

Research Article

ERGA-Phish: Optimized Ensemble Learning Framework with Ridge Regression and Genetic Algorithm for Improved Phishing URL Detection

Chris Ugochukwu Chinaka , Obi Chukwuemeka Nwokonkwo ,
Adetokunbo MacGregor John-Otumu* , Udoka Felista Eze 

Department of Information Technology, Federal University of Technology, Owerri, Nigeria

Abstract

Phishing remains a persistent cybersecurity threat that exploits deceptive web links to compromise users and organizations. Although machine learning and deep learning techniques have demonstrated strong phishing detection capabilities, challenges related to class imbalance, feature redundancy, generalization, and model optimization remain. This study proposes an optimized stacking ensemble framework that integrates Decision Tree, Random Forest, Naïve Bayes, XGBoost, and Artificial Neural Network (ANN) classifiers, with an ANN serving as the meta-learner. The PhiUSIIL Phishing URL Dataset, obtained from the UCI Machine Learning Repository, contains 235,795 labeled URLs, comprising 134,850 legitimate and 100,945 phishing URLs. Data preprocessing involved duplicate removal, mean imputation, one-hot encoding, normalization, and Synthetic Minority Oversampling Technique (SMOTE), producing a balanced dataset of 269,700 instances. Ridge Regression was then applied to reduce the feature space from 57 to 50 relevant features. The stacking ensemble was further optimized using a Genetic Algorithm (GA) to identify suitable parameter configurations for improved predictive performance. Using an 80:20 train-test split, the proposed model achieved 99.54% accuracy, 98.75% precision, 98.85% recall, 98.68% F1-score, and 0.98 ROC-AUC. The proposed model outperformed the evaluated baseline classifiers and most benchmarked studies, although one reported approach achieved 99.89% accuracy. The findings demonstrate that combining heterogeneous classifiers, feature selection, class balancing, and evolutionary optimization can provide a strong framework for phishing URL detection.

Keywords

Phishing Detection, Genetic Algorithm, Stacking Ensemble, Machine Learning, Cybersecurity, URL Classification

1. Introduction

The widespread adoption of the internet, cloud computing, and mobile technologies has profoundly transformed the digital landscape, enabling new possibilities for commerce, communication, and social interaction [1]. These advancements have boosted innovation and connectivity but have

also increased reliance on online systems, thereby expanding exposure to cybersecurity threats [2]. Phishing remains among the most common and sophisticated types of cyber threats [3]. Phishing, which was first reported in 1996 in connection with AOL account breaches, involves using social

*Correspondence: Adetokunbo MacGregor John-Otumu (adetokunbo.johnoutumu@futo.edu.ng)

Received: 29 August 2026; Accepted: 11 September 2026; Published: 30 September 2026



Copyright: © The Author(s), 2026. Published by Science Publishing Group. This is an **Open Access** article, distributed under the terms of the Creative Commons Attribution 4.0 License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution and reproduction in any medium, provided the original work is properly cited.

engineering to persuade victims into surrendering sensitive data such as account logins and financial records [4]. This is typically achieved through fraudulent emails, malicious hyperlinks, or websites designed to mimic trusted platforms.

In the closing quarter of 2022, the APWG recorded more than a million phishing attempts, with about 23% involving the financial sector [5, 6]. One of the most widely used phishing vectors is the alteration of Uniform Resource Locators (URLs). Crafted to closely resemble legitimate links, phishing URLs lure unsuspecting users into credential theft, unauthorized transactions, or malware execution [6, 7]. The consequences include identity theft, financial losses, and large-scale data breaches, underscoring phishing's persistence as a global challenge.

Traditional detection methods such as blacklists and manual inspection are increasingly inadequate, as attackers constantly adapt strategies to evade rule-based systems [8]. Consequently, traditional machine learning (ML) has emerged as a promising solution, learning patterns from data to classify URLs as legitimate or malicious with higher accuracy [9, 10, 28]. Algorithms such as Random Forest (RF), K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Logistic Regression (LR), Artificial Neural Network (ANN), Naïve Bayes (NB), and Extreme Gradient Boosting (XGBoost) have all been utilized effectively to phishing URL detection [11-13, 26]. Typical features include URL length, domain age, SSL/TLS usage, and character distribution, with performance measured using accuracy, precision, recall, F1-score, and ROC AUC.

Despite these advances, key challenges persist. Phishing tactics evolve rapidly, diminishing the generalization ability of existing models on unseen data [14]. Datasets are frequently not balanced, with legitimate URLs outnumbering phishing samples, biasing classifiers toward the majority class [15, 16]. Ensemble learning (EL) has gained prominence for addressing such issues, as it integrates multiple classifiers to improve robustness and predictive performance. RF and XGBoost, in particular, demonstrate strong resilience against overfitting and perform well with large datasets [12].

Complementary to ensemble learning, optimization methods such as Genetic Algorithms (GA) and feature selection strategies like Ridge Regression (L2 regularization) have been shown to enhance phishing detection by reducing dimensionality, improving efficiency, and fine-tuning model parameters [17]. However, few existing studies unify ensemble learning (EL), optimization, and feature selection into a single framework tailored for phishing URL detection.

To bridge this gap, the present study introduces an optimized ensemble learning framework that integrates five classifiers: Decision Tree, Random Forest, Naïve Bayes, XGBoost, and Artificial Neural Network within a stacking structure. Ridge Regression was applied to identify the most significant features, while the Genetic Algorithm fine-tuned the ANN meta-learner to enhance both generalization and pre-

dictive performance. The rest of this paper is organized as follows: Section II provides a review of related works, Section III outlines the proposed methodology, Section IV presents the experimental results along with a discussion of the findings, and Section V concludes with key outcomes and directions for future research.

2. Related Works

This section reviews existing research on phishing detection using different methodological approaches. It first examines traditional machine learning (ML) techniques, followed by deep learning (DL) approaches for detecting more complex phishing patterns. It then considers ensemble and optimization techniques that combine multiple models to improve predictive performance. Recent studies addressing phishing through social media, AI-generated content, and real-time detection are also examined. Finally, existing benchmarking frameworks and awareness models are discussed before the major research gaps motivating the present study are identified.

2.1. Machine Learning Approaches

Traditional machine learning models have been widely applied to phishing detection and have demonstrated strong predictive performance across different datasets. A Random Forest (RF) model using the PhishTank dataset containing 11,000 samples developed for phishing website detection as a plugin service for existing web browsers and it achieved 96% accuracy, 97% precision, 99% recall, and 98% F1-score [12]. Similarly, a Logistic Regression-based model using a limited feature set also reported 98.42% accuracy and 98.8% precision [18].

Furthermore, Similarly, Chiew et al. [19] employed Random Forest and achieved 94.6% accuracy, although limitations in the dataset affected the generalizability of the model. Rashid et al. [20] integrated Principal Component Analysis (PCA) with Support Vector Machine (SVM) and obtained 95.66% accuracy without hyperparameter optimization. Ogude and Onwuachu [21] also developed the PHISH-BOT model for phishing detection, although its applicability was constrained by the coverage of the dataset used. These studies demonstrate the effectiveness of traditional ML techniques while also pointing to persistent challenges involving dataset limitations, feature dependency, and generalization to real-world phishing scenarios [26, 28].

2.2. Deep Learning Approaches

Deep learning techniques have demonstrated considerable potential for identifying complex and nonlinear phishing patterns. John-Otumu et al. [13] applied Ridge Regression in combination with an Artificial Neural Network (ANN) and achieved 98.58% accuracy using 235,795 samples. This

demonstrates the potential benefit of combining feature selection with neural network-based classification, although the integration of regression-based feature selection, ensemble learning, and optimization remains relatively underexplored. Bibi et al. [29] employed an improved multilayer Convolutional Neural Network (CNN) for phishing website detection and achieved 99.1% accuracy. Chy [30] also investigated the application of machine learning techniques to phishing prediction and prevention. Although DL approaches can achieve high predictive performance, their computational requirements and limited interpretability can create challenges for practical deployment.

2.3. Ensemble and Optimization Techniques

Ensemble learning approaches have gained attention because they combine the strengths of multiple classifiers and can provide more robust predictions than individual models. Subasi and Kremic [31] compared AdaBoost and MultiBoost for phishing website detection and reported 97.61% accuracy. Al-Sarem et al. [32] proposed an optimized stacking ensemble model for phishing website detection and achieved 97.16% accuracy. Almohani et al. [33] employed semantic features with Gradient Boosting and Random Forest classifiers and reported approximately 97% accuracy. Alazaidah et al. [14] also compared Random Forest, Filtered Classifier, and J-48 models, with reported accuracy values ranging from 89.95% to 97.26%. These findings demonstrate the potential of ensemble approaches to improve phishing detection; however, the combined use of diverse classifiers, systematic feature selection, and evolutionary optimization within a single framework remains insufficiently explored.

2.4. Social Media, AI-Generated Content, and Real-Time Detection

The phishing threat has expanded beyond conventional email and website-based attacks to include social media platforms and AI-generated content. Hammed and Soyemi [34] applied associative rule mining enhanced with the Firefly Algorithm to classify phishing attacks on social media and reported 99.3% accuracy. Eze and Shamir [35] examined AI-based phishing email attacks and reported strong performance using machine learning techniques, including Naïve Bayes. Keerthivasan et al. [36] emphasized the importance of improving web phishing security and real-time detection, although detailed performance metrics were not reported. These studies demonstrate that phishing detection systems must increasingly account for rapidly changing attack channels and attack-generation techniques.

2.5. Frameworks and Awareness Models

Benchmarking and awareness frameworks have also contributed to phishing research. El-Aassal et al. [37] introduced

PhishBench as a benchmarking and evaluation framework for phishing detection research. However, its validation was primarily focused on benchmarking rather than extensive real-world deployment. Abufardeh and Falah [38] examined phishing countermeasures and emphasized the importance of cybersecurity awareness and user training in reducing the impact of social engineering attacks. Although these approaches are important components of a broader cybersecurity strategy, awareness and benchmarking frameworks alone cannot replace automated technical mechanisms for detecting phishing attacks.

2.6. Synthesis and Research Gap

The reviewed studies show that considerable progress has been made in phishing detection using ML, DL, ensemble learning, and optimization techniques. However, important gaps remain. Traditional ML models continue to depend heavily on the quality and relevance of extracted features, while DL models can be computationally demanding and difficult to interpret. Existing ensemble approaches have also not fully explored the integration of diverse classifiers with advanced feature selection and evolutionary optimization. In addition, the emergence of phishing attacks through social media, AI-generated content, and other dynamic platforms creates a need for detection systems that are more adaptable and robust to changing attack patterns.

The present study therefore addresses these gaps by developing an Optimized Stacking Ensemble Framework that integrates five diverse classifiers, Ridge Regression for feature selection, and Genetic Algorithm-based optimization of the ANN meta-learner. The objective is to improve phishing detection performance, robustness, and adaptability while reducing some of the limitations associated with individual and conventional ensemble classifiers.

3. Materials and Methods

This section presents the methodological framework employed in developing the optimized ensemble ML model for phishing URL detection. It details the research design, experimental setup, dataset characteristics, preprocessing procedures, feature selection approach, ensemble architecture, base classifiers, optimization strategy, and evaluation metrics.

3.1. Research Design

The study adopts a quantitative experimental design, focusing on the development and validation of an optimized ensemble framework for phishing URL detection. The framework integrates five (5) classifiers (DT, RF, NB, XGB and ANN) through a stacking architecture, coupled with optimization and feature selection to improve detection performance. A benchmark phishing URL dataset was utilized, and the model was

implemented in Python 3.9 using Scikit-learn, XGBoost, TensorFlow/Keras, and deployed via Flask. Performance was evaluated using standard classification metrics.

3.2. Dataset Description

This study utilized the PhiUSIIL Phishing URL Dataset obtained from the UCI ML Repository, which consists of 235,795 labeled URLs, of which 134,850 are legitimate and 100,945 are phishing. Each record is described by 57 features

encompassing:

- 1) Lexical attributes: URL length, subdomain count, presence of special characters.
- 2) Domain-based properties: WHOIS information, domain age, DNS records.
- 3) Network-level features: SSL/TLS usage, IP address inclusion.

The dataset file size is 54.2 MB, and its diverse features provide a robust benchmark for phishing detection as summarized in [Table 1](#)

Table 1. Characteristics of the PhiUSIIL Phishing URL dataset.

Dataset Name	PhiUSIIL_Phishing_URL_Dataset
Dataset File Size	54.2MB
Source	UCI Machine Learning Repository
Link to dataset	https://archive.ics.uci.edu/dataset/967/phiusiil+phishing+url+dataset
Feature Type	Real, Categorical, Integer
Number of Instances	235,795
Number of Features	57
Legitimate cases	134,850
Phishing cases	100,945

3.3. Data Preprocessing and Feature Engineering

Preprocessing ensured data quality and readiness for training:

- 1) *Data Cleaning*: duplicate records were removed:

$$D' = \{x_i \in D \mid x_i \neq x_j \text{ for } i \neq j\} \quad (1)$$

where D is the original PhiUSIIL Phishing URL Dataset, and D' is the cleaned dataset.

- 2) *Missing Value Treatment*: Mean imputation for numerical features was used to replace null values:

$$x_j = \frac{1}{N} \sum_{i=1}^N x_{ij}, \text{ if } x_j \text{ is missing} \quad (2)$$

where x_{ij} is the value of feature j in record i , and N is the number of non-missing values.

- 3) *Feature Encoding*: One-Hot Encoding was utilized to transform categorical attributes into numerical representations as shown in eqn (3):

$$x_{ij}^{(k)} = \begin{cases} 1, & \text{if category of feature } j \text{ in record } i = k \\ 0, & \text{Otherwise} \end{cases} \quad (3)$$

where k is one of the possible categories of feature j .

- 4) *Normalization*: Min-Max normalization is utilized to scale numerical features into the range $[0, 1]$:

$$x'_{ij} = \frac{x_{ij} - \min(x_j)}{\max(x_j) - \min(x_j)} \quad (4)$$

where x_{ij} is the original value, $\min(x_j)$ and $\max(x_j)$ are the minimum and maximum of feature j , and x'_{ij} is the normalized value.

- 5) *Handling Class Imbalance*: SMOTE was applied to correct the imbalance in the PhiUSIIL Phishing URL Dataset, where the number of legitimate URLs was considerably higher than phishing URLs. SMOTE creates artificial instances of the minority class, in this case phishing URLs, to achieve a more balanced dataset, thereby enhancing the model's reliability during training (see [Table 2](#)).

$$x_{new} = x_i + \delta \times (x_{nn} - x_i), \delta \sim U(0, 1) \quad (5)$$

where:

x_i is a minority class sample,

x_{nn} is one of its k -nearest neighbors,

δ is a uniformly distributed random number in the range $[0, 1]$,

x_{new} is the generated synthetic instance.

Table 2. Dataset distribution before and after SMOTE balancing.

		Legitimate URLs	Phishing URLs
Initial Dataset	235,795	134,850	100945
After Smote Operation	269,700	134,850	134,850

6) Train-Test Split Strategy: After applying SMOTE, the PhiUSIIL Phishing URL Dataset contained a total of 269,700 instances, evenly distributed between legitimate and phishing URLs, as shown in equations (6) and (7):

$$N_{total} = 269,700, N_{legit} = 134,850, N_{phish} = 134,850$$

The dataset was divided into training and testing subsets using an 80:20 split ratio.

SMOTE was applied before the train-test split, resulting in a balanced dataset of 269,700 instances, which was subsequently divided into 80% training and 20% testing subsets. The resulting training set contained 215,760 instances, while the test set contained 53,940 instances.

a) Training Set (80%)

The total number of training instances was computed as:

$$N_{train} = N_{total} \times 0.80 \quad (6)$$

$$N_{train} = N_{total} \times 0.80 = 269,700 \times 0.80 = 215,760$$

The distribution of legitimate and phishing URLs within the training set was calculated as follows:

$$N_{legit,train} = N_{legit} \times 0.80 = 134,850 \times 0.80 = 107,880$$

$$N_{legit,train} = N_{phish} \times 0.80 = 134,850 \times 0.80 = 107,880$$

Thus, the training set contained 215,760 instances, consisting of 107,880 legitimate URLs and 107,880 phishing URLs.

b) Test Set (20%)

The total number of testing instances was derived as:

$$N_{test} = N_{total} \times 0.20 \quad (7)$$

$$N_{test} = N_{total} \times 0.20 = 269,700 \times 0.20 = 53,940$$

The breakdown of legitimate and phishing URLs in the testing set is given in Equations (11) and (12):

$$N_{legit,test} = N_{legit} \times 0.20 = 134,850 \times 0.20 = 26,970$$

$$N_{legit,test} = N_{phish} \times 0.20 = 134,850 \times 0.20 = 26,970$$

Therefore, the testing set consisted of 53,940 instances, including 26,970 legitimate URLs and 26,970 phishing URLs. This balanced split ensured equal representation of phishing and legitimate URLs in both subsets, thereby improving the model's ability to generalize and enhancing detection reliability.

Table 3. Experimental setup and implementation configuration of the proposed stacking ensemble framework.

Component	Description
Dataset Path	C:\PHISHING_PROJECT\DATASET\PhiUSIIL_Phishing_URL_Dataset.csv
Output Directory	C:\PHISHING_PROJECT\ML_CLASSIFIERS\ENSEMBLE
Target Variable	Label
Feature Transformation	Duplicate removal, mean imputation, one-hot encoding, and Min-Max normalization
Data Balancing	SMOTE for balancing class distribution
Base Models	Gaussian NB, DT, RF, XGBoost, Multilayer Perceptron (ANN)
Ensemble Model	Stacking Classifier with ANN as final estimator
Optimization Technique	Genetic Algorithm (GA) to optimize the ANN meta-learner
Performance Metrics	Accuracy, ROC-AUC, classification report, confusion matrix, and ROC curve

3.4. Feature Selection

Ridge regression (L2 regularization) was applied to reduce redundancy and improve model efficiency in the phishing dataset, which originally contained 57 features. Unlike Lasso, ridge regression shrinks less informative feature coefficients toward zero without removing them, thereby retaining meaningful predictors while minimizing noise.

The optimization function is expressed as:

$$J(\beta) = \sum_{i=1}^n (y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij})^2 + \lambda \sum_{j=1}^p \beta_j^2 \quad (8)$$

where λ controls coefficient shrinkage. By adjusting this parameter, ridge regression reduced the original 57 features to

50 highly relevant features. This dimensionality reduction decreased computational cost, enhanced generalization by reducing overfitting, and improved interpretability by focusing on the most impactful predictors for phishing URL detection.

3.5. Stacking Ensemble Framework

The proposed framework integrates five base classifiers: Decision Tree (DT), Random Forest (RF), Naïve Bayes (NB), XGBoost, and Artificial Neural Network (ANN). Their predictions are combined through an ANN meta-learner, whose hyperparameters are optimized using a Genetic Algorithm (GA). The proposed framework is depicted in Figure 1.

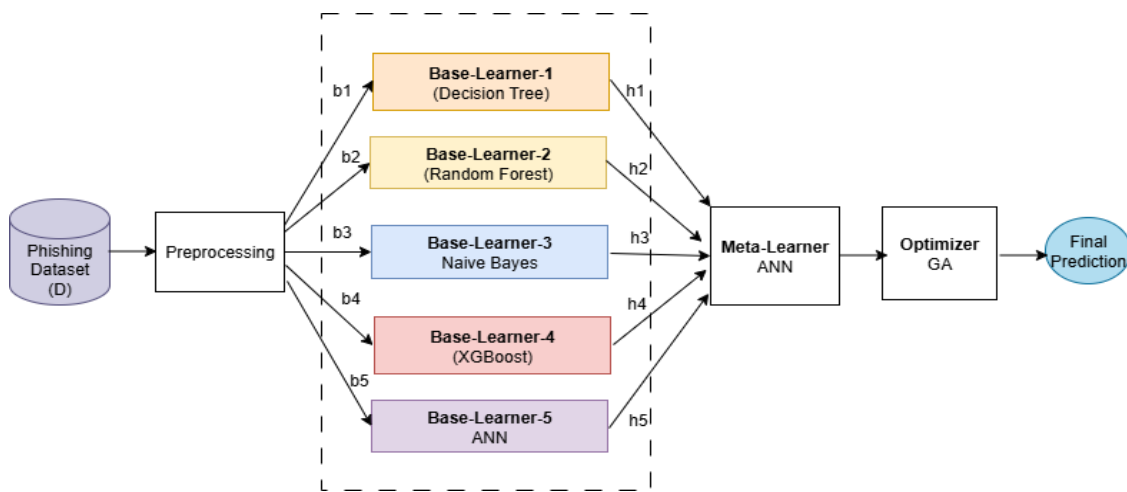


Figure 1. Proposed Stacking Ensemble Framework.

Figure 1 presents the proposed stacking ensemble framework for phishing URL detection. The framework begins with the phishing dataset (D), which serves as the input to the system. The dataset first passes through a preprocessing stage, where the raw data are prepared for subsequent modelling. The preprocessed data are then distributed to five different base learners, labelled b1 to b5, to allow different ML algorithms to learn and make predictions from the same input data.

The five base learners consist of Decision Tree (Base-Learner-1), Random Forest (Base-Learner-2), Naïve Bayes (Base-Learner-3), XGBoost (Base-Learner-4), and Artificial Neural Network (Base-Learner-5). Each learner independently processes the preprocessed data and produces an output, represented as h1 to h5 in the framework. Using different learning algorithms at this stage allows the system to capture different patterns and characteristics of phishing and legitimate URLs.

The outputs from the five base learners are subsequently passed to the meta-learner, which is an Artificial Neural Network (ANN). Rather than relying on the prediction of a single classifier, the ANN meta-learner receives and combines the

outputs generated by the five base learners to produce a more informed final classification.

The framework further incorporates a Genetic Algorithm (GA) optimizer specifically for the ANN meta-learner. The GA searches for suitable ANN hyperparameter configurations by evaluating candidate solutions based on validation performance. The optimized ANN meta-learner subsequently combines the outputs of the five base classifiers to produce the final phishing or legitimate classification.

3.6. Classifiers in the Ensemble

The proposed ensemble integrates five base classifiers, each with unique learning strategies and mathematical formulations, which collectively improve robustness and generalization.

1) Decision Tree (DT)

DT partitions the feature space into recursive regions by maximizing information gain or minimizing impurity. The Gini Index for a node t is defined as:

$$\text{Gini}(t) = 1 - \sum_{i=1}^C p_i^2 \quad (9)$$

where p_i is the probability of class i in node t , and C is the total number of classes. Splits are chosen to minimize impurity across child nodes.

2) Random Forest (RF)

RF builds multiple Decision Trees using bootstrap aggregation (bagging) and random feature subsets. The prediction is obtained through majority voting.

$$y = \text{mode}\{h_1(x), h_2(x), \dots, h_T(x)\} \quad (10)$$

where $h_1(x)$ is the prediction of the t^{th} tree and T is the total number of trees.

3) Naïve Bayes (NB):

NB is a probabilistic classifier based on Bayes' theorem with the assumption of conditional independence among features:

$$P(y|x_1, x_2, \dots, x_n) = \frac{P(y) \prod_{i=1}^n P(x_i|y)}{P(y|x_1, x_2, \dots, x_n)} \quad (11)$$

The predicted class is:

$$\hat{y} = \arg \max_y P(y) \prod_{i=1}^n P(x_i|y) \quad (12)$$

4) Extreme Gradient Boosting (XGBoost)

XGBoost is an optimized gradient boosting framework.

$$Ob_j = \sum_{i=1}^n l(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k) \quad (13)$$

with regularization:

$$\Omega(f_k) = \gamma T + \frac{1}{2} \lambda \|w\|^2 \quad (14)$$

where l is the loss function, T is the number of leaves, and w are leaf weights.

5) Artificial Neural Network (ANN)

ANN was used to capture nonlinear relationships among the phishing URL features. Unlike traditional classifiers, ANN can model complex feature interactions such as abnormal domain lengths, deceptive subdomain patterns, and suspicious keyword usage, which are common in phishing URLs.

The ANN architecture consists of an input layer representing the selected features, multiple hidden layers for hierarchical feature extraction, and an output layer for classification.

In the proposed stacking framework, the ANN is also employed as the meta-learner. It receives the outputs of the five base classifiers and learns their combined contribution for the final phishing URL classification. Its hyperparameters are subsequently optimized using the Genetic Algorithm as described in Section 3.7.

The forward propagation process in one hidden layer is given by:

$$z^{(l)} = w^{(l)} a^{(l-1)} b^{(l)} \quad (15)$$

$$a^{(l)} = f(z^{(l)}) \quad (16)$$

where $w^{(l)}$ and $b^{(l)}$ denote the weights and biases at layer l , $a^{(l-1)}$ represents the activations from the previous layer, and $f(\cdot)$ is the nonlinear activation function. For binary classification between phishing and legitimate URLs, the final prediction is achieved using a sigmoid activation function:

$$\hat{y} = \sigma W^{(L)} a^{(L-1)} b^{(L)} \quad (17)$$

where $\hat{y} \in [0,1]$ corresponds to the probability of a URL being classified as phishing. By learning discriminative weights through backpropagation, ANN emphasized critical phishing indicators while suppressing redundant or noisy signals. This enabled improved detection accuracy and generalization across diverse phishing attack strategies in the dataset.

3.7. Optimization Strategy

The stacking meta-learner was implemented as an Artificial Neural Network (ANN) and optimized using a Genetic Algorithm (GA). The GA was employed to search for an effective configuration of the ANN meta-learner by evaluating different combinations of its hyperparameters. Each candidate solution, represented as a chromosome, consisted of the ANN learning rate, number of hidden layers, and number of neurons per layer.

The GA optimization began with an initial population of candidate ANN hyperparameter configurations. Each candidate configuration was evaluated using the validation loss of the stacking model. Candidate solutions with better validation performance were assigned higher selection probabilities. New candidate solutions were then generated through crossover and mutation, and the process was repeated over successive generations. The candidate configuration that produced the minimum validation loss was selected as the optimal configuration for the ANN meta-learner.

In the experiments, the GA used a population size of 10 and was executed for 10 generations. The crossover probability was set to 0.7, while the mutation probability was set to 0.2. Three-fold cross-validation was used to evaluate the performance of candidate configurations during the optimization process. The hyperparameter chromosome is defined as:

$$g = [g_1, g_2, \dots, g_d]T, g \in G \subset R^{dr} \times Z^{di} \times C^{dc} \quad (18)$$

Validation loss for a given hyperparameter set is:

$$L_{val}(\theta(g)) = \frac{1}{|D_{val}|} \sum_{(x,y) \in D_{val}} \ell(\theta(g)(x), y) \quad (19)$$

The fitness function is defined as:

$$F(g) = -L_{val}(\theta(g)) \quad (20)$$

Population of chromosomes at generation t :

$$P^{(t)} = \{g_1^{(t)}, g_2^{(t)}, \dots, g_N^{(t)}\} \quad (21)$$

Selection probability for chromosome i:

$$P_i^{(t)} = \frac{\max\{0, F(g_i^{(t)})\}}{\sum_{j=1}^N \max\{0, F(g_j^{(t)})\}} \quad (22)$$

Crossover operator:

$$g_{child} = \alpha g_a + (1 - \alpha) g_b, \alpha \sim U(0, 1) \quad (23)$$

Mutation operator:

$$g'_i = \begin{cases} g_i + \delta, & \text{with prob, } p_m, \delta \sim \mathcal{N}(0, \delta^2) \\ g_i, & \text{with prob, } 1 - p_m \end{cases} \quad (24)$$

3.8. Evaluation Metrics

To assess the classification performance, the confusion matrix entries; TP, TN, FP, and FN were used to compute standard evaluation metrics. Where TP=True Positives, TN=True Negatives, FP=False Positives, and FN=False Negatives.

$$\text{Accuracy (Acc)} = \frac{TP + TN}{TP + FN + FP + TN} \quad (25)$$

$$\text{Precision (Prec)} = \frac{TP}{TP + FP} \quad (26)$$

$$\text{Recall (Sensitivity)} = \frac{TP}{TP + FN} \quad (27)$$

$$\text{F1 - score} = 2 \times \frac{(\text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})} \quad (28)$$

$$\text{ROC - AUC} = \int_0^1 TPR(FPR) d(FPR) \quad (29)$$

3.9. Proposed Algorithm

The proposed algorithm combines feature selection, ensemble learning, and evolutionary optimization for accurate phishing URL detection. Ridge Regression is first applied to eliminate redundant features and strengthen the dataset's discriminative ability. The selected features are then fed into a stacking ensemble comprising DT, RF, NB, XGBoost, and ANN as base classifiers. An ANN also functions as the meta-learner, integrating the outputs of the base models to enhance generalization. Hyperparameter tuning of the meta-learner is performed using a Genetic Algorithm, which improves adaptability and reduces overfitting. The detailed workflow is presented in Algorithm 1.

Algorithm 1: Proposed Optimized Stacking Ensemble Model

Input

PhiUSIIL Phishing URL Dataset (D), 57 features, 235,795 instances

Output

Best classifier (C), Meta-learner (M*), Accuracy (Acc), Precision (Prec), Recall (Rec), F1-score, ROC-AUC

Begin

Dataset Preparation)

(a) Remove duplicates: $D' \leftarrow \{x_i \in D \mid x_i \neq x_j, i \neq j\}$

(b) Mean imputation $x_j = \frac{1}{N} \sum_{i=1}^N x_{ij}$, if x_j is missing

(c) Encode categorical features with one-hot encoding $x_{ij}^{(k)} \begin{cases} 1, & \text{if category of feature } j \text{ in record } i = k \\ 0, & \text{Otherwise} \end{cases}$

(d) Normalize numerical features: $x'_{ij} = \frac{x_{ij} - \min(x_j)}{\max(x_j) - \min(x_j)}$

(e) Apply SMOTE: $x_{new} = x_i + \delta \times (x_{nm} - x_i), \delta \sim U(0, 1)$

(f) Balanced dataset: Nlegit = Nphish = 134,850

Train-Test Split

Training: 215,760, Testing: 53,940

Balanced: Legit = Phish = 107,880 (train), 26,970 (test)

Feature Selection

Apply Ridge Regression: $J(\beta) = \sum_{i=1}^n (y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij})^2 + \lambda \sum_{j=1}^p \beta_j^2$

Reduced features: 57 $\rightarrow$ 50

Stacking Ensemble Framework

Base classifiers: {DT, RF, NB, XGB, ANN}

Meta-learner: ANN optimized via GA

(a) Train base models B_k on Dtrain

(b) Collect predictions $\hat{y}_k$

(c) Train meta-learner M on stacked outputs

(d) Optimize M with GA

Genetic Algorithm Optimization

Population size: 10

Number of generations: 10

Crossover probability: 0.7

Mutation probability: 0.2

Cross-validation: 3-fold

Chromosome: $g = [lr, nlayers, nneurons]$

The chromosome represents the learning rate, number of hidden layers, and number of neurons per layer of the ANN meta-learner.

Population $P_t = \{g_1^t, g_2^t, \dots, g_m^t\}$

Fitness: $\min(\text{Lossval}(g))$

Selection: Roulette probability $p_i = \frac{f_i}{\sum f_j}$

Crossover: $gchild = \text{crossover}(gp, gq)$

Mutation: $g^l = \text{mutate}(g)$

Return optimal hyperparameters g^*

Evaluation Metrics

$$(a) \text{Accuracy} = \frac{TP + TN}{TP + FN + FP + TN}$$

$$(b) \text{Precision} = \frac{TP}{TP + FP}$$

$$(c) \text{Recall (Sensitivity)} = \frac{TP}{TP + FN}$$

$$(d) F1 - \text{score} = 2 \times \frac{(\text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})}$$

$$(e) \text{ROC} - \text{AUC} = \int_0^1 TPR(FPR) d(FPR), TPR = \frac{TP}{TP + FN}, FPR = \frac{FP}{FP + TN}$$

Return

Best classifier (C), optimized meta-learner (M^*), metrics {Acc, Prec, Rec, F1, AUC}

3.10. Experimental Setup

The experimental evaluation was carried out on a high-performance workstation configured with an Intel Core i7 processor operating at 2 GHz, 16 GB of system memory, and an NVIDIA GPU with 4 GB of dedicated memory. This hardware specification provided sufficient computational power for handling large datasets and accelerating DL training processes. The implementation environment was based on Python

version 3.9, utilizing a combination of established ML and DL libraries, including TensorFlow/Keras for model construction and training, Scikit-learn for classical machine learning and evaluation functions, and Flask for lightweight web-based deployment and interaction. The overall configuration of the experimental platform is summarized in Table 4, whereas the specific hyperparameters and training settings of the proposed optimized ensemble model are comprehensively outlined in Table 5.

Table 4. Hardware and software configuration.

Component	Specification
Processor	Intel Core i7, 2 GHz
RAM	16 GB
GPU	NVIDIA 4 GB
Language	Python 3.9
Libraries	TensorFlow, Keras, Scikit-learn, Flask

Table 5. Experimental setup and hyperparameter configuration.

Parameter	Value/Range
Test Data Split Ratio	20% of the balanced dataset for testing
SMOTE Random State	42
Train-Test Split Seed	42
GA Population Size	10
GA Generations	10
Crossover Probability	0.7
Mutation Probability	0.2
ANN Meta-learner Parameters Optimized by GA	Learning rate, number of hidden layers, and neurons per layer
Stacking Cross-Validation	3-Fold Cross-Validation used in GA for performance evaluation

4. Results

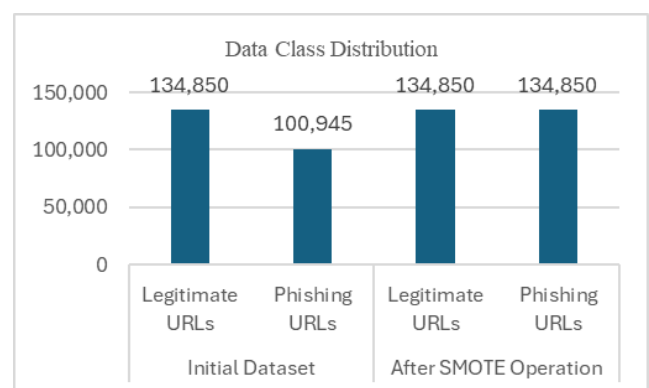
This section presents the empirical evaluation of the proposed Genetic Algorithm (GA)-optimized stacking ensemble framework. The experiments were conducted using the Phi-USIIL phishing URL dataset, following preprocessing, balancing, and optimization steps described earlier. Results are reported in terms of dataset distribution, comparative model performance, and benchmarking against existing studies.

4.1. Experimental Results

4.1.1. Dataset Distribution Analysis

The dataset exhibited class imbalance, with legitimate URLs outnumbering phishing ones, which could distort the training process. To mitigate this, SMOTE was employed, as shown in Figure 2. After SMOTE, the distribution was balanced, ensuring equal representation of both classes. This step

was essential for improving classifier robustness and minimizing false negatives.

**Figure 2.** Comparison of dataset distribution prior to and following the SMOTE procedure.

4.1.2. Baseline Model Performance

Several baseline classifiers, including DT, RF, NB,

XGBoost, ANN, K-Nearest Neighbors (K-NN), Logistic Regression (LR), and Support Vector Machine (SVM), were evaluated to establish benchmark results. The model performance was measured using accuracy, precision, recall, f1-score, and ROC-AUC.

Table 6 presents the performance of the baseline classifiers and the proposed optimized stacking ensemble model for URL phishing detection. The results show that the proposed model achieved the highest accuracy of 99.54%, representing an improvement of 1.27% points over the best-performing baseline model, Naïve Bayes (98.27%). This improvement is important because it means that the proposed ensemble produced fewer classification errors than the individual classifiers. In terms of error rate, the proposed model recorded only 0.46% error, compared with 1.73% for Naïve Bayes, the best baseline in accuracy. This corresponds to approximately 73.4% reduction in classification error relative to Naïve Bayes.

The proposed model also achieved the highest precision of 98.75%. Naïve Bayes produced the closest baseline result at 98.50%, followed by ANN at 97.80% and Random Forest at 96.98%. The higher precision of the proposed model indicates that it was highly effective in limiting false positive predictions, meaning that most URLs classified as phishing were actually phishing cases.

For recall, the proposed model obtained 98.85%, which was substantially higher than every baseline classifier. Decision Tree recorded the highest baseline recall at 97.56%, followed

by XGBoost at 97.92% and Naïve Bayes at 98.22%. The proposed model therefore improved recall by 0.63 percentage points over Naïve Bayes and 0.93 percentage points over XGBoost, indicating stronger capability to identify phishing instances and reduce missed phishing cases.

The proposed model also recorded an F1-score of 98.68%, slightly higher than Naïve Bayes at 98.54%, XGBoost at 97.87%, and Random Forest at 97.70%. This suggests that the proposed ensemble achieved a strong balance between precision and recall. The relatively small improvement over Naïve Bayes is nevertheless meaningful because the ensemble simultaneously achieved a considerably higher overall accuracy.

Regarding ROC-AUC, the proposed model achieved 0.98, which was equal to the highest value reported among the baseline models, namely XGBoost and ANN. Although the proposed model did not exceed these models in ROC-AUC, the result confirms its strong ability to distinguish between phishing and legitimate URLs across classification thresholds.

The results demonstrate that the optimized stacking ensemble provided the strongest overall performance, particularly in accuracy, precision, recall, and F1-score. Its advantage can be attributed to the combination of multiple complementary classifiers rather than relying on the decision boundary of a single model. The GA optimization of the ANN meta-learner further supports the selection of an effective combination of the base-model outputs. Therefore, the results provide empirical support for the proposed ensemble as a robust approach for phishing detection.

Table 6. Performance comparison of baseline classifiers and the proposed optimized ensemble model.

Models	Accuracy	Precision	Recall	F1-Score	ROC_AUC
DT	96.66	96.58	97.56	97.40	0.97
RF	97.97	96.98	96.90	97.70	0.96
NB	98.27	98.50	98.22	98.54	0.97
XGBoost	97.65	96.92	97.92	97.87	0.98
ANN	97.98	97.80	96.66	97.65	0.98
K-NN	96.84	94.82	95.65	96.56	0.95
LR	96.98	96.56	96.68	95.69	0.94
SVM	95.98	94.56	95.54	94.88	0.93
Proposed Model	99.54	98.75	98.85	98.68	0.98

4.1.3. Benchmarking with Existing Studies

To further assess the robustness of the proposed model, its accuracy was benchmarked against findings from prior phishing detection research as depicted in Figure 3.

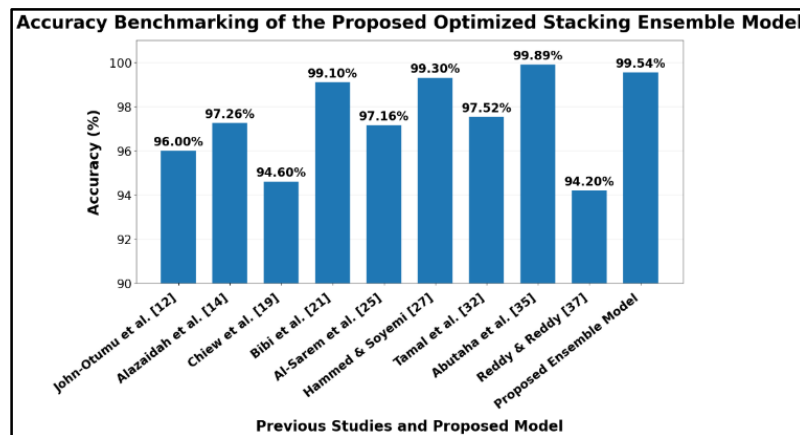


Figure 3. Accuracy evaluation of the proposed ensemble model with prior studies.

Figure 3 compares the accuracy of the proposed optimized stacking ensemble with selected phishing detection approaches reported in previous studies. The proposed model achieved 99.54% accuracy, outperforming most of the benchmarked approaches, including John-Otumu et al. [12] (96.00%), Chiew et al. [19] (94.60%), Al-Sarem et al. [25] (97.16%), Alazaidah et al. [14] (97.26%), Tamal et al. [32] (97.52%), Bibi et al. [21] (99.10%), Hammed and Soyemi [27] (99.30%), and Reddy and Reddy [37] (94.20%). The proposed model therefore demonstrates a competitive performance advantage, with improvements ranging from 0.24 to 5.34 percentage points over these approaches. However, Abutaha et al. [35] achieved a marginally higher accuracy of 99.89%, indicating that the proposed model does not constitute the absolute highest-performing approach among those reviewed. Instead, its 99.54% accuracy places it within the highest-performing group. This performance can be attributed to the integration of heterogeneous classifiers through stacking, Ridge Regression-based feature selection, SMOTE-based class balancing, and Genetic Algorithm optimization. Nevertheless, because the benchmarked studies were conducted using different datasets and experimental configurations, the results should be interpreted as a literature-based comparison rather than a controlled head-to-head evaluation.

5. Discussion

This section examines the performance of the optimized ensemble model relative to baseline classifiers and prior research, emphasizing its contributions as well as the challenges that persist.

5.1. Real-time Implementation and Dynamic Attacks

A notable limitation in prior studies [22, 23, 25] is the limited focus on real-time deployment and adaptability to evolving phishing attacks. For example, while [22] achieved 97.52% accuracy using OFVA, their work did not extend to real-time

testing. Similarly, [23] encountered challenges in handling continuously evolving attack vectors. In contrast, the proposed model integrates diverse classifiers with GA-based optimization, achieving 99.54% accuracy and demonstrating greater robustness for dynamic, real-world environments. Nonetheless, as with many existing studies, further evaluation under real-time conditions remains necessary.

5.2. Datasets

Dataset size, diversity, and balance play a critical role in determining detection accuracy. Prior studies, including those by [27] and [19], were affected by dataset imbalance or limited scope, which constrained their generalizability. In the present study, SMOTE was applied (See Figure 2) to address imbalance, thereby improving classifier stability and yielding strong performance across the standard evaluation metrics utilized (See Table 6). These findings highlight the relevance of preprocessing and balancing techniques for achieving reliable phishing detection outcomes.

5.3. Feature Selection and Classification

Feature dependency remains a notable limitation in many ML-based approaches [12, 18, 20]. For example, the PCA-SVM model proposed by [20] achieved 95.66% accuracy but did not incorporate feature optimization, while the Logistic Regression model by [18] was restricted by a limited set of features. In contrast, the proposed ensemble integrated diverse classifiers, enhancing robustness against feature-specific weaknesses. The stacking framework, supported by Ridge Regression and an ANN meta-learner, achieved a strong balance between feature richness and predictive power, resulting in 98.75% precision and 98.85% recall as shown in Table 6.

5.4. Comprehensive Evaluation

Several preceding studies testified only limited performance metrics, which makes fair comparisons challenging.

For example, Nwokoro et al. [24] did not provide detailed evaluation measures, thereby reducing reproducibility. In contrast, this study addressed such gaps by presenting a complete set of standard metrics which includes accuracy, precision, recall, F1 score, and ROC AUC for all models (See Table 6). Such comprehensive reporting enhances comparability across studies, strengthens reproducibility, and offers clearer insights into the relative strengths and weaknesses of different models.

5.5. Exploration of Advanced Techniques

Advanced optimization and hybrid approaches have demonstrated considerable promise in phishing detection. For example, [31] applied AdaBoost and MultiBoost, reaching an accuracy of 97.61%, while [32] proposed stacking ensembles that achieved 97.16% accuracy. Despite these contributions, optimization techniques remain underexplored in this domain.

The present study distinguishes itself by incorporating Genetic Algorithms (GA) for hyperparameter tuning within a stacking ensemble framework, resulting in a superior accuracy of 99.54% and surpassing most comparable works (Figure 3). These findings highlight the value of meta-heuristic optimization in significantly enhancing the performance of ensemble learning models for phishing detection.

5.6. Scalability and Generalization

Although DL models such as Convolutional Neural Networks (CNNs) proposed by [29] and NLP-driven architectures by [30] have demonstrated high accuracy, they are often computationally demanding and less interpretable. In contrast, the proposed ensemble strikes a balance between accuracy and computational efficiency, making it more suitable for scalable, real-world deployment. The achieved ROC-AUC score of 0.98 further underscores its strong generalization capability relative to individual models. However, large-scale cross-dataset validation remains necessary to confirm scalability across diverse phishing scenarios.

5.7. Implications and Future Work

The findings of this study provide several important implications for both research and practice. The outstanding performance of the proposed optimized stacking ensemble model (99.54% accuracy, 98.75% precision, 98.85% recall, and 0.98 ROC-AUC) demonstrates that integrating diverse classifiers with a GA-tuned meta-learner significantly enhances phishing detection.

This outcome reinforces that hybrid and optimization-driven models are better equipped to handle the complexity of phishing URLs compared to standalone classifiers or conventional ensembles. In addition, addressing dataset imbalance through techniques such as SMOTE proved crucial, contributing to greater stability and improved generalization of the model. This emphasizes the role of data preprocessing as a critical stage in phishing detection pipelines. Moreover, the

use of a comprehensive evaluation across multiple performance metrics highlights the importance of reproducibility and transparent benchmarking in phishing research, helping to close a persistent gap in many prior studies.

Despite these contributions, future work should explore real-time deployment of the proposed model in live network environments, as well as cross-dataset validation using heterogeneous sources such as social media, mobile applications, and AI-generated phishing content. Additionally, explainable AI (XAI) methods could be integrated into the framework to improve interpretability, addressing transparency concerns associated with ensemble and optimization-based approaches. Finally, scalability studies on cloud and edge platforms will be necessary to evaluate the practical deployment potential of the model in large-scale cybersecurity systems.

6. Conclusions

This study proposed an optimized stacking ensemble model for phishing URL detection, integrating multiple machine learning classifiers with ANN as a meta-learner and Genetic Algorithm-based hyperparameter optimization. To mitigate dataset imbalance, SMOTE was applied, leading to a balanced distribution of phishing and legitimate URLs. Experimental results demonstrated that the proposed ensemble consistently outperformed baseline classifiers and recent state-of-the-art models, achieving 99.54% accuracy, 98.75% precision, 98.85% recall, 98.68% F1-score, and a 0.98 ROC-AUC score. These results confirm the robustness, stability, and effectiveness of the approach compared to traditional ML, deep learning, and ensemble techniques reported in previous studies.

The findings highlight several key contributions: (i) the effectiveness of optimization-driven stacking ensembles for phishing detection, (ii) the importance of addressing dataset imbalance through oversampling, and (iii) the value of comprehensive evaluation across multiple metrics to ensure reproducibility and comparability. The study also reinforces the need to balance predictive performance with computational feasibility, enabling practical deployment in cybersecurity systems.

Nevertheless, certain limitations remain. Real-time evaluation, scalability across diverse data sources, and transparency of decision processes require further investigation. Future research will focus on deploying the model in live environments. It should also consider investigating the integration of explainable AI methods to enhance interpretability, and finally, extending the framework to handle phishing threats in emerging domains such as social media, mobile platforms, and AI-generated malicious content. This work advances phishing detection research by demonstrating that a carefully optimized ensemble framework can achieve state-of-the-art performance while providing a scalable foundation for practical, adaptable, and explainable cybersecurity solutions.

Abbreviations

ANN	Artificial Neural Network
AUC	Area Under the Curve
DL	Deep Learning
DT	Decision Tree
GA	Genetic Algorithm
KNN	K-Nearest Neighbors
LR	Logistic Regression
ML	Machine Learning
NB	Naïve Bayes
PCA	Principal Component Analysis
RF	Random Forest
SMOTE	Synthetic Minority Oversampling Technique
SVM	Support Vector Machine
URL	Uniform Resource Locator

Author Contributions

Chris Ugochukwu Chinaka: Conceptualization, Data curation, Methodology, Visualization, Writing – original draft

Obi Chukwuemeka Nwokonkwo: Methodology, Supervision, Validation

Adetokunbo MacGregor John-Otumu: Formal Analysis, Investigation, Methodology, Visualization

Udoka Felista Eze: Methodology, Supervision, Validation

Data Availability Statement

The data that support the findings of this study can be found at: <https://archive.ics.uci.edu/dataset/967/phiusiil+phishing+url+dataset>.

Conflicts of Interest

The authors declare no conflicts of interest.

References

- [1] R. Alazaidah, G. Samara, S. Almatarnah, M. Hassan, M. Aljaidi, H. Mansur. Multi-label classification based on associations, *Applied Sciences*. 2023, 13(8), 5081. <https://doi.org/10.3390/app13085081>
- [2] M. Al-Khateeb, M. Al-Mousa, A. Al-Sherideh, D. Almajali, M. Asassfeha, H. Khafajeh. Awareness model for minimizing the effects of social engineering attacks in web applications, *International Journal of Data and Network Science*. 2023, 7(2), 791-800. <https://doi.org/10.5267/j.ijdns.2022.11.004>
- [3] N. Q. Do, A. Selamat, O. Krejcar, E. Herrera-Viedma, H. Fujita. Deep learning for phishing detection: Taxonomy, current challenges and future directions, *IEEE Access*. 2022, 10, 36429-36463. <https://doi.org/10.1109/ACCESS.2022.3162222>
- [4] M. Al-Luwaici, A. K. Junoh, W. Alzoubi, R. Alazaidah, and W. Al-Luwaici, "New features selection method for multi-label classification based on the positive dependencies among labels," *Solid State Technology*, 2020, 63(2s), p. 9896,
- [5] O. Osho, A. Oluyomi, S. Misra, R. Ahuja, R. Damasevicius, R. Maskeliunas. Comparative evaluation of techniques for detection of phishing URLs, *Communications in Computer and Information Science*. 2019, 1051, 385-394. https://doi.org/10.1007/978-981-13-9190-3_35
- [6] A. Bhardwaj, F. Al-Turjman, V. Sapra, M. Kumar, T. Stephan. Privacy-aware detection framework to mitigate new-age phishing attacks, *Computers & Electrical Engineering*. 2021, 95, 107546. <https://doi.org/10.1016/j.compeleceng.2021.107546>
- [7] Z. Alkhalil, C. Hewage, L. Nawaf, I. Khan. Phishing attacks: A recent comprehensive study and a new anatomy, *Frontiers in Computer Science*. 2021, 3, 563060. <https://doi.org/10.3389/fcomp.2021.563060>
- [8] R. Alazaidah, M. Almaiah, M. Al-Luwaici. Associative classification in multi-label classification: An investigative study, *Jordanian Journal of Computers and Information Technology*. 2021, 1, 7(2). <https://doi.org/10.5455/jjcit.71-1615297634>
- [9] U. S. Nwankwo, O. C. Nwokonkwo, C. Ikerionwu, A. M. John-Otumu, U. F. Eze. Toward Adaptive Intelligent Intrusion Detection: A Review of Zero-day Attack Detection Methods and Future Cyber Defense Directions. *International Journal of Information and Communication Sciences*, 2026, 11(2), 47-69. <https://doi.org/10.11648/j.ijics.20261102.13>
- [10] H. Takci, N. Fatema. Highly accurate spam detection with the help of feature selection and data transformation, *International Arab Journal of Information Technology*. 2023, 20(1), 29-37. <https://doi.org/10.34028/iajit/20/1/4>
- [11] I. K. Nti, O. Narko-Boateng, A. F. Adekoya, A. R. Somanathan. Stacknet based decision fusion classifier for network intrusion detection, *International Arab Journal of Information Technology*. 2022, 19(3A), 478-490. <https://doi.org/10.34028/iajit/19/3A/6>
- [12] A. M. John-Otumu, M. M. Rahman, C. U. Oko. An efficient phishing website detection plugin service for existing web browsers using random forest classifier, *American Journal of Artificial Intelligence*. 2021, 5(2), 66-75. <https://doi.org/10.11648/j.ajai.20210502.12>
- [13] A. M. John-Otumu, V. O. Aniugo, V. C. Nwachukwu. Hy-RANN-UPD: Enhancing phishing URL detection using ridge regression-based feature selection and artificial neural networks, *International Journal of Computer Applications*. 2025, 186(78), 56-62. <https://doi.org/10.5120/ijca2025926422>
- [14] R. Alazaidah, A. Alshaikh, M. R. Almousa, G. Samara. Web-site phishing detection using machine learning techniques, *Journal of Intelligent Information Systems*. 2024, 63(1), 147-161. <https://doi.org/10.1007/s10844-023-00843-4>

- [15] M. Ahsan, R. Gomes, A. Denton. SMOTE implementation on phishing data to enhance cybersecurity, *Proceedings of the IEEE International Conference on Electro/Information Technology*. 2018, 531-536.
<https://doi.org/10.1109/EIT.2018.8500134>
- [16] I. Hassan, R. H. Ali, Z. Ul Abideen, T. A. Khan, R. Kouatly. Significance of machine learning for detection of malicious websites on an unbalanced dataset, *Digital*. 2022, 2(4), 501-519.
<https://doi.org/10.3390/digital2040028>
- [17] A. Chaiban, D. Sovilj, H. Soliman, G. Salmon, X. Lin. Investigating the influence of feature sources for malicious website detection, *Applied Sciences*. 2022, 12(6), 2806.
<https://doi.org/10.3390/app12062806>
- [18] A. K. Jain, B. B. Gupta. A machine learning based approach for phishing detection using hyperlinks information, *Journal of Ambient Intelligence and Humanized Computing*. 2019, 10, 2015-2028. <https://doi.org/10.1007/s12652-018-0798-z>
- [19] K. L. Chiew, C. L. Tan, K. Wong, K. S. Yong, W. K. Tiong. A new hybrid ensemble feature selection framework for machine learning-based phishing detection system, *Information Sciences*. 2019, 484, 153-166.
<https://doi.org/10.1016/j.ins.2019.01.059>
- [20] J. Rashid, T. Mahmood, M. W. Nisar, T. Nazir. Phishing detection using machine learning technique, *Proceedings of the International Conference on Smart Systems and Emerging Technologies (SMARTTECH)*. 2020, 43-46.
<https://doi.org/10.1109/SMARTTECH49988.2020.00021>
- [21] U. C. Ogude, U. C. Onwuachu. Website phishing detection using machine learning algorithm, *Global Scientific Journal*. 2024, 12(4), 1554-1564.
- [22] M. A. Tamal, M. K. Islam, T. Bhuiyan, A. Sattar, N. U. Prince. Unveiling suspicious phishing attacks: Enhancing detection with an optimal feature vectorization algorithm and supervised machine learning, *Frontiers in Computer Science*. 2024, 6, 1428013. <https://doi.org/10.3389/fcomp.2024.1428013>
- [23] C. G. Onyiah, G. F. Yanwalo, N. E. Ajimah. Phishing URL detection: A basic machine learning approach, *The International Journal of Science & Technoledge*. 2024, 12(3), 8-14.
<https://doi.org/10.24940/theijst/2024/v12/i3/ST2403-001>
- [24] I. S. Nwokoro, O. J. Okesola, M. Q. Sambo, O. G. Oshodin, T. O. Akinfenwa, Z. K. Adom-Oduro, S. Y. Ahmed. Phishing attacks prevention using smart based artificial intelligence algorithms for cybersecurity awareness, *International Journal of Innovative Research in Technology*. 2024, 11(2), 1228-1235.
<https://doi.org/10.1177/10439862211001628>
- [25] M. Abutaha, M. Ababneh, K. Mahmoud, S. A. H. Baddar. URL phishing detection using machine learning techniques based on URLs lexical analysis, *Proceedings of the 12th International Conference on Information and Communication Systems (ICICS)*. 2021, 147-152.
<https://doi.org/10.1109/ICICS52457.2021.9464605>
- [26] S. Alrefaai, G. Özdemir, A. Mohamed. Detecting phishing websites using machine learning, *Proceedings of the 2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*. 2022, 1-6.
<https://doi.org/10.1109/HORA55278.2022.9799953>
- [27] V. S. S. Reddy, N. Reddy. Machine learning techniques for identifying and mitigating phishing attacks, *International Journal of Artificial Intelligence & Machine Learning*. 2024, 3(2), 116-129.
- [28] A. Aljofey, Q. Jiang, A. Rasool, H. Chen, W. Liu, Q. Qu, Y. Wang. An effective detection approach for phishing websites using URL and HTML features, *Scientific Reports*. 2022, 12, 8842.
<https://doi.org/10.1038/s41598-022-10841-5>
- [29] H. Bibi, S. R. Shah, M. M. Baig, M. Sharif, M. Mehmood, Z. Akhtar, K. Siddique. Phishing website detection using improved multilayered convolutional neural networks, *Journal of Computer Science*. 2024, 20(9), 1069-1079.
<https://doi.org/10.3844/jcssp.2024.1069.1079>
- [30] M. K. H. Chy. Securing the web: Machine learning's role in predicting and preventing phishing attacks, *International Journal of Scientific Research Archive*. 2024, 13(1), 1004-101.
<https://doi.org/10.30574/ijrsra.2024.13.1.1770>
- [31] A. Subasi, E. Kremic. Comparison of AdaBoost with multi boosting for phishing website detection, *Procedia Computer Science*. 2020, 168, 272-278.
<https://doi.org/10.1016/j.procs.2020.02.259>
- [32] M. Al-Sarem, F. Saeed, Z. G. Al-Mekhlafi, B. A. Mohammed, T. Al-Hadhrani, M. T. Alshammari, A. Alreshidi, T. S. Alshammari. An optimized stacking ensemble model for phishing websites detection, *Electronics*. 2021, 10(11), 1287.
<https://doi.org/10.3390/electronics10111287>
- [33] A. Almohani, M. Alauthman, M. T. Shatnawi, M. Alweshah, A. Alrosan, W. Alomoush, B. B. Gupta. Phishing website detection with semantic features based on machine learning classifiers: A comparative study, *International Journal of Semantic Web and Information Systems*. 2022, 18(1), 1-24.
<https://doi.org/10.4018/IJSWIS.296686>
- [34] M. Hammed, J. Soyemi. Classification of phishing attacks in social media using associative rule mining augmented with firefly algorithm, *International Journal of Computer Science and Engineering*. 2023, 6(6), 1-10.
<https://doi.org/10.30534/ijcse/2023/06662023>
- [35] C. S. Eze, L. Shamir. Analysis and prevention of AI-based phishing email attacks, *Electronics*. 2024, 13(10), 1839.
<https://doi.org/10.3390/electronics13101839>
- [36] A. Keerthivasan, J. Jayaprakash, A. Darwinsdivakar, S. P. Abishalom, S. Sabapathi. Enhancement in web phishing security, *International Journal of Research Publication and Reviews*. 2024, 5(5), 11462-11467.
- [37] A. El-Aassal, S. Baki, A. Das, R. M. Verma. An in-depth benchmarking and evaluation of phishing detection research for security needs, *IEEE Access*. 2020, 8, 22170-22192.
<https://doi.org/10.1109/ACCESS.2020.2970159>

- [38] S. Abufardeh, B. Falah. The state of phishing attacks and countermeasures, International Journal of Computer Science and Security (IJCSS). 2023, 17(4), 54-70.

Research Field

Chris Ugochukwu Chinaka: Information Systems and Security. Machine Learning, IT

Obi Chukwuemeka Nwokonkwo: IT Project Management, Digital Systems, Automation, Information Systems

Adetokunbo MacGregor John-Otumu: Machine Learning, Deep Learning, Computer Vision, MAS, Intelligent Systems

Udoka Felista Eze: Database Systems, Data Science, AI