Enhanced Phishing Website Detection Using Optimized Ensemble Stacking Models

Zainab Alamri[1], Abeer Alhuzali[2], Bassma Alsulami[3], Daniyal Alghazzawi[4]
Department of Computer Science-Faculty of Computing and Information Technology,
King Abdulaziz University, Jeddah, Saudi Arabia[1,2,3]
Information Systems Department-Faculty of Computing and Information Technology,
King Abdulaziz University, Jeddah, Saudi Arabia[4]

***Abstract*—Phishing attacks remain a persistent and evolving cybersecurity threat, necessitating the development of highly accurate and efficient detection mechanisms. This research introduces an optimized ensemble stacking framework for phishing website detection, leveraging advanced machine learning techniques, hybrid feature preprocessing, and meta-learning strategies. The proposed approach systematically evaluates nine diverse base classifiers: XGBoost, CatBoost, LightGBM, Random Forest, Gradient Boosting, Extra Trees, Support Vector Classifier, AdaBoost, and Bagging. We compare baseline classifiers, a standard ensemble stacking model, and four optimized stacking configurations across four balanced and imbalanced datasets. Our optimized ensemble stacking achieves perfect accuracy (one hundred percent) on the first two datasets, and over ninety-nine percent accuracy on the two more challenging imbalanced datasets. A direct comparison with related studies demonstrates that our optimized stacking approach delivers superior detection accuracy.**

***Keywords*—Phishing detection; machine learning; ensemble stacking; cybersecurity**

I. Introduction

The increasing reliance on online services has made individuals and organizations more vulnerable to cyberattacks, with phishing remaining a significant threat [1]. Phishing attacks employ deceptive techniques to steal sensitive information, often masquerading as legitimate entities to trick users into divulging credentials or financial details [2]. Despite existing security measures, phishing continues to evolve, posing persistent challenges to both individuals and organizations [1]. As of 2024, the rate of phishing attacks remains alarmingly high, with reports indicating that the number of phishing attacks detected worldwide ranges from hundreds of thousands to millions each month [3].

Traditional phishing detection systems often rely on machine learning algorithms and manually crafted features [1]. However, these systems struggle to keep pace with the constantly evolving tactics employed by phishers [4]. While various machine learning, deep learning, and other approaches have been proposed, their detection accuracy needs further improvement [5]. Ensemble learning, particularly stacking (stacked generalization), has emerged as a promising technique for enhancing the performance of classification models by combining the strengths of multiple base classifiers [6]. Stacking involves training a meta-classifier on the outputs of individual base classifiers, potentially leading to a more robust and accurate model [7]. In 2024, the proliferation of sophisticated phishing websites poses a significant threat to individuals and organizations, resulting in substantial financial losses and personal data breaches [1]. Despite numerous detection methods, including machine learning and deep learning techniques, the accuracy and generalizability of these approaches remain insufficient to effectively combat the evolving tactics of cybercriminals [4]. The dynamic nature of phishing attacks, characterized by the use of advanced evasion techniques and rapidly changing features, necessitates the development of more robust and adaptive detection systems. Traditional phishing detection systems that rely on machine learning and manual features struggle with evolving tactics [8].

Phishing website detection has become a critical area of research due to the increasing sophistication and frequency of phishing attacks targeting individuals, organizations, and even IoT environments. Machine learning (ML) and deep learning methods, particularly ensemble learning techniques such as stacking, bagging, and boosting, have shown promise in improving detection accuracy by leveraging the strengths of multiple classifiers. Recent studies have demonstrated that stacking ensemble models, especially when optimized, can outperform single classifiers and traditional detection methods in terms of accuracy, recall, and other performance metrics[9], [10]. Despite these advances, several limitations persist in current research. Many studies rely on a fixed set of base classifiers without systematically selecting or optimizing the most effective algorithms for stacking, which can potentially limit model performance [11], [12], [13]. Optimization efforts, when present, often focus on parameter tuning for individual models rather than holistic selection and combination of diverse, high-performing classifiers [5], [11], [14]. Additionally, some approaches lack robust validation across multiple, diverse datasets, raising concerns about the generalizability of their results to real-world scenarios [15], [9], and [5]. There is also a need for more comprehensive feature selection and integration strategies to further enhance detection capabilities [15], [14]. These gaps are significant because suboptimal model selection, insufficient optimization, and limited generalizability can result in lower detection accuracy and increased vulnerability to evolving phishing tactics. Inadequate detection systems may fail to protect users and organizations from financial loss, data breaches, and reputational damage, especially as attackers continually adapt their methods to bypass existing defenses [9], [16]. This study addresses the research question: How can optimized ensemble stacking improve phishing website detection accuracy across diverse datasets with varying sizes, balances, and feature complexities? To answer this, we systematically select and optimize a diverse set of strong base models for stacking, employ advanced optimization techniques, and validate the enhanced ensemble stacking approach across multiple datasets. By focusing on both algorithm selection and parameter optimization, the proposed method achieves higher detection accuracy, improved robustness, and better adaptability to new phishing strategies.

This study responds to these challenges by systematically selecting and optimizing a diverse set of strong base models for stacking, employing advanced optimization techniques, and validating the enhanced ensemble stacking approach across multiple datasets. By focusing on both algorithm selection and parameter optimization, the proposed method aims to achieve higher detection accuracy, improved robustness, and better adaptability to new phishing strategies, thereby strengthening real-world cybersecurity defenses. By extensively comparing various ensemble configurations, this research demonstrates the effectiveness of advanced ensemble stacking techniques in identifying phishing websites with high precision, reliability, and efficiency.

This study provides the following contributions:

- Develop an optimized ensemble stacking model with optimized hyperparameters, preprocessing, and feature engineering for performance improvement.
- Provide ensemble stacking variations for optimal results.
- Evaluate the model comprehensively across multiple datasets for performance assessment.

To achieve the above contributions, we develop an enhanced phishing website detection approach that leverages optimized ensemble stacking models to improve classification accuracy. This method integrates multiple machine learning classifiers within a structured ensemble framework, including Random Forest, Gradient Boosting, XGBoost, CatBoost, LightGBM, and Support Vector Classifier SVC, among others, capitalizing on their complementary strengths. A meta-classifier (Logistic Regression and CatBoost) is used to aggregate the outputs of the base models. Additionally, variations of stacking models are compared across four datasets to determine the optimal configurations. The model is tested on four datasets, each subjected to several variations of the ensemble stacking model to identify the most effective configuration.

By extensively comparing the four *optimized* stacking configurations, this research demonstrates the effectiveness of our advanced stacking techniques in identifying phishing websites with high precision, reliability, and efficiency. As a preview of our principal findings, Optimized Stacking-1 and -2 each achieve **100 %** accuracy on the two clean, balanced benchmarks, while all four variants maintain above **99 %** accuracy on the more challenging imbalanced datasets—underscoring both the robustness and practical deployability of our approach.

The rest of the paper is organized as follows. Section II highlights key related research studies. Sections III, IV, and V explain our methodology, evaluation metrics, and experimental results. Finally, we conclude in Section VII.

II. Related Work

This section reviews relevant research in two areas. First, we examine ensemble learning approaches, particularly stacking, which have shown effectiveness in phishing detection by combining multiple classifiers. Second, we explore optimization techniques that enhance stacking performance, including model selection, hyperparameter tuning, and meta-learner strategies.

A. Ensemble Learning for Phishing Detection

Ensemble methods, which combine multiple individual classifiers, have emerged as a powerful approach to enhance the accuracy and robustness of phishing detection systems. To overcome the limitations of individual classifiers, ensemble methods have been proposed, which combine the predictions of multiple models to improve overall accuracy and robustness [17].

Ensemble methods, while powerful, have certain limitations when applied to phishing detection. One significant limitation is the inability of some ensemble techniques, like random forests, to capture high correlations between features and their joint dependency on the label, which can affect the model's performance in complex datasets [18]. Additionally, traditional ensemble methods may struggle with the evolving nature of phishing attacks, as they often rely on static features that do not adapt well to new phishing tactics [19]. Furthermore, the computational cost and complexity of ensemble methods can be high, which may not be suitable for real-time detection scenarios [19].

Ensemble models typically require more resources for training and prediction, as they involve combining the outputs of multiple base classifiers. This increased computational overhead can be a drawback, especially in scenarios where real-time or low-latency detection is required. Additionally, the complexity of tuning and optimizing ensemble methods may pose challenges, as selecting the appropriate base classifiers and configuring the meta-classifier can significantly impact overall performance [20].

When comparing traditional ensemble methods and stacking ensembles, the latter often demonstrates superior performance in phishing detection tasks. For example, stacking models have achieved higher accuracy and F1 scores compared to standalone ensemble methods, such as random forests or AdaBoost [21]. The ability of ensemble stacking to combine the strengths of multiple algorithms and mitigate their weaknesses makes it a good choice for phishing detection [5], [9].

Stacking addresses these challenges by intelligently combining the strengths of diverse base classifiers, mitigating the risk of overfitting, and improving generalization performance, which can lead to more accurate and robust phishing website detection [17]

B. Optimization Techniques in Stacking Models

Phishing website detection has been a significant area of research due to the increasing sophistication of phishing attacks. Several studies have investigated the application of ensemble machine learning models to improve detection accuracy.

One approach involves the use of optimized stacking ensemble models, which combine multiple machine learning algorithms to improve detection performance. For instance, a study utilized a genetic algorithm to optimize parameters of ensemble methods, including Random Forest, AdaBoost, and XGBoost, achieving detection accuracies of 97.16%, 98.58%, and 97.39% across different datasets [5]. Another study reported a similar approach, achieving a detection accuracy of 97.16% by optimizing ensemble classifiers [22].

Other research has focused on stacking models that integrate multiple classifiers, such as Random Forest, Gradient Boosting, and AdaBoost, with logistic regression as an aggregator. This model achieved an accuracy of 98.72% and demonstrated superior performance compared to individual algorithms [10]. Additionally, a multilayer stacked ensemble learning model achieved accuracies ranging from 96.79% to 98.90% across various datasets, highlighting the effectiveness of layered ensemble techniques [9].

The accuracy of phishing website detection models varies across studies, with several achieving high performance. For example, one study reported an accuracy of 98.72% using a stack ensemble model [10], while another achieved 99.31% with a stacked classifier model employing six algorithms [3]. A different approach using a stacking model with URL and HTML features achieved 97.30% accuracy on one dataset and 98.60% on another [23].

Despite the high accuracy rates reported, there are limitations and gaps in current research. Many studies emphasize the need for further enhancement of detection accuracy and adaptability to evolving phishing tactics [22], [24]. Additionally, while ensemble models show promise, they often require complex optimization and feature selection processes, which can be computationally intensive [15], [25]. There is also a need for real-time detection capabilities, as many models are tested in controlled environments and may not perform as well in dynamic, real-world scenarios [23].

Another study optimized stacking ensemble methods using a Genetic Algorithm to tune parameters of various ensemble methods such as Random Forest, AdaBoost, XGBoost, and others. This approach achieved detection accuracies of 97.16% to 98.58% across different datasets, demonstrating significant improvements over traditional methods [22], [5].

Phishing website detection has been a significant area of research due to the increasing threat posed by phishing attacks. Various studies have explored the use of ensemble models to enhance detection accuracy. A stack ensemble model combining RandomForest, GradientBoosting, and AdaBoost with logistic regression as an aggregator achieved an accuracy of 98.72%, demonstrating superior performance over individual algorithms and existing studies [10]. Another study proposed an optimized stacking ensemble method using a genetic algorithm to tune parameters, achieving detection accuracies of 97.16%, 98.58%, and 97.39% across different datasets [5]. Similarly, a multilayer stacked ensemble learning model reported accuracies ranging from 96.79% to 98.90% across various datasets, outperforming baseline models [9].

Despite the high accuracy rates, several limitations and gaps remain in the current research. First, many models are tested on specific datasets, which may not generalize well to other datasets or real-world scenarios [5], [9]. Second, some models rely heavily on specific features, such as URL and HTML characteristics, which may not be present in all phishing websites [23]. Third, the complexity of ensemble models can lead to increased computational costs, making real-time detection challenging [15], [3]. Fourth, phishing tactics continue to evolve, necessitating ongoing updates and adaptations of detection models to maintain effectiveness [24], [21] . Last, the use of multiple models in stacking can increase the risk of overfitting, particularly if the base models are too complex or if the dataset is not sufficiently large or diverse [9], [5].

In summary, while ensemble models have significantly improved phishing website detection accuracy, challenges remain in optimizing these models for real-time applications and adapting to new phishing strategies. Our work addresses these limitations and designs a more robust and efficient detection system.

III. Methodology

This research employs a systematic and structured methodology to enhance the detection of phishing websites by implementing optimized ensemble stacking models. The proposed approach, aimed at improving phishing website detection, is uniformly applied across four benchmark datasets. It follows a systematic sequence involving data preprocessing, model training, and performance evaluation, and is consistently implemented on the following datasets: Dataset 1 [26], Dataset 2 [27],and Dataset 3 and Dataset 4, both derived from the same source [28]. We ensure consistency in base model selection (RandomForest, XGBoost, CatBoost, GradientBoosting, LightGBM, ExtraTrees, AdaBoost, Bagging, and SVC) across all datasets. However, ensemble stacking models are adjusted in hyperparameters, iterations, and cross-validation strategy for each dataset to balance execution time and accuracy. The methodology incorporates data preprocessing techniques, including feature encoding, SMOTE oversampling for imbalanced datasets, and feature standardization. The evaluation framework uses accuracy, classification reports, and confusion matrices to compare performance across datasets and determine the most effective dataset for phishing detection.

The proposed framework involves several steps, which are summarized below:

- Perform necessary preprocessing steps, such as encoding categorical variables, scaling numerical features, and addressing class imbalances using SMOTE on the selected datasets.
- Train and evaluate multiple individual classifiers, including RandomForest, XGBoost, CatBoost, LightGBM, GradientBoosting, ExtraTrees, SVC, AdaBoost, and Bagging, for each dataset. This establishes a baseline performance for comparison before implementing stacking ensembles. The results include accuracy scores and classification reports for all datasets.
- The structure of the stacking ensemble involves training a traditional stacking ensemble model using a variety of base classifiers, including XGBoost, Random Forest, CatBoost, LightGBM, Gradient Boost-

ing, Extra Trees, SVC, AdaBoost, and BaggingClassifier, with Logistic Regression as the final estimator. The improved stacking ensemble model incorporates optimized hyperparameters, balancing techniques (SMOTE), and boosting-based feature selection implicitly through models like XGBoost, CatBoost, and LightGBM. The improved version uses CatBoost as the final estimator and employs StratifiedKFold cross-validation to enhance generalization.

- Evaluate and compare the performance of individual base models, traditional stacking ensemble, and improved stacking ensemble across all datasets. Metrics such as accuracy, precision, recall, F1-score, and confusion matrices will be used for comparison.
- Illustrate the performance across models and datasets using various visualization methods. These visual aids help identify trends, strengths, and weaknesses in the models for each dataset.

The primary objective of this study is to investigate how variations in dataset characteristics impact the performance of the improved ensemble stacking model and to identify which dataset yields the most effective results. The overall workflow of the proposed approach is illustrated in Fig. 1.

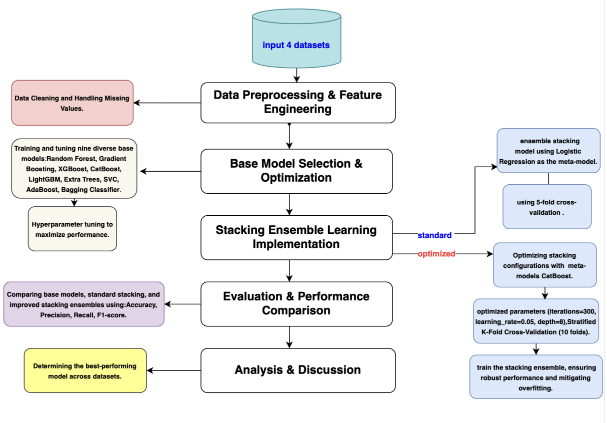
input 4 datasets
Data Preprocessing & Feature Engineering
Data Cleaning and Handling Missing Values.
Base Model Selection & Optimization
Training and tuning nine diverse base models:Random Forest, Gradient Boosting, XGBoost, CatBoost, LightGBM, Extra Trees, SVC, AdaBoost, Bagging Classifier.
Hyperparameter tuning to maximize performance.
Stacking Ensemble Learning Implementation
standard
ensemble stacking model using Logistic Regression as the meta-model.
using 5-fold cross-validation .
optimized
Optimizing stacking configurations with meta-models CatBoost.
optimized parameters (iterations=300, learning_rate=0.05, depth=8),Stratified K-Fold Cross-Validation (10 folds).
train the stacking ensemble, ensuring robust performance and mitigating overfitting.
Evaluation & Performance Comparison
Comparing base models, standard stacking, and improved stacking ensembles using:Accuracy, Precision, Recall, F1-score.
Analysis & Discussion
Determining the best-performing model across datasets.

Fig. 1. Overview of the proposed ensemble stacking workflow.

A. Dataset Description

This study utilizes four datasets of varying sizes, features, and class distributions to train, test, and evaluate the phishing website detection model. These datasets ensure a comprehensive assessment of model performance under different conditions. Balanced datasets facilitate the assessment of baseline performance without introducing bias, thereby providing a clear foundation for evaluating model accuracy. In contrast, unbalanced datasets are used to test the robustness of the models, with SMOTE applied to generate synthetic samples and improve classification fairness. This combination of balanced and unbalanced datasets enabled a comprehensive evaluation of the models' robustness and their practical applicability in real-world phishing detection scenarios.

1) Balanced datasets: include Dataset 1 [26] (11,430 instances, 89 features), titled Web Page Phishing Detection and sourced from Mendeley Data, and Dataset 2 [27] (11,481 instances, 89 features), also titled Web Page Phishing Detection and obtained from Kaggle. Both datasets contain an equal number of phishing and legitimate website samples. Specifically, Dataset 1 includes 5,715 phishing and 5,715 legitimate samples, while Dataset 2 consists of 5,740 phishing and 5,741 legitimate samples. These balanced distributions eliminate class bias and support reliable model evaluation without the need for resampling techniques. Both datasets feature a rich set of attributes relevant to phishing detection, such as length_url, nb_dots, domain_age, iframe, tld_in_path, google_index, and ratio_digits_host. These features capture URL structure, domain trustworthiness, and content behavior, which are critical for distinguishing phishing from legitimate websites.

2) Unbalanced datasets: include Dataset 3 [28] (88,647 instances, 112 features) and Dataset 4 [28] (58,645 instances, 112 features), both derived from the Phishing Websites Dataset available on Mendeley Data, both of which exhibit varying degrees of class imbalance. Dataset 3 contains 30,648 phishing and 57,999 legitimate samples (34.57% phishing,65.43% legitimate), while Dataset 4 includes 30,651 phishing and 27,994 legitimate samples (52.26% phishing, 47.74% legitimate). To address this imbalance, the Synthetic Minority Oversampling Technique (SMOTE) was applied. These datasets include more advanced and diverse features such as ttl_hostname,qty_redirects, url_shortened, tls_ssl_certificate, and domain_google_index, which reflect redirection patterns, SSL usage, and DNS behaviors.

B. Data Preprocessing

To ensure consistency and accuracy, a structured preprocessing pipeline was applied across all datasets. First, non-numeric columns were identified, and missing values in categorical features were replaced with the string ”Missing” to retain the information. Second, LabelEncoder was used to convert categorical features and the target variable into a numerical format for compatibility with machine learning models. Third, SMOTE was applied with an 80% sampling strategy to balance minority and majority classes, improving model generalization. Each dataset was split into 80% training and 20% testing, ensuring fair evaluation on unseen data. Lastly, StandardScaler transformed the features to have a mean of 0 and a standard deviation of 1, benefiting models that are sensitive to feature magnitude. All four datasets underwent identical preprocessing steps to maintain fairness in performance comparisons. By applying these steps, the data was cleaned, balanced, and optimized, ensuring reliable training and evaluation of the phishing detection models.

C. Selected Base Models and Ensemble Learning Design

To ensure robust and generalizable phishing detection, a diverse set of nine base classifiers was selected and consistently applied across four benchmark datasets: Dataset 1, Dataset 2, Dataset 3, and Dataset 4. These models were chosen to leverage complementary strengths in handling non-linear patterns, categorical features, boosting mechanisms, and high-dimensional data.

The base models used in all experiments include gradient boosting algorithms (XGBoost, CatBoost, LightGBM, GradientBoostingClassifier), ensemble tree methods (RandomForestClassifier, ExtraTreesClassifier), kernel-based classifiers (SVC), and ensemble meta-learners such as AdaBoostClassifier and BaggingClassifier. Each model is known for its strong performance in phishing detection and tabular classification problems [29], [30], [31], [8]. The gradient boosting algorithms are known for their high accuracy and ability to capture complex relationships [8]. XGBoost is also efficient and scalable. CatBoost is particularly good at handling categorical features. The ensemble tree-based models can handle complex, nonlinear feature interactions well and have demonstrated high accuracy in phishing detection tasks [29], [30], [32] . Support Vector Classifier (SVC) with nonlinear kernels captures subtle lexical and structural patterns in URLs, contributing to high classification accuracy and generalization across datasets [30], [31]. AdaBoost enhances the performance of weak learners by focusing on complex samples, thereby improving overall detection accuracy [30]. Bagging reduces variance by aggregating multiple models trained on different data samples, improving robustness and generalization [31].

Comparative studies highlight the effectiveness of combining diverse classifiers, including SVC, AdaBoost, and Bagging, in the phishing detection task [30]. These diverse machine learning models, including ensemble tree-based methods, gradient boosting algorithms, kernel-based classifiers, and ensemble stacking techniques that reduce variance or improve weak learners, have been effectively used together to detect phishing websites with high accuracy [33], [19], [4].

1) Training and evaluation of models: To systematically assess model performance, a dedicated function was designed to train and evaluate multiple classifiers on the phishing datasets. This function accepts training and testing subsets for each dataset and outputs standardized evaluation metrics for comparative analysis. The procedure involves training each classifier on the training data, generating predictions on the test data, and evaluating performance using key classification metrics. Each model is evaluated on all four datasets to examine its generalizability across varying data characteristics. By applying this process uniformly across balanced and unbalanced datasets, it was possible to identify base models that consistently deliver high performance.

Specifically:

- Models were trained and evaluated on Dataset 1 and Dataset 2, both of which are balanced.
- The same process was applied to Dataset 3 and Dataset 4, which are imbalanced and larger in scale.

For each dataset, model results were stored and analyzed using:

- Accuracy: Overall classification accuracy for each model.
- Classification Report: Detailed metrics including precision, recall, and F1-score per class.

D. Standard Ensemble Stacking Approach

The ensemble stacking approach integrates the predictive capabilities of multiple diverse base classifiers into a unified model, aiming to significantly enhance phishing website detection accuracy. This layered architecture consists of a base layer where classifiers are trained independently, and a meta-layer that learns to optimally combine their predictions into a final decision. The effectiveness of Logistic Regression as a meta-learner in stacking ensembles has been demonstrated in phishing detection systems, achieving 98% accuracy when combined with base classifiers like Random Forest and XGBoost [34]. To ensure generalization and reduce overfitting, 5-fold cross-validation was employed during model training, as recommended in recent phishing detection research [35].

In this study, a consistent standard stacking framework was applied across our four datasets, including the two balanced datasets. This structural consistency enables a fair comparison and allows analysis of how dataset characteristics affect ensemble performance. The base layer comprised nine machine learning classifiers selected for their proven effectiveness and complementary strengths: RandomForest, Gradient Boosting, XGBoost, CatBoost, LightGBM, ExtraTrees, SVC, AdaBoost, and Bagging. Each classifier was chosen to contribute diverse perspectives in identifying phishing behavior patterns. For this stacking configuration, Logistic Regression was used as the meta-learner to combine the outputs of the base classifiers. To ensure generalization and reduce overfitting, 5-fold cross-validation was employed during model training.

E. Optimized Ensemble Stacking Approach

To improve the accuracy and efficiency of phishing website detection, we conduct four variations to optimize the ensemble stacking models. These models are systematically refined to strike a balance between complexity, computational efficiency, and detection performance.

- Optimized Stacking-1: comprehensive stacking model. This version employs a comprehensive ensemble stacking approach that integrates the nine base classifiers, including base classifiers RandomForest, XGBoost, CatBoost, LightGBM, Gradient Boosting, Extra Trees, SVC, AdaBoost, and Bagging. The CatBoost classifier was utilized as the meta-model, with hyperparameter tuning applied to optimize performance.
- Optimized Stacking-2: performance-oriented selection. This architecture prioritizes high-performing models by selecting XGBoost, CatBoost, and RandomForest as base classifiers, thereby reducing computational complexity while maintaining strong classification capabilities. The CatBoost classifier is used as the meta-model.
- Optimized Stacking-3: efficiency-optimized stacking. To enhance computational efficiency, the number of base models is reduced, retaining only the most effective classifiers: XGBoost, CatBoost, RandomForest, LightGBM, Gradient Boosting, and SVC. The CatBoost classifier continued to serve as the meta-model, ensuring robustness in predictions.
- Optimized Stacking-4: minimalist high-performance model. Designed for optimized execution and minimal computational cost, this model leverages a CatBoost classifier with advanced hyperparameter tuning as

the meta-model. The base models are reduced to XGBoost, CatBoost, RandomForest, LightGBM, and Gradient Boosting, achieving high efficiency while preserving detection accuracy. This ensemble model achieves a balance between execution time and detection accuracy.

1) Design and configuration of the optimized ensemble stacking: Each stacking configuration followed a unified pipeline. Non-numeric features are encoded using LabelEncoder, and class imbalance is addressed using SMOTE. Feature scaling is performed using StandardScaler to ensure consistency across numerical values. Stratified K-Fold cross-validation is used for training: 10-fold for balanced or smaller datasets (Datasets 1, 2, and 4), and 5-fold for the larger dataset (Dataset 3) to reduce execution time. A diverse set of nine base classifiers is employed to enhance generalization and capture a broad range of learning patterns: tree-based models (RandomForestClassifier, ExtraTreesClassifier, and GradientBoostingClassifier), boosting techniques (XGBClassifier, CatBoostClassifier, LGBMClassifier, and AdaBoostClassifier), a kernel-based model (SVC), and a bagging-based method (BaggingClassifier).

2) Base model optimization: The same base model structure is applied across all datasets, but hyperparameters are adjusted to match the dataset size and complexity. For the balanced datasets (Datasets 1 and 2), base models are configured with higher complexity. For the large, imbalanced datasets (Datasets 3 and 4), parameter values are reduced to accelerate training. Specifically, we perform the following key adjustments:

- Number of Estimators: 500 estimators for Datasets 1 and 2; reduced to 100 or 50 for Datasets 3 and 4.
- Learning Rate: It is set to 0.05 for smaller datasets; increased to 0.1 for larger datasets.
- Depth: Models used greater depth (10–25) for balanced data; shallower depth (6–10) for large-scale datasets in Optimized Stacking-1 and Stacking-3.

It is important to note how hyperparameter settings influenced performance across datasets. Increasing the number of estimators (e.g., 500 on Datasets 1 and 2) improved accuracy and reduced variance, but also increased training time, which is why smaller values (100 or 50) were used for the larger datasets. Similarly, greater tree depth (10–25) enhanced recall by capturing complex phishing patterns, but shallower depths (6–10) were more efficient for large-scale datasets and reduced the risk of overfitting. Adjusting the learning rate also played a critical role: lower values (0.05) improved model stability and recall on imbalanced data, while higher values (0.1) accelerated convergence but risked losing fine-grained detection capability. For the CatBoost meta-model, more iterations (300) strengthened performance on smaller balanced datasets, whereas fewer iterations (50–100) were sufficient for larger datasets to maintain efficiency without significant accuracy loss. These observations confirm that hyperparameter tuning was essential not only for maximizing accuracy but also for balancing robustness and computational efficiency across datasets.

3) Meta-model optimization: The meta-model used in all stacking ensembles is `CatBoostClassifier`, replacing Logistic Regression to improve generalization and robustness. The CatBoostClassifier demonstrates strong performance due to its efficient handling of categorical features and robustness on imbalanced datasets, which are commonly encountered in phishing detection scenarios [25]. Configurations are tailored per dataset:

- Balanced datasets 1 and 2: 300 iterations, depth 8, learning rate 0.05, with 10-fold cross-validation.
- Large and imbalanced Dataset 3:
  - Optimized Stacking-1 and Optimized Stacking-3: 50 iterations, depth 6, learning rate 0.1, with 5-fold CV.
  - Optimized Stacking-2 and Optimized Stacking-4: 300 iterations, depth 8, learning rate 0.05, with 10-fold CV.
- Dataset 4:
  - Optimized Stacking-1: 100 iterations, depth 6, learning rate 0.1, with 10-fold CV.
  - Optimized Stacking-2, Optimized Stacking-3, Optimized Stacking-4: 300 iterations, depth 6, learning rate 0.1, with 10-fold CV.

IV. Models Evaluation

A. Experimental Setup

To evaluate the effectiveness of the proposed optimized ensemble stacking models for phishing website detection, all experiments were executed in Google Colab, a cloud-based Jupyter environment. The programming environment was configured with Python 3 and accelerated using the v5e-1 TPU to handle the computational demands of training complex ensemble models on large datasets. We utilized several libraries such as scikit-learn [36], XGBoost [37], LightGBM [38], CatBoost [39], Matplotlib [40], Seaborn [41], Pandas [42], and NumPy [43].

B. Evaluation Metrics

To thoroughly evaluate the performance of the proposed *optimized ensemble stacking model* for phishing website detection, multiple classification metrics were employed. These metrics provide a comprehensive view of each model's effectiveness, especially in distinguishing between legitimate and phishing websites across datasets with varying distributions.

- Accuracy: Represents the overall proportion of correctly classified instances:

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

  However, while accuracy is informative in balanced datasets, it may be misleading in imbalanced scenarios.

- Precision: Measures the proportion of true phishing detections among all predicted phishing cases:

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

High precision reduces false alarms and is essential for protecting legitimate websites from misclassification.

- Recall (Sensitivity): Evaluates the model's capability to correctly detect actual phishing sites:

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

High recall ensures phishing threats are not overlooked.

- F1-Score: The harmonic mean of precision and recall, providing a single balanced metric:

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

It is especially useful in imbalanced data situations.

- AUC-ROC: Measures the area under the Receiver Operating Characteristic curve, reflecting the model's ability to distinguish between classes independently of the classification threshold. A value close to 1.0 indicates excellent performance.

- Confusion Matrix: Summarizes true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN), offering detailed insights into classification errors.

- Execution Time: Records the time required for model training and evaluation. This is crucial for determining the feasibility of deploying the model in real-time phishing detection systems.

V. Experiments and Results

To evaluate the proposed optimized ensemble stacking models, experiments were conducted on four phishing datasets differing in size, class balance, and feature structure. The same stacking architecture and base models were applied across all datasets, including four optimized stacking (optimized-stacking1 to optimized-stacking4) to ensure consistent comparison. Preprocessing involved label encoding, standard scaling, and SMOTE for imbalanced datasets. For large datasets, model parameters were adjusted to reduce training time without sacrificing accuracy. Evaluation used an 80:20 split and stratified cross-validation, measuring accuracy, precision, recall, F1-score, and confusion matrices to assess performance under different data conditions.

A. Experiment 1: Evaluation of Base Models Individually

Before implementing the stacking ensembles, nine base classifiers were independently evaluated to assess their generalizability and standalone performance across four phishing datasets: Dataset 1, Dataset 2, Dataset 3, and Dataset 4. These models include Random Forest, Gradient Boosting, XGBoost, CatBoost, LightGBM, Extra Trees, Support Vector Classifier (SVC), AdaBoost, and Bagging. Each model was selected based on its distinct learning paradigm and complementary strengths.

All classifiers were trained using an identical experimental pipeline that included preprocessing steps such as categorical encoding, SMOTE for handling class imbalance (in unbalanced datasets), and feature scaling. A fixed random seed was used for reproducibility, and models were evaluated on an 80/20 train-test split. Performance was measured using standard classification metrics including accuracy, precision, recall, F1-score, and AUC. The detailed performance metrics for all base classifiers are presented in Tables I through IV, with corresponding ROC curves illustrated in Fig. 2 to 5.

TABLE I. Base Models' Performance on Dataset 1

Model	Precision	Recall	F1-Score	Accuracy (%)	AUC
Random Forest	0.9695	0.9693	0.9694	96.94	0.9945
Gradient Boosting	0.9593	0.9593	0.9593	95.93	0.9919
XGBoost	0.9733	0.9733	0.9733	97.33	0.9956
CatBoost	0.9729	0.9728	0.9729	97.29	0.9959
LightGBM	0.9720	0.9720	0.9720	97.20	0.9958
Extra Trees	0.9696	0.9692	0.9694	96.94	0.9952
SVC	0.9628	0.9628	0.9628	96.28	0.9930
AdaBoost	0.9536	0.9536	0.9536	95.36	0.9884
Bagging	0.9554	0.9553	0.9554	95.54	0.9873

TABLE II. Base Models' Performance on Dataset 2

Model	Precision	Recall	F1-Score	Accuracy (%)	AUC
RandomForest	0.9808	0.9809	0.9808	98.08	0.9966
GradientBoosting	0.9599	0.9600	0.9599	95.99	0.9915
XGBoost	0.9856	0.9856	0.9856	98.56	0.9976
CatBoost	0.9856	0.9856	0.9856	98.56	0.9972
LightGBM	0.9835	0.9834	0.9835	98.35	0.9970
ExtraTrees	0.9839	0.9839	0.9839	98.39	0.9979
SVC	0.9665	0.9665	0.9665	96.65	0.9921
AdaBoost	0.9473	0.9474	0.9473	94.73	0.9873
Bagging	0.9691	0.9691	0.9691	96.91	0.9904

TABLE III. Base Models' Performance on Dataset 3

Model	Precision	Recall	F1-Score	Accuracy (%)	AUC
RandomForest	0.9656	0.9676	0.9666	96.98	0.9951
GradientBoosting	0.9475	0.9497	0.9486	95.34	0.9893
XGBoost	0.9654	0.9662	0.9658	96.90	0.9951
CatBoost	0.9662	0.9679	0.9671	97.02	0.9952
LightGBM	0.9611	0.9634	0.9622	96.58	0.9944
ExtraTrees	0.9644	0.9664	0.9654	96.86	0.9945
SVC	0.9361	0.9400	0.9380	94.37	0.9829
AdaBoost	0.9300	0.9300	0.9300	93.67	0.9839
Bagging	0.9624	0.9629	0.9626	96.62	0.9891

TABLE IV. Base Models' Performance on Dataset 4

Model	Precision	Recall	F1-Score	Accuracy (%)	AUC
RandomForest	0.9581	0.9578	0.9580	95.81	0.9915
GradientBoosting	0.9303	0.9299	0.9301	93.03	0.9805
XGBoost	0.9574	0.9574	0.9574	95.75	0.9914
CatBoost	0.9575	0.9574	0.9574	95.75	0.9914
LightGBM	0.9510	0.9508	0.9509	95.10	0.9895
ExtraTrees	0.9567	0.9565	0.9566	95.67	0.9913
SVC	0.9139	0.9130	0.9134	91.36	0.9723
AdaBoost	0.9116	0.9118	0.9117	91.18	0.9707
Bagging	0.9521	0.9522	0.9522	95.23	0.9836

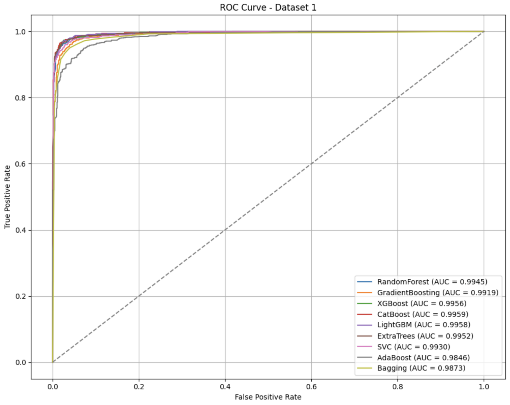

Fig. 2. ROC Curve of base model - Dataset 1.

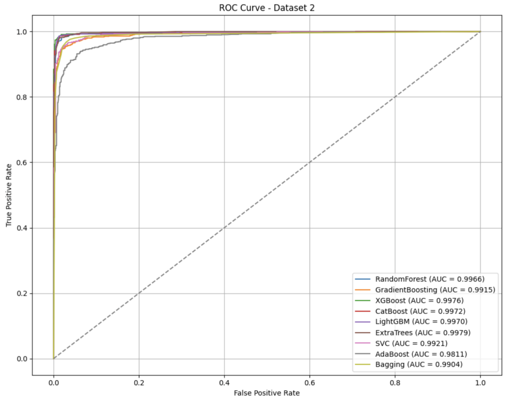

Fig. 3. ROC Curve of base model - Dataset 2.

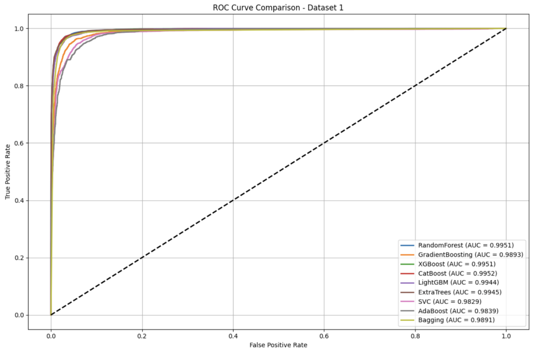

Fig. 4. ROC Curve of base model - Dataset 3.

The results showed that XGBoost and CatBoost consistently outperformed other models across all datasets, especially on the balanced ones, where both achieved perfect or near-perfect scores. Conversely, models like AdaBoost and Gradient Boosting showed performance degradation on imbalanced datasets due to sensitivity to class distribution. Tree-based ensembles such as Random Forest, Extra Trees, and Bagging demonstrated strong stability across all data conditions. SVC generally underperformed, possibly due to its limited scalability in high-dimensional spaces. These findings informed the selection of models used in subsequent ensemble stacking experiments.

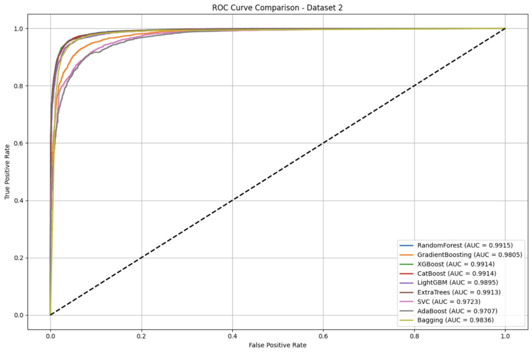

Fig. 5. ROC Curve of base model - Dataset 4.

B. Experiment 2: Performance of Standard Ensemble Stacking Model

This experiment implements a standard ensemble stacking architecture applied consistently across all four datasets. The stacking ensemble integrates tree-based models, including Random Forest and Extra Trees, which are known for handling high-dimensional data and reducing variance, alongside the Bagging classifier to enhance robustness against overfitting. It also incorporates boosting-based classifiers such as Gradient Boosting, XGBoost, LightGBM, and CatBoost, which iteratively improve model accuracy by focusing on difficult samples. Linear and kernel-based models, such as Support Vector Classifier (SVC), are included to capture non-linear patterns, while AdaBoost provides adaptive boosting of weak learners. Logistic Regression is used as the final meta-estimator due to its computational efficiency and ability to aggregate diverse base model outputs effectively.

The same ensemble architecture and base model configuration were used across all datasets to ensure fair comparison. The results, shown in Table V, demonstrate the model's high classification capability across both balanced and imbalanced data distributions. The stacking ensemble achieved high accuracy and consistently strong performance across all datasets. Particularly, Datasets 1 and 2, both balanced and clean, yielded the highest results, with Dataset 2 reaching 98.65% accuracy and an AUC of 0.9983. Dataset 3 and Dataset 4, exhibited slightly lower but still competitive performance. The stacking ensemble demonstrates superior capability in both balanced and imbalanced conditions, confirming its robustness. The high AUC values and consistent classification metrics validate the effectiveness of using heterogeneous learners within a stacking framework. These results justify the ensemble's role as a strong foundation for subsequent optimized stacking improvements.

TABLE V. PERFORMANCE OF STANDARD STACKING ENSEMBLE ACROSS ALL DATASETS

Dataset	Precision	Recall	F1-Score	Accuracy	AUC
Dataset 1	0.97	0.97	0.97	97.38%	0.9962
Dataset 2	0.99	0.99	0.99	98.65%	0.9983
Dataset 3	0.97	0.97	0.97	97.42%	0.9960
Dataset 4	0.96	0.96	0.96	96.34%	0.9935

Fig. 6 and 7 show the ROC curves for Datasets 1 and 2, indicating an ideal classification boundary with near-perfect separation between classes. Fig. 8 and Fig. 9 illustrate the ROC curves for Datasets 3 and 4.

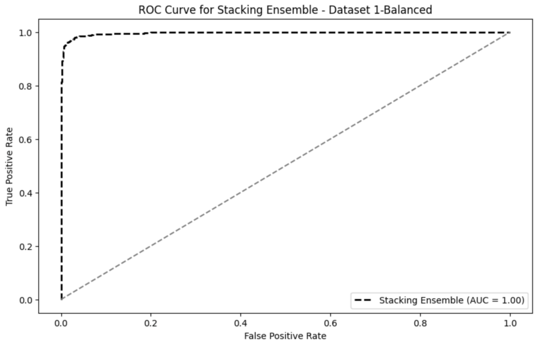

Fig. 6. ROC curve – standard ensemble stacking (Dataset 1).

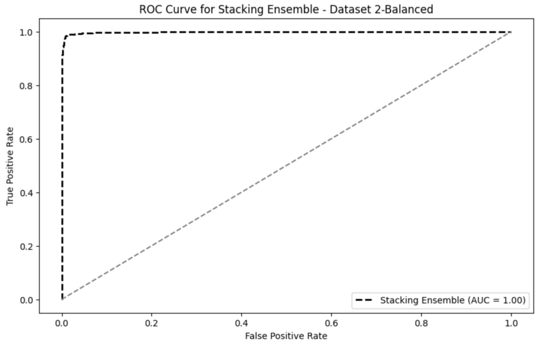

Fig. 7. ROC curve – standard ensemble stacking (Dataset 2).

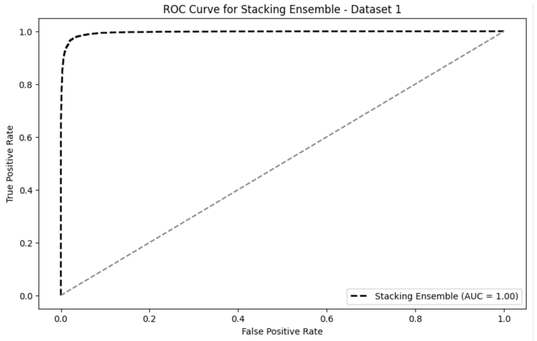

Fig. 8. ROC curve – standard ensemble stacking (Dataset 3).

C. Experiment 3: Optimized Ensemble Stacking Performance

To further enhance classification accuracy and address the limitations of the standard stacking approach, four optimized stacking configurations (Optimized Stacking-1 through Optimized Stacking-4)are implemented and consistently evaluated across four phishing detection datasets. All optimized stacking configurations retained the same architectural structure across all datasets to ensure a fair and consistent comparison. The base models and meta-model remained unchanged, with only hyperparameter adjustments made to accommodate computational constraints, particularly for the larger datasets.

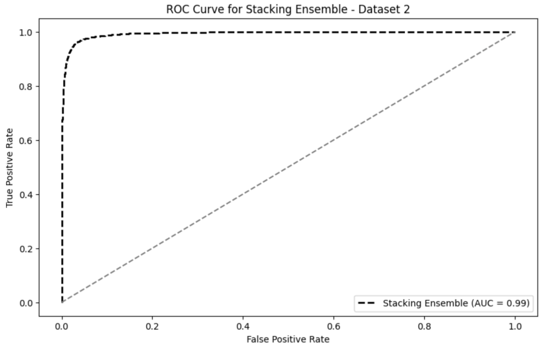

Fig. 9. ROC curve – standard ensemble stacking (Dataset 4).

TABLE VI. Performance of Optimized Stacking Ensembles on Dataset 1

Model	Accuracy (%)	Precision	Recall	F1-Score
optimized Stacking-1	100.00	1.00	1.00	1.00
optimized Stacking-2	99.91	1.00	1.00	1.00
optimized Stacking-3	99.78	1.00	1.00	1.00
optimized Stacking-4	99.96	1.00	1.00	1.00

TABLE VII. Performance of Optimized Stacking Ensembles on Dataset 2

Model	Accuracy (%)	Precision	Recall	F1-Score
optimized Stacking-1	100.00	1.00	1.00	1.00
optimized Stacking-2	100.00	1.00	1.00	1.00
optimized Stacking-3	100.00	1.00	1.00	1.00
optimized Stacking-4	100.00	1.00	1.00	1.00

TABLE VIII. Performance of Optimized Stacking Ensembles on Dataset 3

Model	Accuracy (%)	Precision	Recall	F1-Score
optimized Stacking-1	99.30	0.99	0.99	0.99
optimized Stacking-2	99.71	1.00	1.00	1.00
optimized Stacking-3	99.24	0.99	0.99	0.99
optimized Stacking-4	99.44	0.99	0.99	0.99

TABLE IX. Performance of Optimized Stacking Ensembles on Dataset 4

Model	Accuracy (%)	Precision	Recall	F1-Score
optimized Stacking-1	99.27	0.99	0.99	0.99
optimized Stacking-2	99.26	0.99	0.99	0.99
optimized Stacking-3	98.99	0.99	0.99	0.99
optimized Stacking-4	99.09	0.99	0.99	0.99

TABLE X. Cross-Dataset Accuracy Comparison of Improved Stacking Ensembles

Model	Dataset 1	Dataset 2	Dataset 3	Dataset 4
optimized Stacking-1	100.00	100.00	99.30	99.27
optimized Stacking-2	99.91	100.00	99.71	99.26
optimized Stacking-3	99.78	100.00	99.24	98.99
optimized Stacking-4	99.96	100.00	99.44	99.09

Tables VI, VII, VIII, and IX demonstrate that all four optimized stacking ensembles consistently achieved high classification performance across diverse datasets, regardless of balance or size. Optimized Stacking-1, which included a comprehensive set of base classifiers such as gradient boosting, bagging, SVM, and decision trees, reached 100% accuracy on Dataset 1 and Dataset 2 and maintained strong generalization with accuracies of 99.30% and 99.27% on Dataset 3 and Dataset 4, respectively. Optimized Stacking-2 adopted a more streamlined architecture, utilizing only CatBoost, XGBoost, and Random Forest as base models. This approach reduced computational complexity while preserving accuracy, achieving 100% on balanced datasets and the highest recorded accuracy (99.71%) on the large-scale Dataset 3. Optimized Stacking-3 focused on execution efficiency by reducing the ensemble size based on performance impact. While slightly lower in accuracy compared to Ensembles 1 and 2, it still achieved competitive results—99.78% on Dataset 1 and 98.99% on Dataset 4—demonstrating that minimal model diversity can still yield strong outcomes when well-selected. Optimized Stacking-4 further simplified the ensemble configuration to prioritize speed and resource efficiency. Despite this reduction in complexity, it achieved excellent accuracy levels across all datasets, peaking at 99.96% on Dataset 1 and attaining 99.44% on Dataset 3, which confirms its effectiveness for real-time or resource-constrained environments.

As shown in the cross-dataset comparison in Table X, each optimized stacking variant maintained stability and high precision across varied data characteristics. These results collectively validate that the proposed optimized ensemble stacking configurations are not only robust and adaptable but also scalable to different operational constraints and dataset profiles. This makes them highly suitable for deployment in real-world phishing detection systems, where both accuracy and computational efficiency are critical.

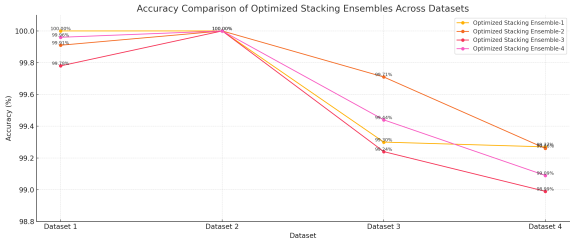

Fig. 10. Comparison of all dataset performance of optimized stacking.

Fig. 11 to Fig. 14 show a comparison of the accuracy of base models,Standard Ensemble Stacking, and improved stacks across datasets.

In addition, Fig. 15 to Fig. 30 illustrate the true and false positives and negatives for the best-performing model on each dataset.

VI. Discussion

We enhance phishing website detection by developing optimized ensemble stacking models and evaluating their performance across four distinct datasets. The Optimized ensemble stacking models outperformed both standard stacking and individual base models due to several key enhancements. First, they utilized optimized feature preprocessing techniques, including SMOTE for class balancing, feature scaling, and label encoding, which allowed for a more accurate representation of phishing patterns. Second, replacing the traditional Logistic Regression meta-model with the more robust CatBoost classifier improved the models' ability to generalize and handle nonlinear relationships and class imbalance. Additionally, reordering and selecting diverse base models enhanced the ensemble's ability to capture complementary learning patterns, resulting in stronger predictions. These improvements also ensured scalability, allowing the models to perform consistently across both small balanced datasets and large imbalanced ones. Finally, the enhanced variants demonstrated significant gains in execution efficiency, maintaining high accuracy while reducing computational time and complexity.

The comparative performance of the optimized ensemble stacking models varies across the four benchmark datasets due to inherent differences in their characteristics. Dataset 1 and Dataset 2 are balanced and relatively clean, with a near 50:50 distribution of phishing and legitimate instances. This balance enables the models to achieve perfect or near-perfect accuracy (approaching 100%), as the absence of significant class imbalance allows the learning process to capture phishing patterns more effectively.

In contrast, Dataset 3 is considerably larger (88,647 instances) and initially imbalanced, necessitating the use of SMOTE to address skewness. Even after balancing, the dataset's scale and the complexity of its features introduce additional variance, which explains the slight reduction in accuracy compared to the smaller, balanced datasets. Similarly, Dataset 4 presents a more challenging structure, with closer ratios between phishing and legitimate cases combined with a higher diversity of complex URL-based features. These factors contribute to minor fluctuations in the achieved results, even when using optimized stacking configurations.

Overall, these findings demonstrate that dataset-specific characteristics—such as class balance, dataset size, and feature richness—play a critical role in shaping model performance. At the same time, the consistently high accuracy achieved across all four datasets highlights the robustness and adaptability of the optimized ensemble stacking framework, which effectively generalizes across both balanced and imbalanced datasets while efficiently handling diverse feature distributions.

A. Dataset Characteristics and their Impact on Model Design

Each of the four datasets used Dataset 1, Dataset 2, Dataset 3, and Dataset 4 presented unique characteristics in terms of size, feature composition, and class distribution, all of which significantly influenced model design choices and performance outcomes. Dataset 1 and Dataset 2 were balanced, with a 50-50 split between legitimate and phishing samples, and exhibited high-quality data with no missing values. Their balanced nature eliminated the need for resampling techniques and allowed direct application of stacking models. Dataset 3 and Dataset 4 were imbalanced, requiring the use of SMOTE to address class distribution skewness. Dataset 3 was particularly large (88,647 records), demanding adjustments to model complexity and training time. Key dataset features influencing stacking performance included URL length, number of special characters, domain-related attributes, redirection behavior, and SSL

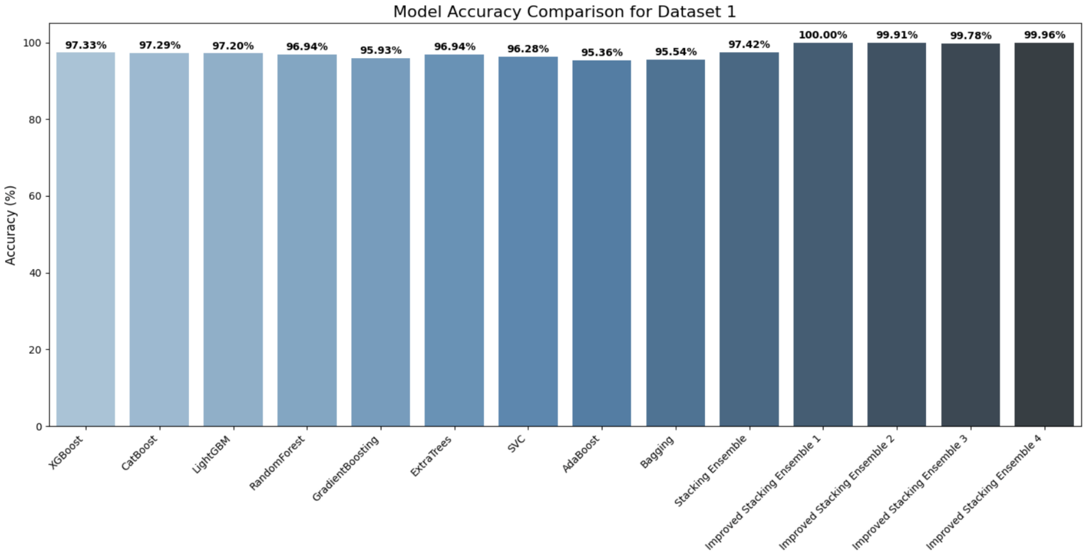

Fig. 11. Accuracy comparison on Dataset 1 for all models.

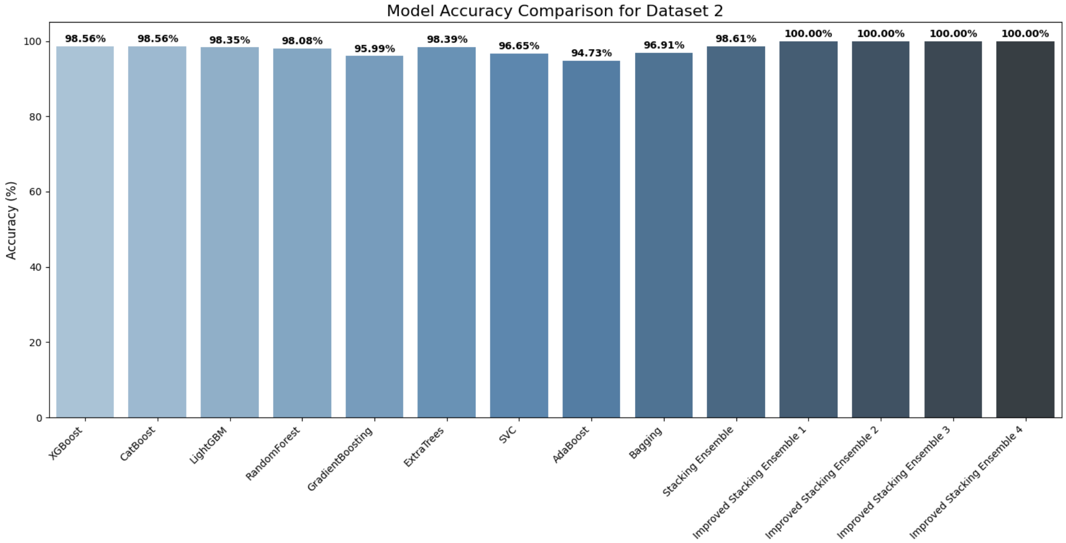

Fig. 12. Accuracy comparison on Dataset 2 for all models.

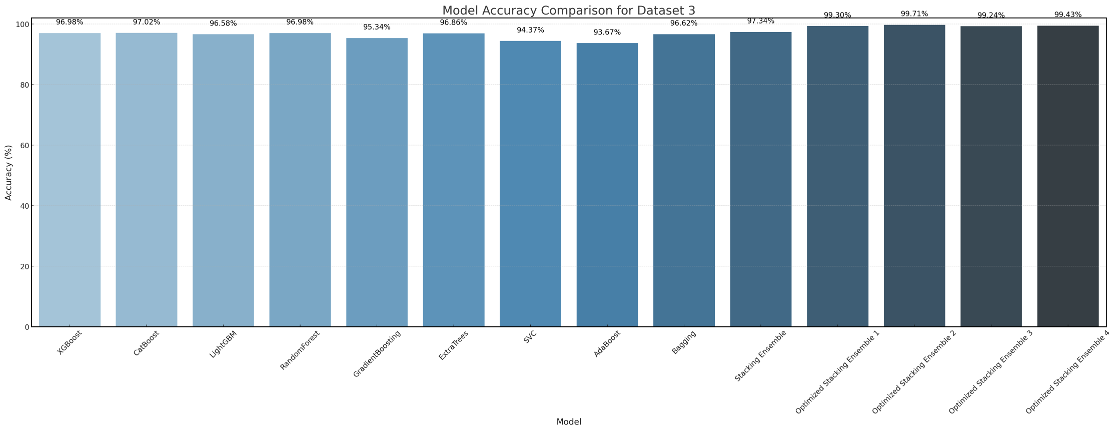

Fig. 13. Accuracy comparison on Dataset 3 for all models.

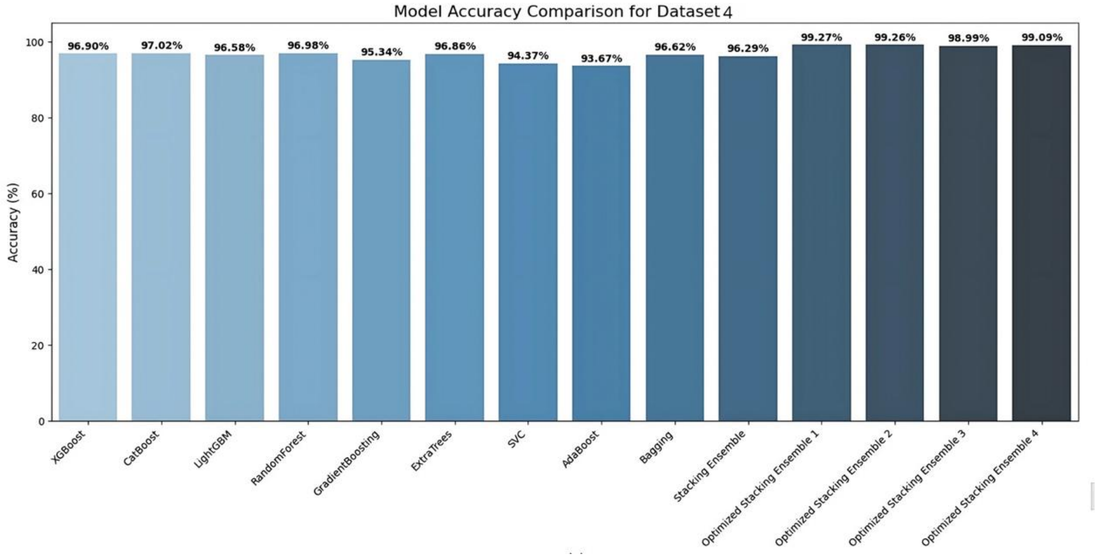

Fig. 14. Accuracy comparison on Dataset 4 for all models.

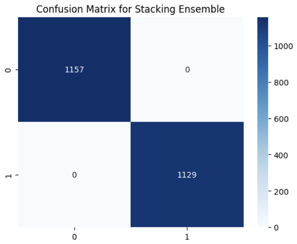

Fig. 15. Optimized ensemble stacking-1 (Dataset 1).

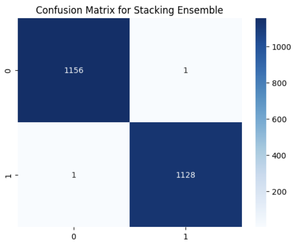

Fig. 16. Optimized ensemble stacking-2 (Dataset 1).

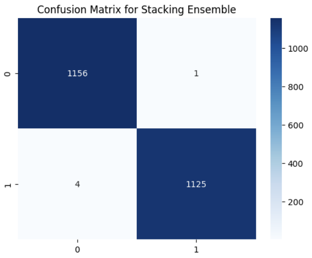

Fig. 17. Optimized ensemble stacking-3 (Dataset 1).

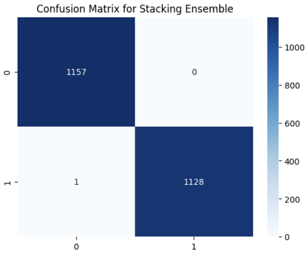

Fig. 18. Optimized ensemble stacking-4 (Dataset 1).

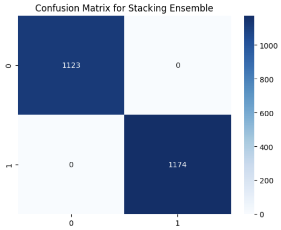

Fig. 19. Optimized ensemble stacking-1 (Dataset 2).

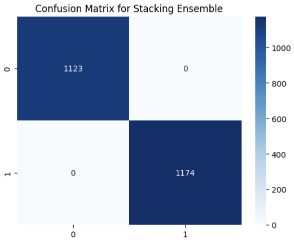

Fig. 20. Optimized ensemble stacking-2 (Dataset 2).

certificate indicators. These were effectively captured through hybrid feature engineering strategies.

To adapt to each dataset's scale and quality:

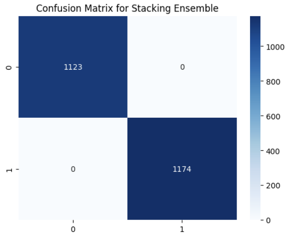

Fig. 21. Optimized ensemble stacking-3 (Dataset 2).

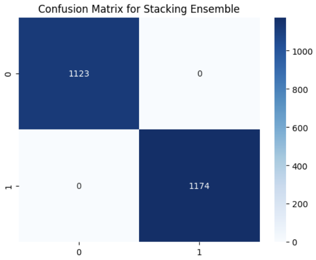

Fig. 22. Optimized ensemble stacking-4 (Dataset 2).

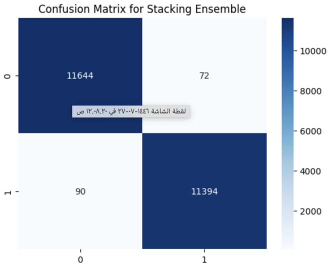

Fig. 23. Optimized ensemble stacking-1 (Dataset 3).

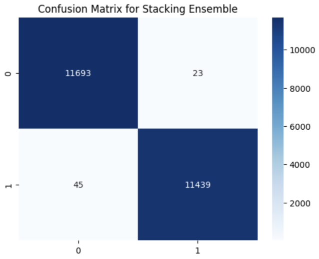

Fig. 24. Optimized ensemble stacking-2 (Dataset 3).

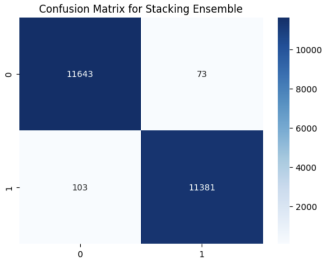

Fig. 25. Optimized ensemble stacking-3 (Dataset 3).

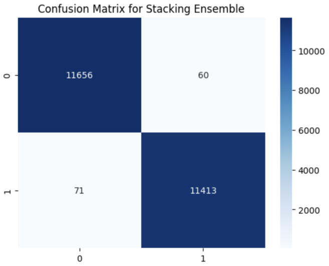

Fig. 26. Optimized ensemble stacking-4 (Dataset 3).

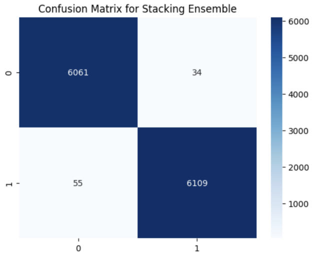

Fig. 27. Optimized ensemble stacking-1 (Dataset 4).

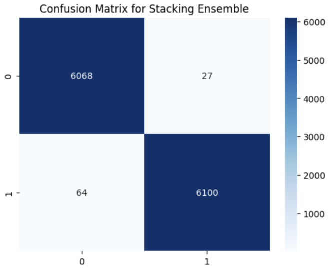

Fig. 28. Optimized ensemble stacking-2 (Dataset 4).

Feature encoding and scaling were uniformly applied. SMOTE was selectively used for unbalanced datasets. Cross-validation (StratifiedKFold) ensured consistent generalization across all experiments. Ultimately, the characteristics of each dataset informed key design decisions in the stacking architecture. While the same base and meta-model configurations were used across all datasets to ensure comparability, larger and imbalanced datasets necessitated specific adjustments—such as the application of SMOTE, reduced model iterations, and fewer cross-validation folds—to manage computational complexity without sacrificing accuracy. preprocessing, standardization, and encoding strategies were uniformly applied. These adaptations highlight how dataset size, balance, and structure influenced model pruning strategies and the depth of optimization required to achieve efficient and high-performing ensemble detection systems.

B. Performance Comparison of the Proposed Approach with Previous Studies

To objectively assess the effectiveness of the proposed optimized ensemble stacking model, it is essential to compare its performance with existing approaches reported in the literature.

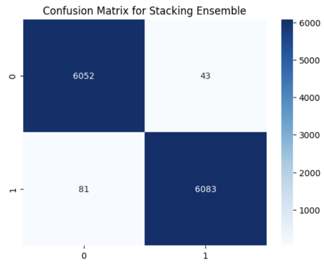

Fig. 29. Optimized ensemble stacking-3 (Dataset 4).

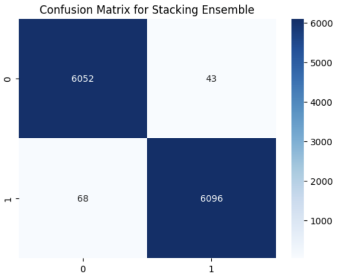

Fig. 30. Optimized ensemble stacking-4 (Dataset 4).

Several recent studies have explored various ensemble learning techniques to enhance phishing website detection, including optimized stacking, such as Genetic Algorithm (GA), to optimize the parameters of various ensemble classifiers, Diverse Base Learners, Hyperparameter Tuning, multilayer-stacked models, and boosting-based ensembles. This section presents a comparative analysis of classification accuracy across these studies, focusing on the same or similar benchmark datasets. A comparative summary of classification accuracy reported in these studies, focusing on similar benchmark datasets, is provided in Table XI.

TABLE XI. Comparison with Related Studies of the Proposed Method

Paper	Classifiers	Accuracy (%)
[5]	optimized stacking	D3:97.3%
[9]	Multilayer-Stacking	D3:96.79%, D4:98.43%
[15]	Boosting-based multi-layer stacked ensemble model	D4:96.16%, D3:98.95%
[4]	Stacking ensemble classifier	D1:98.20, D3:97.48%

Our enhanced ensemble stacking architecture clearly demonstrates advantages over existing methods. For example, [5] achieved 97.3% accuracy on Dataset 3 using an optimized stacking approach, while [9] reported 96.79% on Dataset 3 and 98.43% on Dataset 4 with multilayer stacking. Similarly, boosting-based multi-layer ensembles [15] obtained 98.95% on Dataset 3 and 96.16% on Dataset 4. In contrast, our optimized stacking variants consistently exceed 99% accuracy on the same datasets, with Optimized Stacking-2 reaching 99.71% on Dataset 3 and 99.28% on Dataset 4. Furthermore, while [4] reported 97.48% accuracy on Dataset 3, our method achieved over 99% under the same conditions. These comparisons highlight that our approach not only achieves higher accuracy, but also exhibits greater robustness when applied to imbalanced datasets (Datasets 3 and 4), where many prior methods suffered performance drops. By integrating CatBoost as a meta-learner, employing diverse base classifiers, and applying robust validation strategies, our model advances the state-of-the-art in phishing website detection.

C. Limitations

While the optimized ensemble stacking models achieved state-of-the-art performance across multiple datasets, several limitations should be acknowledged. First, training complex ensembles with multiple strong base learners and a CatBoost meta-model incurs significant computational cost and longer execution times, particularly for very large datasets. Second, achieving 100% accuracy on smaller balanced datasets may indicate a potential risk of overfitting, underscoring the need for careful validation on real-world, unseen data. Third, phishing tactics continuously evolve, meaning that static models trained on historical datasets may degrade in performance over time. Therefore, periodic retraining and the incorporation of adaptive or online learning mechanisms are essential to maintain robustness.

Finally, while SMOTE and preprocessing strategies helped address class imbalance, synthetic oversampling may not fully capture the diversity of real phishing examples. Future work should validate the models on live network traffic and diverse sources to ensure practical applicability.

VII. Conclusions

This study demonstrated the effectiveness of optimized ensemble stacking models in enhancing phishing website detection across datasets with diverse characteristics. The proposed framework consistently achieved superior results, including 100% accuracy on the two balanced datasets (Datasets 1 and 2) and above 99% accuracy on the larger and imbalanced datasets (Datasets 3 and 4). These outcomes confirm the robustness and scalability of the approach under varying data conditions. Among the four optimized variants, Optimized Stacking-2 and Optimized Stacking-4 provided the best trade-off between detection accuracy and computational efficiency, making them well-suited for real-world deployment.

The key contributions of this work include: systematic optimization of diverse base learners within the stacking framework, replacement of traditional Logistic Regression with CatBoost as the meta-classifier to improve generalization and handle imbalance, and comprehensive validation across multiple datasets to ensure reliability and robustness. These advances distinguish the proposed method from prior approaches that often relied on fixed classifiers, limited datasets, or less rigorous optimization.

From a practical standpoint, the findings highlight that phishing detection systems can achieve both high accuracy and efficiency by adopting optimized ensemble stacking. The integration of explainability tools such as SHAP further enhances usability by providing interpretable insights for security analysts, enabling informed decision-making in operational environments.

Future work will focus on extending this research by validating the framework on live phishing traffic, evaluating inference latency for real-time deployment, and exploring semi-supervised or adversarial learning techniques to counter evolving zero-day attacks. Additionally, lightweight versions of the model will be developed to improve adaptability in resource-constrained environments, such as IoT and mobile devices.

In conclusion, this research establishes optimized ensemble stacking as a highly effective and practical solution for phishing website detection, offering both state-of-the-art accuracy and strong adaptability to dynamic cybersecurity challenges.

References

[1] S. Aslam, H. Aslam, A. Manzoor, H. Chen, and A. Rasool, "Antiphishstack: Lstm-based stacked generalization model for optimized phishing url detection," *Symmetry*, vol. 16, no. 2, p. 248, 2024.

[2] F. S. Bidabadi and S. Wang, "A new weighted ensemble model for phishing detection based on feature selection," *arXiv preprint arXiv:2212.11125*, 2022.

[3] P. Meena, P. Singla, and P. Ranjan, "Enhanced phishing url detection through stacked machine learning model," in *2024 International Conference on Intelligent Systems for Cybersecurity (ISCS)*. IEEE, 2024, pp. 1–6.

[4] A. Newaz, F. S. Haq, and N. Ahmed, "A sophisticated framework for the accurate detection of phishing websites," *arXiv preprint arXiv:2403.09735*, 2024.

[5] M. Al-Sarem, F. Saeed, Z. G. Al-Mekhlafi, B. A. Mohammed, T. Al-Hadhrami, M. T. Alshammari, A. Alreshidi, and T. S. Alshammari, "An optimized stacking ensemble model for phishing websites detection," *Electronics*, vol. 10, no. 11, p. 1285, 2021.

[6] S. S. M. M. Rahman, T. Islam, and M. I. Jabiullah, "Phishstack: evaluation of stacked generalization in phishing urls detection," *Procedia Computer Science*, vol. 167, pp. 2410–2418, 2020.

[7] A. F. Nugraha, R. F. A. Aziza, and Y. Pristyanto, "Penerapan metode stacking dan random forest untuk meningkatkan kinerja klasifikasi pada proses deteksi web phishing," *Jurnal Infomedia: Teknik Informatika, Multimedia, dan Jaringan*, vol. 7, no. 1, pp. 39–44, 2022.

[8] D. W. Kiseki, V. Havyarimana, L. Zabagunda, W. I. Wail, T. Niyonsaba *et al.*, "Artificial intelligence in cybersecurity to detect phishing," *Journal of Computer and Communications*, vol. 12, no. 12, pp. 91–115, 2024.

[9] L. R. Kalabarige, R. S. Rao, A. Abraham, and L. A. Gabralla, "Multilayer stacked ensemble learning model to detect phishing websites," *Ieee Access*, vol. 10, pp. 79 543–79 552, 2022.

[10] H. A. Wabi, J. A. Ojeniyi, I. Idris, and S. O. Subairu, "Stack ensemble model for detection of phishing website," in *2024 International Conference on Science, Engineering and Business for Driving Sustainable Development Goals (SEB4SDG)*. IEEE, 2024, pp. 1–6.

[11] Z. G. Al-Mekhlafi, B. A. Mohammed, M. Al-Sarem, F. Saeed, T. A. Hadhrami, M. T. Alshammari, A. Alreshidi, and T. S. Alshammari, "Phishing websites detection by using optimized stacking ensemble model," *Comput. Syst. Sci. Eng.*, vol. 41, pp. 109–125, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:240426096

[12] B. A. Mohammed and Z. G. Al-Mekhlafi, "Optimized stacking ensemble model to detect phishing websites," in *International Conference on Advances in Cybersecurity*, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:244846728

[13] S. Giri and S. Banerjee, "Ensemble learning approach for phishing website detection using an optimal greedy stacking model," *Journal of The Institution of Engineers (India): Series B*, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:272639107

[14] M. K. Pandey, M. K. Singh, S. Pal, and B. B. Tiwari, "Prediction of phishing websites using stacked ensemble method and hybrid features selection method," *SN Computer Science*, vol. 3, pp. 1–11, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:252498524

[15] L. R. Kalabarige, R. S. Rao, A. R. Pais, and L. A. Gabralla, "A boosting-based hybrid feature selection and multi-layer stacked ensemble learning model to detect phishing websites," *IEEE Access*, vol. 11, pp. 71 180–71 193, 2023.

[16] N. Belsare, P. Sonaje, A. Ahmed, V. Gugale, S. Barve, and D. Chikmurge, "Enhancing phishing website detection using stacking ensemble techniques," *2024 3rd International Conference for Advancement in Technology (ICONAT)*, pp. 1–9, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:274641655

[17] R. Soleymanzadeh, M. Aljasim, M. W. Qadeer, and R. Kashef, "Cyberattack and fraud detection using ensemble stacking," *AI*, vol. 3, no. 1, pp. 22–36, 2022.

[18] M. A. Alsharaiah, A. A. Abu-Shareha, M. Abualhaj, L. H. Baniata, O. Adwan, A. Al-Saaidah, and M. Oraiqat, "A new phishing-website detection framework using ensemble classification and clustering," 2023.

[19] Y. Wei and Y. Sekiya, "Sufficiency of ensemble machine learning methods for phishing websites detection," *IEEE Access*, vol. 10, pp. 124 103–124 113, 2022.

[20] N. Lawrance, M.-A. Guerry, and G. Petrides, "Cost-sensitive stacking: an empirical evaluation," *arXiv preprint arXiv:2301.01748*, 2023.

[21] G. Dharmaraju, T. N. Kumar, P. P. Mohan, R. R. Pbv, and A. Lakshmanarao, "Phishing website detection through ensemble machine learning techniques," in *2024 2nd International Conference on Computer, Communication and Control (IC4)*. IEEE, 2024, pp. 1–5.

[22] Z. Ghaleb Al-Mekhlafi, B. Abdulkarem Mohammed, M. Al-Sarem, F. Saeed, T. Al-Hadhrami, M. T. Alshammari, A. Alreshidi, and T. Sarheed Alshammari, "Phishing websites detection by using optimized stacking ensemble model," *Computer Systems Science and Engineering*, vol. 41, no. 1, pp. 109–125, 2022.

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

[24] H. Baliyan and A. R. Prasath, "Enhancing phishing website detection using ensemble machine learning models," in *2024 OPJU International Technology Conference (OTCON) on Smart Computing for Innovation and Advancement in Industry 4.0*. IEEE, 2024, pp. 1–8.

[25] M. S. I. Ovi, M. H. Rahman, and M. A. Hossain, "Phishguard: A multilayered ensemble model for optimal phishing website detection," *arXiv preprint arXiv:2409.19825*, 2024.

[26] A. Hannousse, "Web page phishing detection [data set]," https://doi.org/10.17632/c2gw7fy2j4.1, 2020, mendeley Data.

[27] M. K. Chaurasia, "Web page phishing detection [data set]," https://www.kaggle.com/datasets/manishkc06/web-page-phishing-detection, 2021, kaggle.

[28] G. Vrbancic, "Phishing websites dataset," https://data.mendeley.com/datasets/72ptz43s9v/1, 2020, mendeley Data, vol. 1.

[29] J. Kolla, S. Praneeth, M. S. Baig, and G. reddy Karri, "A comparison study of machine learning techniques for phishing detection," *Journal Of Business And Information Systems (e-ISSN: 2685-2543)*, vol. 4, no. 1, pp. 21–33, 2022.

[30] K. Omari, "Comparative study of machine learning algorithms for phishing website detection," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 9, 2023.

[31] M. R. Chinguwo and R. Dhanalakshmi, "Detecting cloud based phishing attacks using stacking ensemble machine learning technique," *International Journal for Research in Applied Science & Engineering Technology (IJRASET). ISSN*, pp. 2321–9653, 2023.

[32] B. Deekshitha, C. Aswitha, C. S. Sundar, and A. K. Deepthi, "Url based phishing website detection by using gradient and catboost algorithms," *Int. J. Res. Appl. Sci. Eng. Technol*, vol. 10, no. 6, pp. 3717–3722, 2022.

[33] M. Khatun, A. Mozumder, N. Polash, M. R. Hasan, K. Ahammad, and M. S. Shaiham, "An approach to detect phishing websites with features selection method and ensemble learning," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 13, no. 8, 2022.

[34] M. Z. Hassan, M. H. Kabir, and M. A. Hamja, "Phishing website identification: Unleashing the potential of machine learning and stacking ensembles techniques," *Journal of Science and Technology*, vol. 122, p. 129, 2024.

[35] X. Zhou and R. M. Verma, "Phishing sites detection from a web developer's perspective using machine learning." in *HICSS*, 2020, pp. 1–10.

[36] scikit-learn developers, *scikit-learn: Machine Learning in Python*, 2024, version 1.7.0. [Online]. Available: https://scikit-learn.org

[37] XGBoost Developers, *XGBoost: Scalable, Portable and Distributed Gradient Boosting*, 2024, version 3.1.0. [Online]. Available: https://xgboost.readthedocs.io

[38] LightGBM Developers, *LightGBM: A Fast, Distributed, High Performance Gradient Boosting (GBDT, GBRT, GBM or MART) Framework*, 2024, version 4.3.0. [Online]. Available: https://lightgbm.readthedocs.io

[39] CatBoost Developers, *CatBoost: Gradient Boosting with Categorical Features Support*, 2024, version 1.2.3. [Online]. Available: https://catboost.ai

[40] Hunter, J. D. and matplotlib development team, *matplotlib: Visualization with Python*, 2024, version 3.8.4. [Online]. Available: https://matplotlib.org

[41] Waskom, M. L. and seaborn development team, *seaborn: Statistical Data Visualization*, 2024, version 0.13.2. [Online]. Available: https://seaborn.pydata.org

[42] pandas development team, *pandas: Powerful Python Data Analysis Toolkit*, 2024, version 2.2.2. [Online]. Available: https://pandas.pydata.org

[43] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith *et al.*, “Array programming with numpy,” *Nature*, vol. 585, no. 7825, pp. 357–362, 2020.