HyRANN-UPD: Enhancing Phishing URL Detection using Ridge Regression-based Feature Selection and Artificial Neural Networks

Adetokunbo John-Otumu
Dept of Information Technology
Federal University of Tech
Owerri, Nigeria

Victor O. Aniugo
Dept of Mechatronics Eng
Federal University of Tech
Owerri, Nigeria

Victor C. Nwachukwu
Dept of Information Technology
Federal University of Tech
Owerri, Nigeria

ABSTRACT

Phishing attacks have become a major cybersecurity threat, making it essential to develop advanced detection models to protect online users. This study presents a machine learning-based approach for detecting phishing URLs, utilizing an Artificial Neural Network (ANN) to improve accuracy and reliability. The PhiUSIIL Phishing URL Dataset from the UCI Machine Learning Repository, containing 235,795 instances with 55 features, was used for training and evaluation. The dataset includes 134,850 legitimate URLs and 100,945 phishing URLs, with no missing values. To enhance performance, Ridge Regression (L2 Regularization) was applied to reduce the feature set from 55 to 50, improving efficiency without compromising accuracy. Several machine learning models which include Random Forest (RF), Naïve Bayes (NB), Logistic Regression, K-NN, XGBoost, and ANN were tested to compare their effectiveness. Among them, the ANN model outperformed the others, achieving an accuracy of 98.58%, precision of 97.80%, recall of 97.66%, and an F1-score of 97.65%. The ROC-AUC score of 0.98 further demonstrated the model's ability to differentiate between phishing and legitimate URLs. The proposed ANN model is efficient, scalable, and suitable for integration into existing security frameworks such as intrusion detection systems and anti-phishing tools. This research contributes to the growing field of AI-driven cybersecurity solutions, offering a highly effective and reliable approach to counter phishing threats.

General Terms

Cybersecurity, Machine Learning, Phishing Detection, Pattern Recognition, Artificial Intelligence, Algorithms, Web Security, Data Science.

Keywords

Phishing URL, Detection, Artificial Neural Networks, Ridge Regression, Feature Selection, Cybersecurity, Machine Learning.

1. INTRODUCTION

The rapid evolution of the internet, cloud computing, and mobile technology has transformed modern life, creating a vast digital ecosystem where businesses, social interactions, and daily activities flourish [1]. While this interconnectedness offers economic opportunities, it has also led to an alarming rise in cybersecurity threats, particularly phishing attacks, which have become more sophisticated over time [2-3].

Phishing is a cybercrime where attackers deceive users into disclosing sensitive information, such as passwords and financial details, through fraudulent emails, malicious links, or counterfeit websites. The first recorded phishing attack targeted America Online (AOL) in 1996 [4]. Today, phishing remains a major cybersecurity concern, with the Anti-Phishing Work Group (APWG) reporting over a million incidents in late 2022, particularly targeting financial institutions [5-6]. A key element of phishing attacks is the use of deceptive URLs such as fake web addresses designed to resemble legitimate ones, luring users into providing confidential information. These URLs lead to fraudulent websites that facilitate identity theft, financial loss, and data breaches [6, 7]. Traditional detection methods, such as manual inspection and rule-based systems, are no longer sufficient due to attackers' evolving techniques. Consequently, researchers are leveraging Machine Learning (ML) to develop automated phishing detection models [8].

ML has proven effective in cybersecurity, fraud detection, and medical diagnosis. Random Forest improves accuracy by constructing multiple decision trees, while XGBoost, a more advanced boosting algorithm, optimizes performance through regularization techniques [9-10]. In phishing URL detection, ML models analyze URL features such as length, domain name, and special characters to distinguish between legitimate and phishing links. Commonly used algorithms include Random Forest (RF), Support Vector Machines (SVM), K-Nearest Neighbor (K-NN), and Artificial Neural Networks (ANN), with performance assessed using metrics such as accuracy, precision, recall, and F1-score [11]. Despite these advancements, phishing detection faces ongoing challenges due to attackers' adaptive techniques. Recent research suggests that ensemble learning approaches, particularly Random Forest and XGBoost, enhance detection accuracy by aggregating multiple models' predictions [12].

This study aims to develop an ML-based phishing URL detection model with the following objectives:

(a) Preprocess a phishing dataset from the UCI repository and apply Ridge Regression L2 Regularization for feature selection.
(b) Train six ML models such as ANN, Random Forest, Naive Bayes, Logistic Regression, and XGBoost using the preprocessed dataset.
(c) Evaluate model performance using accuracy, precision, recall, F1-score, and ROC-AUC metrics on an independent test set.
(d) Develop a web-based application using Python, HTML, CSS, and Flask to integrate the best-performing model for real-world use.

This research focuses exclusively on phishing URL detection and does not cover other phishing types such as email, voice (vishing), text, or social media phishing. By enhancing phishing URL detection, this study aims to strengthen digital security and mitigate cyber threats effectively.

2. RELATED WORKS

This section explores key previous studies on phishing URL detection, highlighting various methodologies and their effectiveness.

According to [13], a study was conducted on phishing detection using a dataset of 274,446 URLs, with 134,500 phishing URLs and 139,946 legitimate ones. They introduced an Optimal Feature Vectorization Algorithm (OFVA) alongside Supervised Machine Learning classifiers to improve detection accuracy. Their model achieved 97.52% accuracy, 97.50% precision, and an AUC of 97%. However, the study did not focus on real-time implementation, and there is a risk of overfitting due to the nature of their model. Research work by [14] also explored phishing detection but focused on analyzing website features. They employed the XGBoost algorithm, achieving an accuracy of 86.6%. While their approach showed promise, it may struggle to detect dynamic phishing attacks, where cybercriminals frequently change tactics to bypass detection systems.

Research work by [15] took a different approach by examining the structure of URLs, domain characteristics, and SSL/TLS information to detect phishing websites. They applied Machine Learning techniques, including Deep Learning and Natural Language Processing (NLP), to improve detection accuracy. The study demonstrated significant effectiveness in identifying phishing websites, but its scope was limited, as it did not fully address newer and more sophisticated phishing attack strategies. Also, research work by [16], explored the role of Artificial Intelligence (AI) in cybersecurity, particularly in preventing phishing attacks. They discussed AI-driven phishing protection methods and how they enhance security systems. However, the study lacked specific experimental results or performance metrics, making it difficult to evaluate the practical impact of their proposed solutions. A study on phishing detection was carried out by [17], using Logistic Regression (LR) and URL-based feature analysis. Their model achieved a high accuracy of 98.42%, a precision of 98.8%, and an F1-score of 98.59%. However, the study was limited by a narrow feature set, which may not generalize well to all phishing attempts. Additionally, there is a potential for data bias, as their dataset composition was not explicitly discussed. In another research work by [18], a phishing detection model was developed using feature selection techniques and a Random Forest classifier. Their approach yielded an accuracy of 94.6%, showing good performance. However, the reliance on Random Forest alone may limit the adaptability of the model. Additionally, the study used a relatively small dataset, which could impact the reliability of the results when applied to real-world scenarios. A research work by [19] introduced PhishBench, a benchmarking framework for evaluating phishing detection techniques. Their findings revealed that imbalanced datasets significantly affected performance, with F1 scores dropping between 5.9% and 42%. A major limitation of this study was the lack of real-world testing, making it uncertain how well their framework performs on dynamic datasets, where phishing tactics evolve over time. A research work conducted by [20] proposed a phishing detection model using Adaboost and Support Vector Machine (SVM) classifiers. Their model achieved a high accuracy of 97.61%, with strong performance on AUC, ROC, and F-measure metrics. However, their approach was limited to specific classifier combinations, and the study did not explore alternative ensemble learning techniques that might improve results further. Similar research work carried out by [21] focused on feature selection using Principal Component Analysis (PCA) and classification using an SVM model. They achieved 95.66% accuracy, showing that PCA helped in reducing feature complexity while maintaining performance. However, their study did not explore hyperparameter tuning, which could further enhance the model's effectiveness. Additionally, potential dataset bias may impact generalization to newer phishing attack patterns. John-Otumu et al [22] details the development of a phishing website detection plugin designed to improve the security of online transactions within existing web browsers. The plugin utilizes a novel architecture, trained and tested using a Random Forest classifier. The training data consisted of 9,900 samples, while the testing data comprised 1,100 samples, drawn from a larger dataset of 11,000 data points with 30 features each, sourced from PhishTank and informed by 27 research articles. Python was used for model development, and the frontend, intended for seamless browser integration, was built with Microsoft Visual Studio Code, Jupyter Notebook, Anaconda, HTML/CSS, and JavaScript. The resulting plugin achieved impressive performance, including 96% accuracy, a 0.04 error rate, 97% precision, 99% recall, and a 98% F1-score, surpassing the performance of previously developed models. Finally, research by [23] conducted an extensive study using a dataset containing over one million URLs. They used a combination of lexical analysis and SVM classification, achieving an impressive accuracy of 99.89%. Despite its high accuracy, the study did not address real-time detection or how the model would perform against new and evolving phishing techniques

The related works demonstrated various approaches to phishing detection, utilizing different machine learning models and feature engineering techniques. While some studies focused on traditional classifiers such as Logistic Regression, SVM, and Random Forest, others explored ensemble methods and deep learning techniques like XGBoost and NLP-based phishing detection. Although high accuracy levels were achieved in some cases, limitations such as dataset bias, lack of real-time implementation, and the inability to adapt to evolving phishing strategies were observed. Moreover, most prior works did not optimize feature selection efficiently, leading to potential computational overhead without significant accuracy gains. The proposed research improves upon these limitations by employing an Artificial Neural Network (ANN), which demonstrated superior performance over traditional classifiers. By utilizing the PhiUSIIL Phishing URL Dataset, which is both large and well-structured, and applying Ridge Regression (L2 Regularization) to refine feature selection, the study enhances both accuracy and computational efficiency. The ANN model achieved an impressive 98.58% accuracy, outperforming other tested models such as Random Forest, Naïve Bayes, and XGBoost. The high ROC-AUC score of 0.98 confirms its robustness in distinguishing phishing and legitimate URLs. Unlike many previous works, this research emphasizes scalability and real-world applicability, making the proposed model suitable for integration into intrusion detection systems and anti-phishing tools. Thus, this study presents a highly effective and AI-driven cybersecurity solution that enhances phishing detection with improved accuracy, reliability, and adaptability.

3. METHODOLOGY

3.1 Data Collection

This study used the PhiUSIIL Phishing URL Dataset, which is available on the UCI Machine Learning Repository. The dataset source and description as shown in Table 1, has a size of 54.2MB and contains 235,795 entries with 55 features. These features are a mix of real, categorical, and integer types. Importantly, the dataset has no missing values. Although it is imbalanced, with 134,850 legitimate cases compared to 100,945 phishing cases, its large size and variety make it valuable for research on phishing detection.

Table 1: Dataset Source and Description

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	55
Missing Values	None
Legitimate cases	134,850
Phishing cases	100,945
Comments	Imbalance dataset but very large and diverse

3.2 Feature Extraction and Selection

Ridge Regression is applied to extract and select the most relevant features from the dataset, reducing its dimensionality from 55 to 50 features while preserving critical information.

3.3 Handling Class Imbalance

The Synthetic Minority Oversampling Technique (SMOTE) is used to address class imbalance by generating synthetic samples of the minority class, ensuring balanced representation of legitimate and phishing URLs as shown in Table 2.

Table 2. SMOTE Operations on Initial Dataset

Initial Dataset	**Legitimate cases**	**Phishing cases**
235,795	134,850	100,945
After SMOTE Operations	**Legitimate cases**	**Phishing cases**
269,700	134.850	134.850

Table 1 shows the effect of applying the Synthetic Minority Over-sampling Technique (SMOTE) on the initial dataset. Initially, the dataset contained 235,795 cases, with 134,850 classified as legitimate and 100,945 as phishing. After applying SMOTE, which balances the dataset by generating synthetic samples for the minority class, the number of phishing cases increased to match the legitimate cases at 134,850. As a result, the total dataset size grew to 269,700, ensuring a balanced distribution between legitimate and phishing cases for improved model training.

3.4 Dataset Split Strategy

The dataset is divided into an 80% training set, consisting of 215,700 samples, and a 20% test set, comprising 53,940 samples. This split is used for the development and evaluation of the model, as illustrated in Fig. 1.

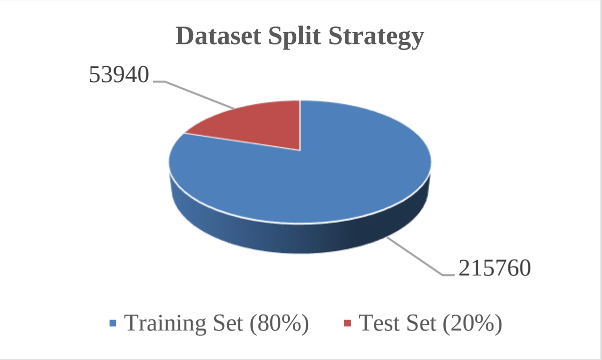

Fig 1. Dataset Split Ratio

3.5 Model Selection and Development

Initially, five machine learning models were chosen as the base models for experimentation: XGBoost, Naive Bayes (NB), Random Forest (RF), K-Nearest Neighbors (K-NN), Logistic Regression (LR), and an Artificial Neural Network (MLP). During this phase, model architectures were established and hyperparameters were defined. Ultimately, the ANN outperformed the others, which led to its exclusive use in the full-scale experiments. The ANN training parameters is shown in Table 3, while the proposed model architecture is illustrated in Fig 2.

Table 3. ANN Training Parameters

Category	Parameter
Data Preprocessing	StandardScaler for feature scaling
	SMOTE for class balancing (random_state=42)
Input Features (X)	High-cardinality hashing (1000 buckets)
	LabelEncoder for categorical encoding
Dataset Splitting	Test size: 20%
	Random state: 42
ANN Architecture	Layers:
	Dense (64 nodes, activation: relu)
	Dropout (rate: 0.3)
	Dense (32 nodes, activation: relu)
	Dropout (rate: 0.3)
	Dense (1 node, activation: sigmoid)
Model Compilation	Optimizer: Adam
	Loss: Binary crossentropy
	Metrics: Accuracy
Training Parameters	Epochs: 50
	Batch size: 32
	Validation split: 20%
	Early stopping: Patience = 5

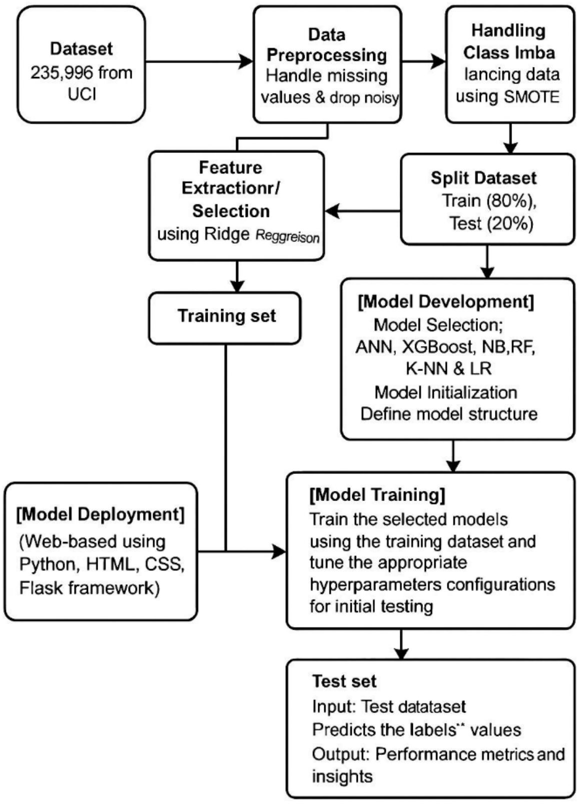
Dataset
235,996 from UCI
Data Preprocessing
Handle missing values & drop noisy
Handling Class Imba lancing data using SMOTE
Feature Extractionr/ Selection using Ridge Reggreison
Split Dataset
Train (80%), Test (20%)
Training set
[Model Development]
Model Selection; ANN, XGBoost, NB,RF, K-NN & LR
Model Initialization
Define model structure
[Model Deployment]
(Web-based using Python, HTML, CSS, Flask framework)
[Model Training]
Train the selected models using the training dataset and tune the appropriate hyperparameters configurations for initial testing
Test set
Input: Test datataset
Predicts the labels" values
Output: Performance metrics and insights

Fig 2. Proposed System Architecture

3.6 Mathematical Notation of the Proposed System Architecture

This section illustrates the mathematical expressions and notations that represent each step in Fig 2.

- **Dataset Representation**
  The dataset contains N = 235,795 from UCI, with two classes:
  - Legitimate cases: N_L = 134,850
  - Phishing cases: N_P = 100,945
- **Data Preprocessing**

Let X = {x_1, x_2,...,x_N}be the feature set and Y={y_1,y_2,...,y_N} be the label set.

Preprocessing involves handling missing values and removing duplicates:

$X' = f_{\text{preprocess}}(X)$ (1)

where $f_{\text{preprocess}}$ includes normalization, missing value imputation, and duplicate removal.

- **Feature Extraction and Selection using Ridge Regression**

Feature extraction using Ridge Regression can be represented as:

$$\beta = \arg \underset{\beta}{min} + \sum_{n=1}^{\infty} (y_i - X_i\, \beta)^2 + \lambda\, ||\beta||^2 \qquad (2)$$

where λ is the regularization parameter.

- **Handling Class Imbalance using SMOTE**

Synthetic Minority Over-sampling Technique (SMOTE) generates synthetic samples to balance the dataset:

$N_P' = N_L$ = 134,850

The new total dataset size becomes:

$N' = N_L + N_P'$ = 269,700

- **Dataset Splitting**

The dataset is divided into training and testing sets:

Train set = 80% × N′ = 0.8 × 269,700

Test set = 20% × N′ = 0.2 × 269,700

- **Model Development and Selection**
  The set of models selected includes:

M = {ANN, XGBoost, NB, RF, K-NN, LR} (3)

Each model m_i ∈ M has its structure and hyperparameters initialized.

- **Model Training**
  Training involves minimizing a loss function L, such as cross-entropy for classification:

$$L = -\sum_{i=1}^{n} yilog(\hat{y}i) + (1 - yi)\, log(1 - \hat{y}i) \qquad (4)$$

- **Model Evaluation**
  Performance is measured using metrics such as accuracy, precision, recall, and F1-score:

$$\text{Accuracy} = \frac{\text{TP + TN}}{\text{TP + FN + FP + TN}} \qquad (5)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP + FP}} \qquad (6)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP + FN}} \qquad (7)$$

$$\text{F1-score} = 2\, x\, \frac{\text{(Precision x Recall)}}{\text{(Precision + Recall)}} \qquad (8)$$

- **Model Deployment**
  The final trained model is deployed using a web-based framework, utilizing Python, Flask, HTML, and CSS. The deployed model takes an input Xnew and predicts ŷ using:

$\hat{y} = f_{\text{model}}(X_{\text{new}})$ (9)

4. RESULTS AND DISCUSSION

The findings of this research are presented and discussed in this section.

4.1 Dataset Splitting

The dataset used in this study was divided into two parts to facilitate model training and evaluation. A total of 80% (215,700 samples) was allocated for training the model, while the remaining 20% (53,940 samples) was reserved for testing its generalization performance. This split, illustrated in Figure 1, ensures a robust evaluation of the model's predictive capability as shown in Table 3 and Figure 5.

4.2 ANN Model Training Performance

The training performance of the proposed Artificial Neural Network (ANN) model over 50 epochs is presented in Figures 3 and 4, illustrating training accuracy and training loss, respectively.

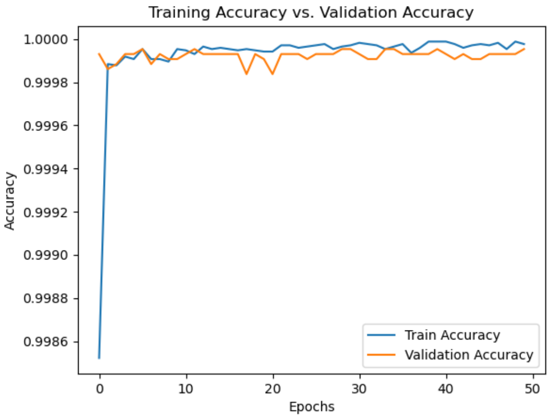

Fig 3. ANN model training accuracy

Figure 3 shows a consistent upward trajectory in both training and validation accuracy, with values plateauing near 1.0. This suggests that the model learned effectively and achieved strong generalization capabilities. The close alignment between training and validation accuracy further indicates minimal overfitting.

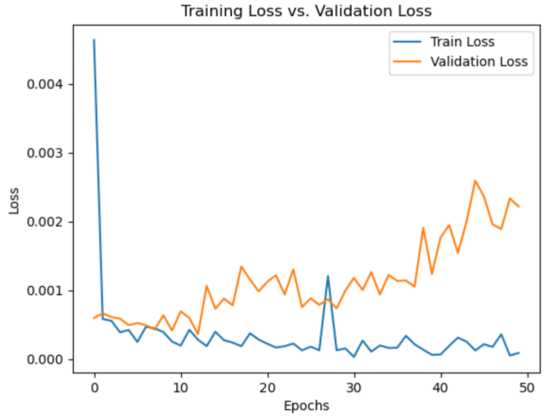

Fig 4. ANN model training Loss

Figure 4 displays the training and validation loss trends. While the training loss steadily decreased and remained low throughout, the validation loss, after an initial drop, showed slight fluctuations and a marginal increase in the later epochs. This behavior, though common, indicates a potential onset of overfitting, albeit within acceptable limits given the consistently low error margins.

4.3 Classification Performance of ANN Model

A detailed analysis of the classification performance of the Multilayer Perceptron (MLP) Artificial Neural Network (ANN) employed in this study is presented in this section. The results are summarized in Table 4 and visualized in Figure 5.

Table 3. ANN Model Classification Results

Accuracy	Precision	Recall	F1-Score
98.58	97.80	97.66	97.65

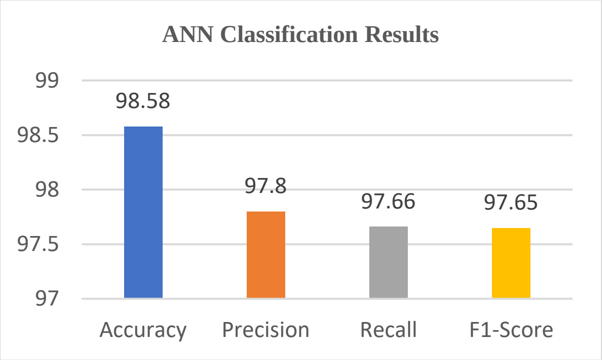

Fig 5. Column Graph Showing ANN Classification Results

Figure 5 presents the ANN classification results, showcasing high performance across key metrics. Accuracy reached 98.58%, indicating excellent classification. Precision (97.8%) and recall (97.66%) demonstrate a strong balance in identifying positive cases, reflected in a near-identical F1-score of 97.65%. These results highlight the model's effectiveness in the classification phishing URL task.

4.4 Comparative Model Evaluation

A comparative analysis of different machine learning models implemented in this study is illustrated in Figure 6. Models evaluated include Naive Bayes (NB), Random Forest (RF), Logistic Regression (LR), K-Nearest Neighbors (K-NN), XGBoost, and the proposed ANN model.

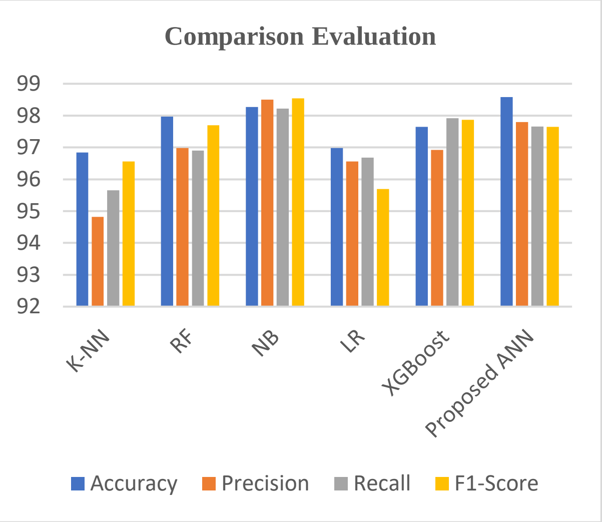

Fig 6. Model comparison evaluation

Figure 6 visually compares the performance of the different machine learning models used for phishing URL detection. Naive Bayes (NB) achieved the highest F1-score (98.54%), followed closely by Random Forest (RF) at 97.7%. The proposed ANN model demonstrated competitive performance, achieving 98.58% accuracy, slightly better than RF, though NB had a higher F1-score. K-Nearest Neighbors (K-NN) and Logistic Regression (LR) performed reasonably well but were less effective than the top performers. XGBoost showed strong performance, similar to RF. In essence, Figure 6 highlights the relative strengths and weaknesses of each model for phishing URL detection, showcasing the effectiveness of the proposed ANN.

5. CONCLUSION

This research successfully developed an Artificial Neural Network - Multi-Layer Perceptron (ANN-MLP) model for enhanced phishing URL detection and classification, addressing a critical challenge in cybersecurity. By evaluating six machine learning models—K-Nearest Neighbors (K-NN), Logistic Regression, Random Forest, Naïve Bayes, XGBoost, and ANN; alongside Ridge Regression for feature selection, the study demonstrated that ANN-MLP outperformed other models in accuracy, robustness, and efficiency. The model achieved high accuracy with low false positives, making it a viable solution for real-world phishing detection. The success of this research was driven by the increasing sophistication of phishing attacks and the limitations of traditional detection systems, highlighting the necessity of advanced machine-learning approaches.

Feature selection played a crucial role in optimizing performance by reducing dimensionality while retaining essential predictive features. This not only enhanced classification accuracy but also improved computational efficiency, making the model scalable for large-scale deployments. Compared to conventional machine learning models, ANN-MLP demonstrated superior learning capabilities, adaptability to new phishing strategies, and a higher generalization ability due to the diverse dataset used in training.

Despite the effectiveness of the proposed ANN-MLP model, there are several promising directions for future research:

- Exploration of Advanced Deep Learning Architectures: While ANN-MLP provided strong results, future research can explore Convolutional Neural Networks (CNNs), Vision Transformers, Autoencoders, and Large Language Models (LLMs) for phishing detection. These models have demonstrated remarkable success in feature extraction and classification in cybersecurity and other domains.
- Ensemble Learning for Improved Robustness: Implementing ensemble techniques such as Stacking, Bagging, and Boosting could further enhance model performance by leveraging the strengths of multiple classifiers. This could provide higher accuracy, better generalization, and improved detection of evolving phishing tactics.
- Integration with Real-Time Threat Intelligence Systems: Deploying the phishing detection model as part of a real-time cybersecurity framework could help organizations detect and mitigate phishing threats dynamically, adapting to new attack vectors in real-time.
- Expanding Dataset Diversity: While this research used a large and diverse dataset, further studies could integrate real-time phishing datasets from threat intelligence sources, allowing models to stay up-to-date with the latest phishing trends.
- Incorporation of Explainable AI (XAI): Future research should incorporate XAI techniques such as SHAP (Shapley Additive Explanations) and LIME (Local Interpretable Model-agnostic Explanations) to improve model interpretability. This would enhance transparency and allow cybersecurity experts to understand how predictions are made, increasing trust and adoption in real-world scenarios.
- Multi-Layer Security Approaches: Combining phishing detection with Natural Language Processing (NLP) for email analysis, heuristic-based detection techniques, and anomaly detection models could create a multi-layer security system capable of identifying phishing attempts across multiple attack vectors.
- Cloud-Based or Edge AI Deployment: Implementing the phishing detection model in cloud environments or edge devices would enable real-time, scalable, and low-latency phishing detection for enterprise security systems and individual users.

This research marks a significant step forward in phishing URL detection, demonstrating the effectiveness of ANN-MLP in identifying phishing threats with high accuracy. The findings provide a strong foundation for future advancements in phishing detection through deep learning, ensemble models, and real-time AI-driven security systems. As phishing techniques continue to evolve, continuous model adaptation and integration with next-generation cybersecurity frameworks will be essential in ensuring robust protection against cyber threats.

6. REFERENCES

[1] Alazaidah, R., Samara, G., Almatarneh, S., Hassan, M., Aljaidi, M., & Mansur, H. (2023). Multi-Label Classification Based on Associations. Applied Sciences, 13(8), 5081.

[2] Al-Khateeb, M., Al-Mousa, M., Al-Sherideh, A., Almajali, D., Asassfeha, M., & Khafajeh, H. (2023). Awareness model for minimizing the effects of social engineering attacks in web applications. International Journal of Data and Network Science, 7(2), 791-800.

[3] Do, N. Q., Selamat, A., Krejcar, O., Herrera-Viedma, E., & Fujita, H. (2022). Deep learning for phishing detection: Taxonomy, current challenges and future directions. Ieee Access, 10, 36429-36463.

[4] Junoh, A. K., AlZoubi, W. A., Alazaidah, R., & Al-luwaici, W. (2020). New features selection method for multi-label classification based on the positive dependencies among labels. Solid State Technology, 63(2s).

[5] Osho, O., Oluyomi, A., Misra, S., Ahuja, R., Damasevicius, R., & Maskeliunas, R. (2019). Comparative evaluation of techniques for detection of phishing URLs. Communications in Computer and Information Science, 1051 CCIS, 385–394. https://doi.org/10.1007/978-3-030-32475-9_28

[6] Gupta, B. B., Arachchilage, N. A., & Psannis, K. E. (2018). Defending against phishing attacks: taxonomy of methods, current issues and future directions. Telecommunication Systems, 67, 247-267.

[7] Alkhalil, Z., Hewage, C., Nawaf, L., & Khan, I. (2021). Phishing attacks: A recent comprehensive study and a new anatomy. Frontiers in Computer Science, 3, 563060.

[8] Alazaidah, R., & Almaiah, M. A. (2021). Associative classification in multi-label classification: An investigative study. Jordanian Journal of Computers and Information Technology, 7(2).

[9] Al-Batah, M. S., Alzyoud, M., Alazaidah, R., Toubat, M., Alzoubi, H., & Olaiyat, A. (2022). Early Prediction of Cervical Cancer Using Machine Learning Techniques. Jordanian Journal of Computers and Information Technology, 8(4).

[10] Al-Sarem, M., Saeed, F., Al-Mekhlafi, Z. G., Mohammed, B. A., Al-Hadhrami, T., Alshammari, M. T., Alreshidi, A., & Alshammari, T. S. (2021). An optimized stacking ensemble model for phishing websites detection. Electronics (Switzerland), 10(11). https://doi.org/10.3390/electronics10111285

[11] Wen, L., & Hughes, M. (2020). Coastal wetland mapping using ensemble learning algorithms: A comparative study of bagging, boosting and stacking techniques. Remote Sensing, 12(10), 1683.

[12] Alazaidah, R., Alshaikh, A., Almousa, M. R., & Samara, G. (2024). Website phishing detection using machine learning techniques. Journal of Intelligent Information Systems, 63(1), 147-161.

[13] Tamal M. A, Islam M. K, Bhuiyan T, Sattar A, and Prince N. U. (2024). Unveiling suspicious phishing attacks: enhancing detection with an optimal feature vectorization algorithm and supervised machine learning. Frontiers in Computer Science. 6:1428013. doi: 10.3389/fcomp.2024.1428013

[14] Onyiagha C. G., Yanwalo, G. F., and Ajimah N. E. (2024). Phishing URL Detection: A Basic Machine Learning Approach, The International Journal of Science & Technoledge, 12(3): 8 – 14. doi:

10.24940/theijst/2024/v12/i3/ST2403-001

[15] Chy, M. K. H. (2024). Securing the web: Machine learning's role in predicting and preventing phishing attacks, International Journal of Science and Research Archive, 2024, 13(01), 1004–101. https://doi.org/10.30574/ijsra.2024.13.1.1770

[16] Nwokoro, I. S., Okesola, O. J., Sambo, M. Q., Oshodin, O. G., Akinfenwa, T. O., Adom-Oduro, Z. K., and Ahmed S. Y. (2024). Phishing Attacks Prevention using Smart Based Artificial Intelligence Algorithms for Cyber Security Awareness, International Journal of Innovative Research in Technology (IJIRT), 11(2): 1228 – 1235. https://doi.org/10.1177/10439862211001628.

[17] Jain, A. K., & Gupta, B. B. (2019). A machine learning based approach for phishing detection using hyperlinks information. Journal of Ambient Intelligence and Humanized Computing, 10, 2015-2028.

[18] Ammara, Z., Khan, H. U., Yousaf, N., Aslam, F., Anjum, A., & Hamdani, M. (2020). Phishing website detection using diverse machine learning algorithms. Journal of Intelligent Information Systems, 57(2), 287-303

[19] Bibi, H., Shah, S. R., Baig, M. M., Sharif, M., Mehmood, M., Akhtar, Z., & Siddique, K. (2024). Phishing website detection using improved multilayered convolutional neural networks. Journal of Computer Science, 20(9), 1069- 1079. https://doi.org/10.3844/jcssp.2024.1069.1079

[20] Subasi, A., & Kremic, E. (2020). Comparison of AdaBoost with multi boosting for phishing website detection. Procedia Computer Science, 168, 272-278.

[21] Rashid, J., Mahmood, T., Nisar, M. W., & Nazir, T. (2020). Phishing detection using machine learning technique. In 2020 first international conference of smart systems and emerging technologies (SMARTTECH) (pp. 43-46). IEEE.

[22] John-Otumu, A, M., Rahman, M. M., and Oko. C. U. (2021). An Efficient Phishing Website Detection Plugin Service for Existing Web Browsers Using Random Forest Classifier. American Journal of Artificial Intelligence, 5(2), 66-75. https://doi.org/10.11648/j.ajai.20210502.13

[23] Abutaha, M., Ababneh, M., Mahmoud, K., & Baddar, S. A. H. (2021, May). URL phishing detection using machine learning techniques based on URLs lexical analysis. In 2021 12th International Conference on Information and Communication Systems (ICICS) (pp. 147-152). IEEE.