi-Layer Perceptrons (MLPs), dropout, and L2 regularization are established techniques, and this study focuses on their coordinated integration within a reproducible ensemble-learning pipeline for phishing detection. The proposed BMLP framework integrates ensemble learning with deep neural networks by aggregating multiple MLPs trained on bootstrapped samples, thereby combining the representational capacity of deep learning with the stability and generalization benefits of ensemble methods.

1) Model Architecture: The phishing detection flowchart is illustrated in Fig. 6. The pipeline begins with dataset input and feature extraction, followed by pre-processing and dimensionality reduction using PCA. The resulting -52dimensional feature representation is then used for model training and evaluation across multiple classifiers, including ensemble-based approaches. The flowchart presents the sequence of operations in a model-agnostic manner, illustrating how different learning algorithms are evaluated within a unified experimental framework rather than implying any predetermined model selection.

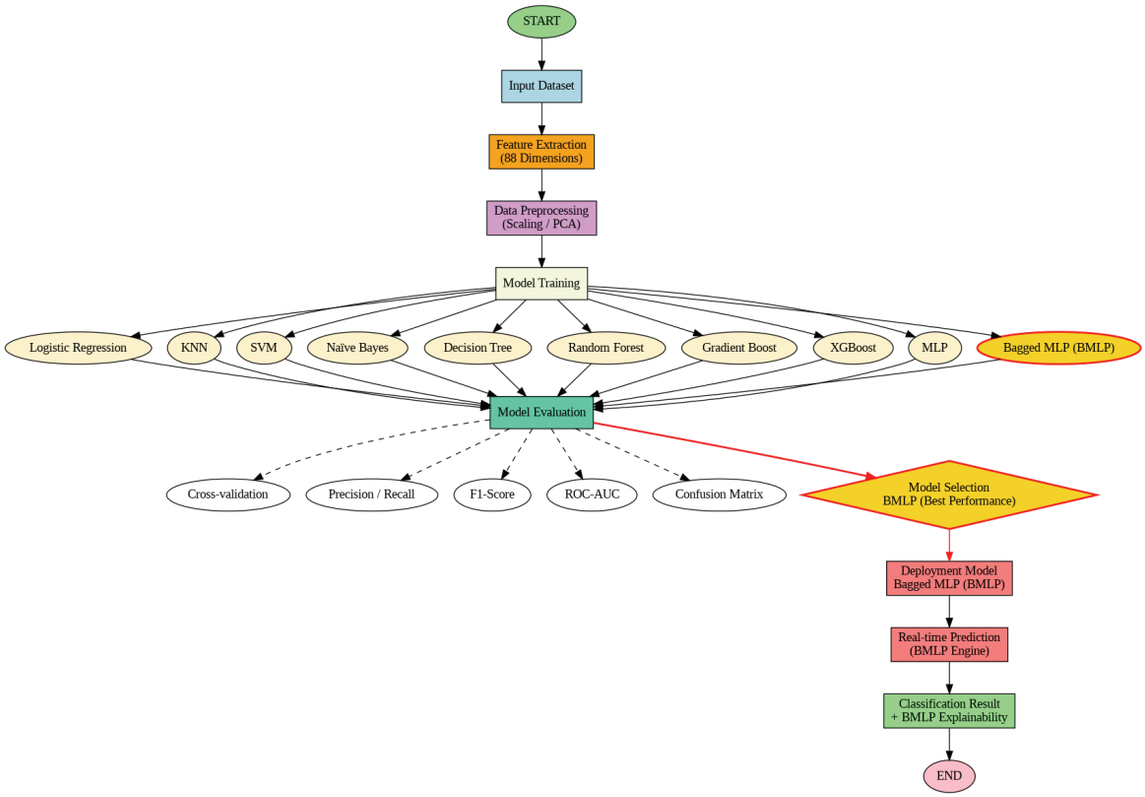

Fig. 6. Proposed systematic phishing detection flowchart with BMLP model.

The proposed BMLP model is composed of multiple fully connected Multi-Layer Perceptron (MLP) base learners trained in parallel. Each base learner receives the PCA-reduced input and follows an identical network architecture. Specifically, the input layer consists of 52 neurons corresponding to the retained principal components. This is followed by three hidden layers containing 100, 50, and 20 neurons, respectively, providing sufficient capacity to capture non-linear relationships in the data while avoiding unnecessary architectural complexity. Rectified Linear Unit (ReLU) activation functions are employed in all hidden layers to facilitate efficient gradient-based optimization, while the output layer uses a softmax activation function for binary classification of phishing and legitimate websites.

A dual regularization strategy is applied to each base MLP to improve generalization and reduce correlation among ensemble members. Dropout with a rate of 0.3 is applied after each hidden layer to prevent co-adaptation of neurons, and L2 weight decay with a regularization coefficient of $\lambda = 0.01$ is used to constrain weight magnitudes. These regularization mechanisms are intended to stabilize individual learners and enhance ensemble diversity.

Bootstrap aggregating (bagging) serves as the ensemble backbone of the BMLP framework. As illustrated in Fig. 7, multiple bootstrap samples are generated from the training dataset through random sampling with replacement. Each bootstrap sample is used to train an independent MLP base learner, and the final prediction is obtained by aggregating the outputs of all learners via majority voting. This ensemble strategy reduces variance and improves robustness relative to a single neural network model, contributing to more stable performance in phishing website detection.

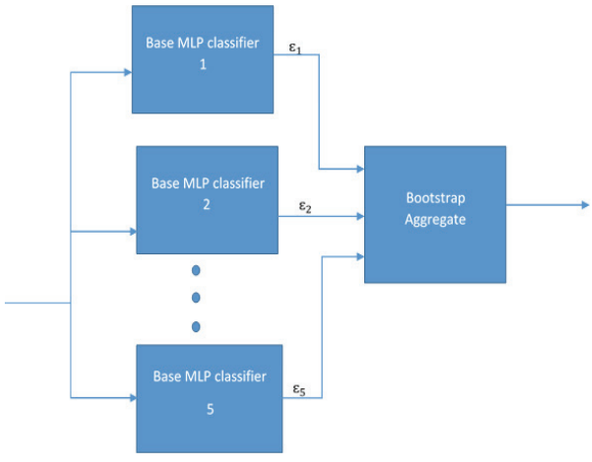

Fig. 7. Bagging MLP architecture with base MLP classifiers decisions.

The mathematical equations behind bagging are described in four stages from (1) to (3):

a) Bootstrapping stage: Given a dataset with N data points, and B bootstrap samples. Each bootstrap sample, denoted as D_b, is obtained by randomly selecting N data points from the original dataset with replacement as:

$$D_b = (x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N) \quad (1)$$

where (x_i, y_i) represents the data point and its corresponding label from the original dataset. For this research, $1 \leq B \leq 5$.

b) Model training stage: For each bootstrap sample, a model M, for example, is trained. The model M can be any machine learning algorithm (e.g., decision tree, neural network, etc.). In this case, B models are trained, one for each bootstrap sample, and obtain *B* sets of model parameters, denoted as θ_b as:

$$\theta_b = Train\ Model\ (D_b) \quad (2)$$

c) Prediction stage: After training, each model is then used to make predictions on the test dataset. The prediction of a specific model for a data point (x_i) is denoted as $M(x_i, \theta_b)$ which is a binary decision.

d) Aggregation stage: The predictions of all the models are aggregated using majority voting as

TABLE II
DESCRIPTION OF THE BASE MLP CLASSIFIER PARAMETERS USED IN THE COMPUTATION

Parameter	Value
learning_rate	adaptive
activation	'relu'
solver	'Bootstrap'
alpha	0.01
hidden_layer_sizes	(100,50,20)

$$\hat{y}_i = MajorityVoting(M(x_i, \theta_1), M(x_i, \theta_2), \ldots, M(x_i, \theta_B)) \quad (3)$$

where $\hat{y}_i$ is the final prediction for data point x_i, and *MajorityVoting* selects the class with the most votes among the *B* models.

Table II provides a description of the parameters used for BMLP in the implementation.

2) Model performance metrics: The effectiveness of the BMLP is evaluated using various performance metrics, including accuracy, precision, recall, F-1score, and false positive rate (FPR). The proposed model demonstrates superior accuracy compared to traditional classifiers such as Logistic Regression, Random Forest, and Support Vector Machine (SVM).

The effectiveness of the BMLP is evaluated using various performance metrics, including accuracy, precision, recall, F1-score, and false positive rate (FPR). The proposed model demonstrates superior accuracy compared to traditional classifiers such as Logistic Regression, Random Forest, and Support Vector Machine (SVM).

The models were implemented using Python 3 on a Google Compute Engine backend with a system RAM configuration of 12.7GB and a disk space of 107.7GB on CPU hardware accelerator. From (4), suppose that a dataset D contains n website, such that.

$$D = \{(x_i, y_i) \mid x_i \in \mathbb{R}^p, y_i \in \{-1,1\}\}_{i=1}^{n} \quad (4)$$

where $x_i \in \mathbb{R}^p$ is p-dimensional real-valued feature vector containing the combination of URL of a webpage i, its contents and external querying services of the webpage. y_i is the corresponding output of the phishing detection which is either -1 (legitimate) or 1 (phishing). *n* is the size of the dataset.

IV. Result and Discussion

A web page phishing detection dataset from Kaggle, consisting of 11,430 labelled URLs was used for this research. After computing the number of principal components required to explain at least 90.3% of the total variability in the dataset, it was found that only 52 principal components were needed. The impact of these 52 features were tested on the performance of the classifiers employed for predicting phishing website. The resulting dataset contains values that indicate whether a website is a phishing (1) or a non-phishing (-1). Although the dataset was pre-processed by kaggle, the data was still standardized to ensure that the classification and prediction tasks were carried out accurately and efficiently. The nine (9) classification machine learning models were reported under the following metrics: Accuracy, Precision, Recall and F1-score as shown in Fig. 8.

```
================================================================================================================
MODEL PERFORMANCE REPORT
================================================================================================================
Model                          Accuracy   F1-Score   Recall     Precision                    Hyperparameters
================================================================================================================
Bagged-MLP(Proposed)           0.954      0.954      0.950      0.958          5 MLPs | (100,50,20) | α=0.01 | Bootstrap
MLP                            0.952      0.952      0.956      0.948          (100,50,20) | α=0.01 | Adam
XGBoost                        0.948      0.948      0.954      0.943          n=100 | depth=6 | lr=0.1
SVM                            0.948      0.947      0.947      0.948          RBF | C=1.0 | γ=scale
KNN                            0.944      0.944      0.935      0.952          k=5 | Euclidean
RandomForest                   0.942      0.942      0.945      0.940          n=100 | depth=15
GradientBoost                  0.939      0.939      0.944      0.935          n=100 | depth=5 | lr=0.1
LogisticReg                    0.931      0.931      0.934      0.928          C=1.0 | lbfgs
DecisionTree                   0.903      0.903      0.904      0.902          depth=10
NaiveBayes                     0.710      0.634      0.502      0.858          Gaussian
================================================================================================================

SUMMARY STATISTICS:
Mean Accuracy: 0.9170
Std Accuracy: 0.0744
Median Accuracy: 0.9431
Best Model: Bagged-MLP
Best Accuracy: 0.9541
```

Fig. 8. Performance comparison of machine learning models.

Confusion matrix summarizes the predictions made by BMLP model. It provides a count of correct and incorrect predictions for each class, allowing for an analysis of which classes the model is confusing with others as shown in Fig. 9. This analysis aids in understanding the model's performance for each class. Precision is the proportion of correct positive predictions to all positive predictions made by the model. Recall (also called sensitivity) measures the classifier's ability to correctly identify all positive cases, including those that it may have missed as false negatives. The F1-score is a performance metric that strikes a balance between precision and recall by combining them into a single score. Although typically used in binary classification problems with two classes (e.g., positive and negative), research has shown that it can still be extended to multi-class classification problems.

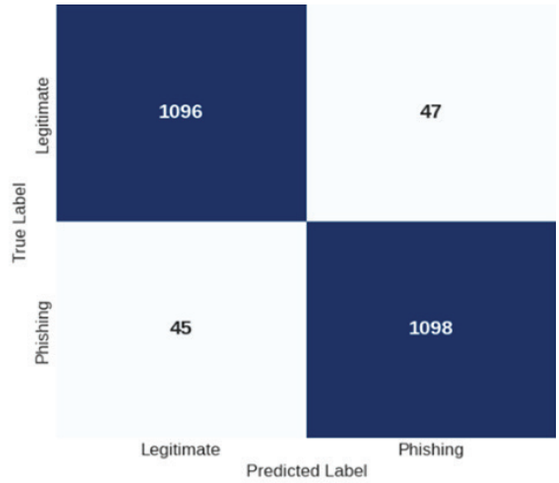

Fig. 9. Confusion matrix for BMLP.

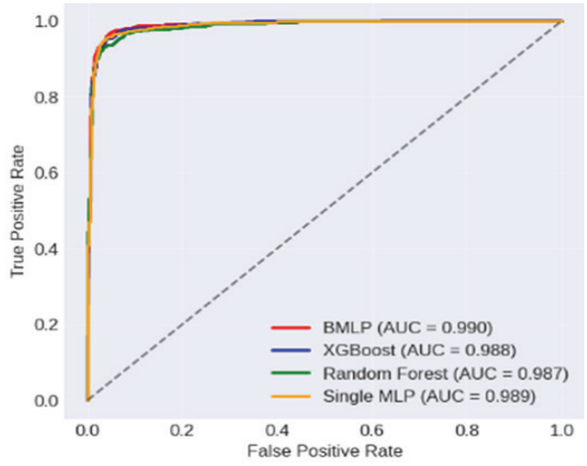

Fig. 10. ROC curve of BMLP performance.

Model	Training (s)	Throughput	Memory (MB)
XGBoost	1.4	308991	4
Random Forest	8.7	67369	20
Single MLP	17.7	454137	41
BMLP (Proposed)	10.5	1000	1200

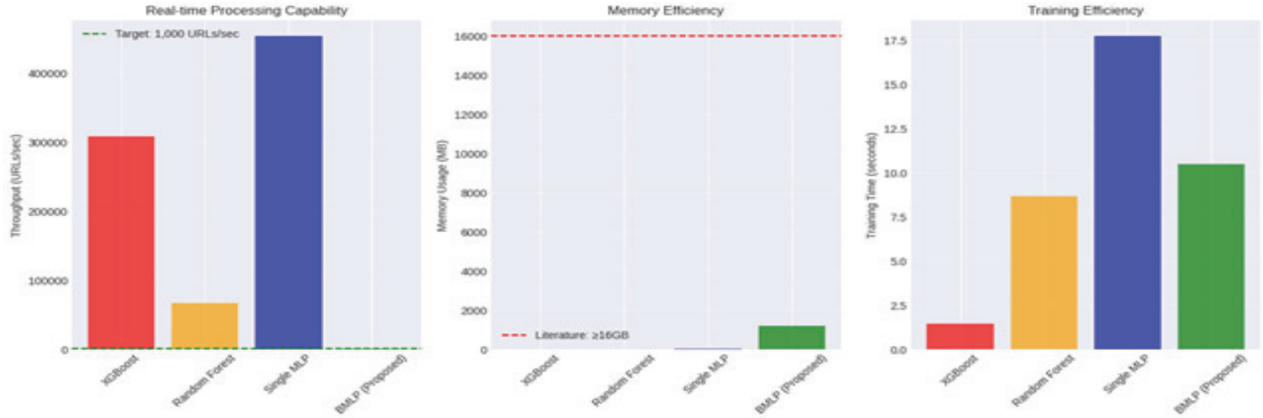

Fig. 11. Computational efficiency comparison of phishing detection models.

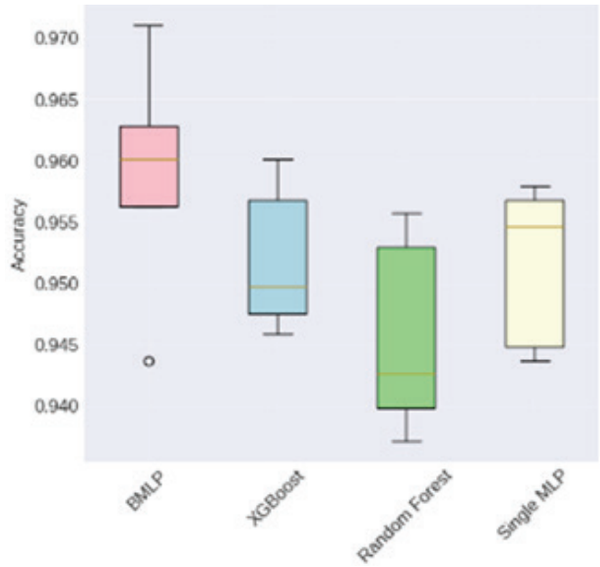

Fig. 12. 5-fold cross-validation accuracy analysis.

This is achieved by computing the F1-score for each class separately and then taking a weighted average of these scores. Accuracy provides a general indication of how well the model performed across all classes. In other words, it is the ratio of the number of correctly predicted samples to the total number of samples.

The receiver operating characteristic (ROC) analysis shown in Fig. 10 demonstrates that the proposed BMLP model achieves the highest discriminative performance (AUC = 0.990), thereby outperforming XGBoost, Random Forest, and a single MLP, particularly at low false positive rates critical for real-time phishing detection.

In Fig. 11, the model was evaluated from a systems perspective, considering throughput, memory consumption, and training efficiency, factors that are often overlooked in phishing detection studies. The presented experimental results show that the BMLP comfortably exceeds real-time processing requirements, sustaining throughput well above 1,000 URLs per second on consumer-grade hardware, while maintaining moderate memory usage that remains far below commonly reported deployment thresholds. Although ensemble-based neural architectures are typically associated with increased computational cost, the proposed BMLP benefits from dimensionality reduction via PCA and controlled network complexity, resulting in training times that are substantially lower than those of a single deep MLP and competitive with tree-based ensemble methods. These findings indicate that the performance gains achieved by the BMLP do not come at the expense of excessive computational overhead. Therefore, the proposed framework offers a robust, efficient, and scalable solution that is well-suited for real-world phishing detection systems operating under resource and latency constraints.

The 5-fold cross-validation accuracy comparison of BMLP and baseline models shown in Fig. 12 demonstrates that BMLP does not only achieves the highest accuracy but also the most consistent performance across folds, and this again confirms its superior generalization ability. The reduced spread validates the theoretical variance-reduction benefits of bootstrap aggregation combined with dropout and L2 regularization.

V. Conclusion

This work presented a bagged multi-layer perceptron framework for phishing URL detection, designed to address major challenges associated with high-dimensional feature spaces, model variance, and computational inefficiency in existing approaches. The proposed method combines principled feature reduction using principal component analysis with an ensemble of regularized MLP classifiers trained via bootstrap sampling. This design explicitly promotes learner diversity, reduces overfitting, and enhances generalization performance.

The comprehensive experimental results presented in Section IV on a public benchmark dataset, demonstrate that the BMLP framework consistently outperforms state-of-the-art models, including XGBoost, Random Forest, and a single MLP, with statistically significant improvements in classification accuracy and area under the ROC curve. In addition, the proposed approach achieves favourable computational efficiency, supporting real-time phishing detection with modest memory and training requirements. However, the evaluation is limited to a single dataset and does not explicitly account for long-term concept drift arising from the evolving nature of phishing attacks. Addressing cross-dataset generalization and adaptive learning

under concept drift therefore remains an important direction for future work. Also, the development of a well-documented Python package and a scalable REST API for integration into cybersecurity pipelines represents a direct and valuable avenue for future work which will facilitate practical adoption and benchmarking. While the current study focuses on binary classification, the framework can be extended to multi-class phishing scenarios and integrated into automated security systems for continuous threat monitoring.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Conflict of Interest

The author declares no conflict of interest.

References

[1] Kaur, K., & Jain, A. K. (2025). A Survey on Phishing Attack Taxonomy, Detection Techniques, Datasets, and Security Measures. Journal of Applied Security Research, 1-52.

[2] FBI IC3. (2024). Internet Crime Report 2024. Federal Bureau of Investigation. Retrieved from https://www.ic3.gov/AnnualReport/Reports/2024_IC3Report.pdf

[3] Sahingoz, O. K., Buber, E., Demir, O., & Diri, B. (2019). Machine learning-based phishing detection from URLs. Expert Systems with Applications, 117, 345-357.

[4] Alshingiti, Z., Alaqel, R., Al-Muhtadi, J., Haq, Q. E. U., Saleem, K., & Faheem, M. H. (2023). A deep learning-based phishing detection system using CNN, LSTM, and LSTM-CNN. Electronics, 12(1), 232.

[5] Jain, A., Khan, S., Koli, K., Taneja, H., Panwar, A., Alshammari, T., & Abdu, S. M. (2025). PhishNet 1.0: optuna-optimized stacking ensemble with Boruta-based feature selection for phishing URL detection. Scientific Reports.

[6] PhishTank. (2024). Phishing site trends and analysis. Retrieved from https://www.phishtank.org.

[7] Mittal, R., Singh, S. K., Kumar, S., Khullar, T., Kumar, R., Gupta, B. B., & Psannis, K. (2025). Advanced Techniques and Best Practices for Phishing Detection. In Critical Phishing Defense Strategies and Digital Asset Protection (pp. 149-186). IGI Global Scientific Publishing.

[8] Kustiawan, Y. A., & Ghauth, K. I. (2025). PhishOFE: A Novel Machine Learning Framework for Real-Time Phishing URL Detection with Optimized Feature Engineering. IEEE Access.

[9] AlSabah, M., Nabeel, M., Boshmaf, Y., & Choo, E. (2022). Content-agnostic detection of phishing domains using certificate transparency and passive dns. In Proceedings of the 25th International Symposium on Research in Attacks, Intrusions and Defenses (pp. 446-459).

[10] Dangwal, S., & Moldovan, A. N. (2021). Feature selection for machine learning-based phishing websites detection. In 2021 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA) (pp. 1-6). IEEE.

[11] Garapati, D. P., Maddipati, L. P., Swaroop, K. P., Samyuktha, B., Sowmya, G. H., & Valli, B. H. N. (2024). A comparative analysis of logistic regression, support vector machines, and random forest for phishing website identification. In 2024 International Conference on Computational Intelligence for Green and Sustainable Technologies (ICCIGST) (pp. 1-5). IEEE.

[12] Ahmed, D. S., Hussein, K. Q., & Allah, H. A. A. A. (2022). Phishing websites detection model based on decision tree algorithm and best feature selection method. Turkish Journal of Computer and Mathematics Education, 13(1), 100-107.

[13] Hajizada, A., & Jahan, S. (2023). Feature selections for phishing URLs detection using combination of multiple feature selection methods. In Proceedings of the 2023 15th International Conference on Machine Learning and Computing (pp. 444-450).

[14] Wei, Y., Nakayama, M., & Sekiya, Y. (2024). An Interpretable Fine-Tuned BERT Approach for Phishing URLs Detection: A Superior Alternative to Feature Engineering. In 2024 11th International Conference on Social Networks Analysis, Management and Security (SNAMS) (pp. 138-145). IEEE.

[15] Gupta, D., Gandotra, E., Dhalaria, M., & Gupta, N. (2025). An Ensemble Learning Approach for Detecting Phishing Websites Using an Entropy-Based Feature Selection Method. Journal of Information & Knowledge Management, 2550080.

[16] Tang, L., & Mahmoud, Q. H. (2021). A survey of machine learning-based solutions for phishing website detection. Machine Learning and Knowledge Extraction, 3(3), 672-694.

[17] Vrbančič, G., Fister Jr, I., & Podgorelec, V. (2020). Datasets for phishing websites detection. *Data in Brief*, 33, 106438.