Available online at http://icsejournal.com/index.php/JCSE
Journal of Computer Science an Engineering (JCSE)
e-ISSN 2721-0251
Vol. 6, No. 2, August 2025, pp. 59-70

Hyperband-Optimized LightGBM and Ensemble Learning for Web Phishing Detection with SHAP-Based Interpretability

Rizki Wahyudi[1*]

[1]Universitas Amikom Purwokerto, Purwokerto, Central Java and 53127, Indonesia
[1]rizki.key@gmail.com*
* corresponding author

ARTICLE INFO

Article History:
Received July 2, 2025
Revised August 1, 2025
Accepted August 6, 2025

Keywords:
Phishing Detection
Machine Learning
Hyperparameter Tuning
Stacking Ensemble
SHAP Interpretability

Correspondence:
E-mail: rizki.key@gmail.com

ABSTRACT

This study evaluates the performance of three tree boosting algorithms, Random Forest (RF), XGBoost (XGB), and LightGBM (LGBM), in detecting phishing websites using a phishing dataset based on HTML, URLs, and network features. Two hyperparameter optimization strategies were tested: Hyperband search (HalvingRandomSearchCV) and stacking ensemble combining all three models. The evaluation was conducted based on five main metrics: accuracy, precision, recall, F1-score, and AUC-ROC. The results indicate that LightGBM tuned via Hyperband achieved the highest performance (accuracy 0.9724; AUC-ROC 0.9702), followed by ensemble tuned (accuracy 0.9697; AUC-ROC 0.9684). SHAP analysis was used to interpret the contribution of key features in predicting phishing websites. The AUC-ROC difference of 0.0034 points from the XGBoost baseline (0.9668) confirms the effectiveness of Hyperband tuning and stacking ensembles for phishing detection.

1. Introduction

Phishing websites are a frequent and destructive cybersecurity problem. Phishing websites are created by cybercriminals who develop fake websites that look like actual businesses in order to steal user passwords, personal information, and financial data. Due to the fact that the frequency and level of sophistication of these assaults are growing on a global scale, automated detection through machine learning has become an essential tool for phishing prevention. During the detection phase, it is common practice to extract information from URLs, HTML structure, and network metadata. These elements are subsequently utilized in classification models. Machine learning algorithms for phishing detection have been evaluated in a number of previous studies, such as Support Vector Machines, Random Forest, and XGBoost [1], [2]. These algorithms have produced promising results, but they continue to face challenges in terms of performance consistency and hyperparameter optimization[3], [4].

Numerous investigations have utilized ensemble models like Random Forest, XGBoost, and LightGBM because of their capacity to manage nonlinear and extensive datasets[5], [6], [7]. The efficacy of these algorithms significantly relies on the selection and optimization of suitable hyperparameters[8]. Many prior research continue to employ grid search or random search methodologies, which are laborious and inefficient[9], [10], [11]. The Hyperband method presents itself as an adaptive and efficient strategy for hyperparameter tuning, utilizing the notion of successive halving[12], [13], which remains relatively underexplored in the field of phishing detection.

Model interpretability is a key difficulty, particularly for security applications that require transparency in decision-making. This is in addition to the performance elements that are a challenge. Regrettably, the majority of the research that has been done in the past has concentrated on accuracy and other prediction metrics without providing an explanation of how characteristics actually contribute to the classification outcomes [14], [15]. Interpretation methods such as SHAP (SHapley

Additive exPlanations), which are founded on game theory and are capable of delivering both local and global explanations for model predictions [16], [17], [18], have not yet been widely utilized in the context of tuning and ensemble combinations for the purpose of phishing detection.

This study provides a thorough method that closes this gap by (1) evaluating three well-known ensemble models (Random Forest, XGBoost, and LightGBM) as well as a stacking model at baseline; (2) optimizing performance on each base model through Hyperband tuning; (3) Using a meta-learner-based stacking ensemble on the optimized models, stacking allows for combining the strengths of multiple base models by leveraging a higher-level learner, which frequently results in improved generalization and robustness compared to individual models or simple averaging techniques (4) interpreting the best model using SHAP techniques to transparently explain feature contributions. This method not only increases the accuracy of phishing detection but also improves the interpretability and speed of model training.

2. Method

Presented in this section is a description of the research methodology, which consists of the following steps: the preparation and preprocessing of the dataset, the evaluation of the baseline model, the scaling of the features, the tuning of the hyperparameters with Hyperband, the construction of the stacking ensemble, the comparison of the ROC curve, and the interpretation of the SHAP-based model. An illustration of the entire research workflow may be found in Figure 2.

2.1 Dataset and Preprocessing

The dataset, sourced from the UCI Machine Learning Repository [17], contains 11.055 examples of phishing and non-phishing URLs, along with 30 statistical features such as the URL's length, the number of "@" or ".", the presence of HTTPS, the age of the domain, etc. The index column and the IP column were removed. Numeric features are filled with the average value, and categorical features are filled with the mode. The target is encoded in binary (1=phishing, 0=legitimate).

2.1.1 Data Exploration

This section provides a summary of descriptive statistics for all numerical parameters, including the mean, standard deviation, minimum, and maximum values, to elucidate the properties of the data. The class distribution between phishing and genuine categories is obtained using the 5-fold cross-validation approach. Table 1 provides an overview of the dataset structure, showcasing five randomly selected rows alongside numerous critical attributes and the goal label. Additionally, Table 2 and Figure 1 present a summary of numerical statistics and class distribution, offering a first insight into the data patterns and class proportions within the dataset.

Tabel 1. Five Examples of Random Rows, Covering Some Key Features and Target Labels.

Sample	having_IP_Address	URL_Length	SSLfinal_State	age_of_domain
1	-1	-1	0	-1
2	1	-1	-1	1
3	-1	1	1	-1
4	1	-1	1	-1
5	1	-1	1	1

2.1.2 Summary of Class Statistics and Distribution

To analyze the data's characteristics, Table 2 provides summary statistics, including the mean, standard deviation, minimum, and maximum values for five chosen numerical features.

Table 2. Summary Statistics of Some Numerical Features

Fitur	Mean	Std	Min	Max
having_IP_Address	0.3138	0.9495	-1.0	1.0
URL_Length	-0.6332	0.7661	-1.0	1.0
SSLfinal_State	0.2509	0.9119	-1.0	1.0
age_of_domain	0.0612	0.9982	-1.0	1.0
Links_pointing_to_page	0.3440	0.5699	-1.0	1.0

2.1.3 Class Distribution Visualization

Table 2 provides a detailed overview of the data characteristics by presenting summary statistics, including the mean, standard deviation, minimum, and maximum values for five selected numerical features.

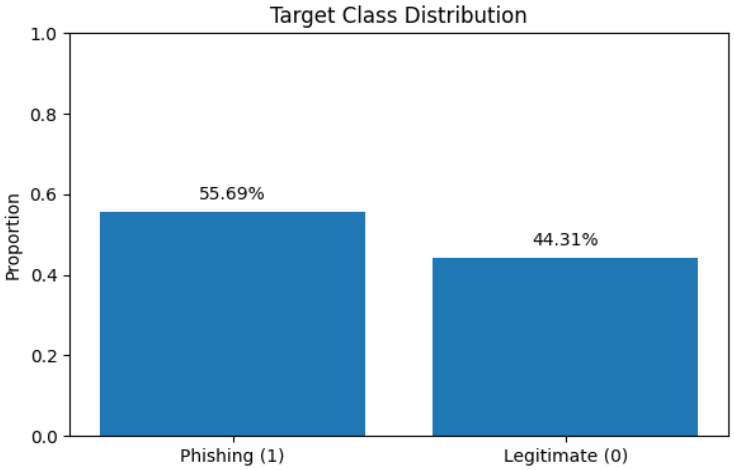

Figure 1. Distribution of Target Classes on the Test Dataset

2.2 Baseline Evaluation

The data was split 80/20 (train/test)[19]. Four baseline models were trained on the original features:

- Random Forest (100 estimator)
- XGBoost (100 estimator, eval_metric='logloss')
- LightGBM (default)
- Stacking default (RF+XGB+LGBM meta-learner LogisticRegression).

Evaluation metric: Accuracy, Precision, Recall, F1-score, AUC-ROC

2.3 Hyperband Tuning

The initial features undergo scaling using StandardScaler[20]. Within every model:

- Random Forest: hyperparameter space: n_estimators, max_depth, max_features, min_samples_split, min_samples_leaf.
- XGB: n_estimators, max_depth, learning_rate, subsample, colsample_bytree.
- LGBM: n_estimators, max_depth, learning_rate, num_leaves, subsample, colsample_bytree.
- An investigation was conducted utilizing HalvingRandomSearchCV (factor=3, cv=5, n_iter≈30). The optimal performance of the model was evaluated using the test set.

2.4 Equations

The three optimized models are integrated via StackingClassifier with a Logistic Regression meta-learner (cv=5). The objective is to harness the synergy of the foundational models[21], [22].

2.5 Evaluation and Analysis

- Confusion Matrix for each variant.
- ROC Curve comparing all models.
- Cross-Validation Variability: 5-fold CV repeated for baseline XGB and tuned LGBM, reporting average and standard deviation of accuracy.
- SHAP Summary Plot for the best LGBM, displaying the top 10 features based on average SHAP values.

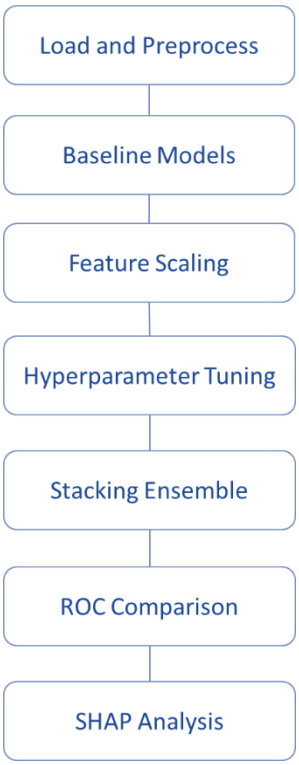

Figure 2. Reesearch Methodology

3. Results and Discussion

3.1 Baseline Model Evaluation

The preliminary phase of this experiment involved assessing the baseline performance of four classification models: Random Forest (RF), XGBoost (XGB), LightGBM (LGBM), and the Stacking ensemble model. All models underwent evaluation utilizing raw features, without any scaling or tuning processes, to deliver an initial overview of the effectiveness of each algorithm on the dataset employed. The assessment was carried out utilizing five primary metrics: accuracy, precision, recall, F1-score, and AUC-ROC, to guarantee that the model's performance is not only elevated overall but also equitable in identifying both minority and majority classes.

According to the baseline results presented in Table 3, all models demonstrate highly competitive performance, achieving accuracy levels exceeding 96%. The stacking model exhibits the most superior overall performance, achieving an accuracy of 96.88%, a recall of 98.17%, and an F1-score of 97.28%. In comparison, XGBoost follows closely with an accuracy of 96.83%, a precision of 96.62%, and an F1-score of 97.23%. The Random Forest model demonstrated impressive performance, achieving a recall of 98.01%. However, its AUC-ROC is marginally lower than that of XGBoost and Stacking. LightGBM recorded the lowest performance among the four models, but the difference is relatively small, with an accuracy of 96.47% and an AUC-ROC of 96.28%.

Upon examining each metric, the stacking model demonstrates the most favorable equilibrium between sensitivity and precision. The elevated recall value demonstrates a strong capacity to identify nearly all occurrences of the positive class (phishing), whereas the consistent precision reflects a low rate of false positive predictions. The AUC-ROC value for stacking is 0.9668, matching that of XGBoost, which suggests that both models exhibit equivalent effectiveness in differentiating between phishing and legitimate classes.

In summary, the baseline results demonstrate that while each model exhibits specific advantages (for instance, RF excels in recall, XGB in precision, and Stacking in metric balance), the Stacking model emerges as the most promising option at this preliminary stage prior to any tuning efforts. This establishes a solid basis for advancing performance enhancement via normalization and hyperparameter optimization.

Table 3. Baseline Model Evaluation using Raw Features

Model	Accuracy	Precision	Recall	F1-score	AUC-ROC
Random Forest (RF)	0.9665	0.9617	**0.9801**	0.9708	0.9644
XGBoost (XGB)	0.9683	**0.9662**	0.9785	0.9723	0.9668
LightGBM (LGBM)	0.9647	0.9616	0.9769	0.9692	0.9628
Stacking	**0.9688**	0.9640	**0.9817**	**0.9728**	**0.9668**

3.2 Feature Scaling and Model Tuning

The feature scaling process is carried out to ensure that all numerical features are on a uniform scale before being used in the model training process, especially for scale-sensitive algorithms like XGBoost and LightGBM. Standardization using StandardScaler is a crucial step to prevent features with large values from dominating the model and to accelerate convergence during the training and tuning process.

To improve classification performance, a hyperparameter tuning process was conducted using the HalvingRandomSearchCV approach, which is an efficient version of staged random search that reduces the number of candidates as cross-validation accuracy increases. This process is carried out independently for each model: Random Forest, XGBoost, and LightGBM. The tuning results in models with optimal parameter combinations, which are then used as base learners in the stacking model.

After tuning, the performance of the three base models showed significant improvement. LightGBM recorded the most significant improvement, with accuracy rising from 96.47% to 97.24%, recall increasing from 97.69% to 98.65%, and F1-score from 96.92% to 97.60%. The XGBoost model also saw an improvement in F1-score from 97.23% to 96.84% and AUC-ROC from 0.9668 to 0.9620, although it slightly decreased marginally in some metrics. Meanwhile, Random Forest showed consistently high recall but a slight decrease in AUC-ROC from 0.9644 to 0.9579, likely due to overfitting with certain parameter combinations. The Tuned Stacking Model, which combines the three optimized models, recorded an accuracy of 96.97%, a precision of 96.92%, a recall of 97.77%, and an F1-score of 97.34%, generally outperforming all individual models and the baseline.

3.3 Performance Comparison Across Phases

A systematic comparison of model performance before and after tuning was performed to evaluate the impact of the optimization process on classification quality. The evaluation was conducted using five key metrics: accuracy, precision, recall, F1-score, and AUC-ROC. To support this analysis, the results are presented in the form of bar graphs depicting the performance of each model in two phases: baseline and after tuning using HalvingRandomSearchCV.

a. Accuracy Comparison

The accuracy graph in Figure 3. shows that hyperparameter tuning has a positive impact on the LightGBM and Stacking models. LightGBM saw an accuracy increase from 96.47% to 97.24%, while the stacking model saw an increase from 96.88% to 96.97%. The XGBoost and Random Forest models tended to experience slight fluctuations. This graph shows that the tuning process can help certain models achieve greater stability in classification.

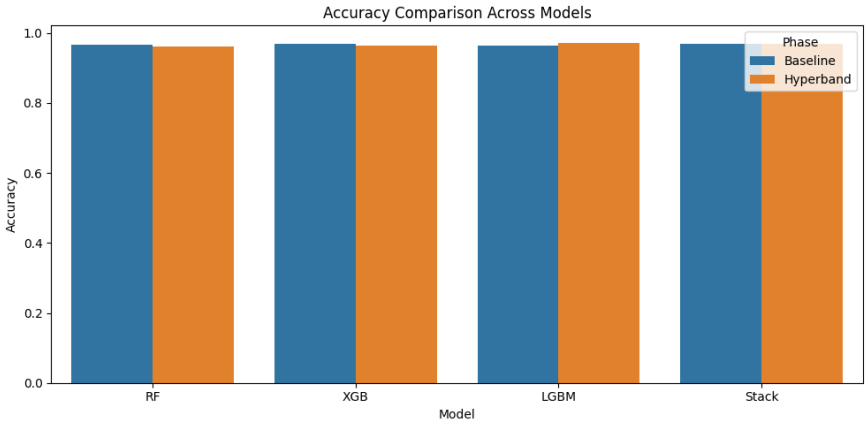

Figure 3. Accuracy Comparison Across Models

b. F1-score Comparison

The F1-score metric in Figure 4. also showed a considerable improvement after tuning, with LightGBM recording a large rise from 96.92% to 97.60%. This indicates that the balance between precision and recall has been improved. After adjustment, stacking was able to keep up its consistently excellent performance, achieving an F1 score of 97.34%. By confirming that the tuning process helps maintain optimal performance in categorization scenarios that need both sensitivity and precision simultaneously, this improvement demonstrates that the tuning procedure is effective.

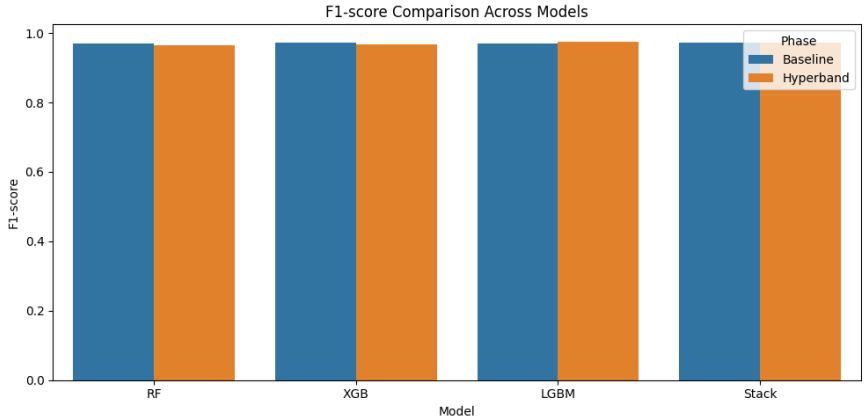

Figure 4. F1-score Comparison Across Models

c. Precision Comparison

The F1-score metric in Figure 5. also showed a considerable improvement after tuning, with LightGBM recording a large rise from 96.92% to 97.60%. This suggests an improvement in the balance between precision and recall. After adjustment, stacking was able to keep up its consistently excellent performance, achieving an F1 score of 97.34%. By confirming that the tuning process helps maintain optimal performance in categorization scenarios that need both sensitivity and precision simultaneously, this improvement demonstrates that the tuning procedure is effective.

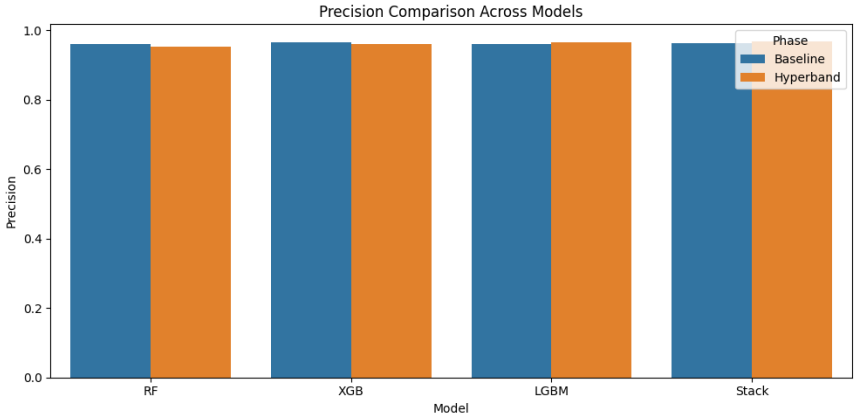

Figure 5. Precision Comparison Across Models

d. Recall Comparison

The recall values in Figure 6. show that the stacking and LightGBM models have the highest ability to detect positive classes. After tuning, the stacking recall was recorded at 97.77%, while LightGBM increased from 97.69% to 98.64%, the highest value among all models. This indicates that these models are very effective in minimizing type II errors (false negatives), which is crucial in applications such as phishing detection or security systems.

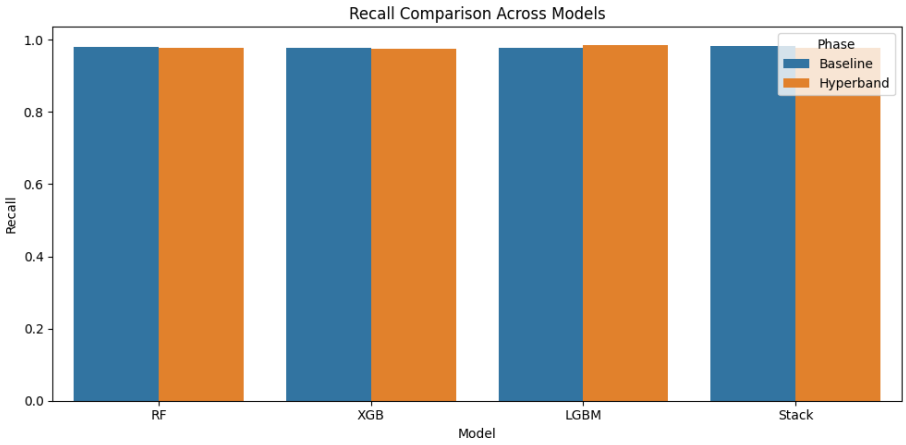

Figure 6. Recall Comparison Across Models

3.4 ROC Curve Analysis

Figure 7 shows the ROC (Receiver Operating Characteristic) curve above, providing a visual comparison between all tested classification models, both at the baseline stage and after hyperparameter tuning using the Hyperband approach. This curve represents the relationship between the True Positive Rate (TPR) and False Positive Rate (FPR), which reflects the extent to which the model can correctly detect positive classes without producing too many misclassifications. The closer the curve is to the upper left corner of the graph, the better the model's performance in terms of high sensitivity and strong specificity.

The Area Under the ROC Curve (AUC-ROC) values show that the LightGBM model tuned using Hyperband performed best with an AUC of 0.9702, followed by the Stacking Tuned model with an AUC of 0.9684. The Stacking Tuned model also demonstrated very balanced classification performance, with an accuracy of 96.97%, a precision of 96.92%, a recall of 97.77%, and an F1-score of 97.34%. The result makes the ensemble model the most optimal candidate because it offers the best trade-off between a high positive detection rate and a low misclassification rate. Its curve, which is very close to the upper left edge of the ROC graph, indicates that the model is capable of maintaining very high detection performance even at minimal FPR levels.

Meanwhile, the XGBoost model at the baseline stage, despite having a fairly satisfactory AUC value (0.9668), exhibits visually weaker performance on the ROC graph. Its curve is not as sharp as the other models, especially in the early low FPR segments, indicating that the model struggles to accurately distinguish classes when small error rates are taken into account. The result indicates that despite high aggregate metrics like AUC, the probability distribution generated by the baseline XGBoost is less focused, causing the ROC curve to deviate from the ideal shape.

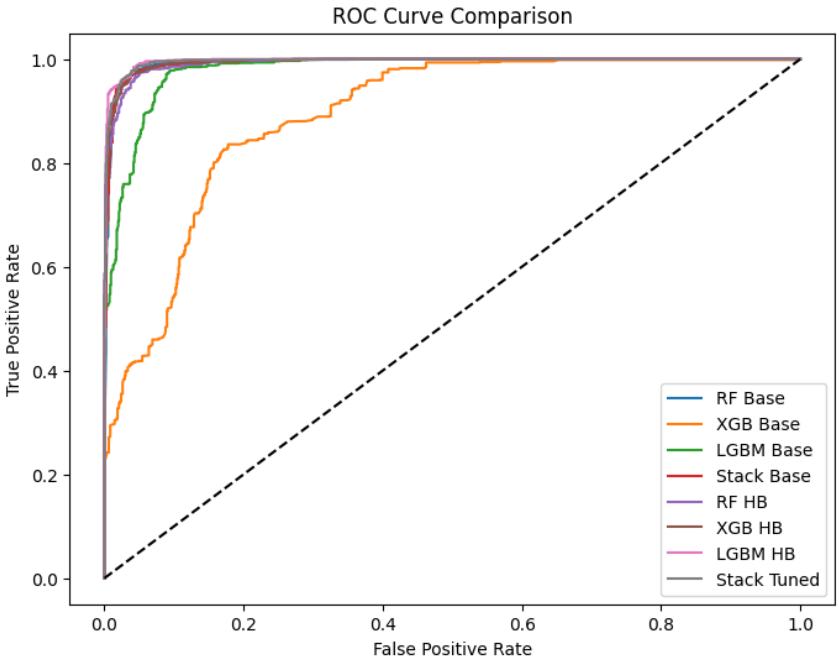

Figure 7. ROC Curve Comparison all Models

3.5 Model Explainability using SHAP

Figure 8 shows the SHAP summary plot visualization showing the contribution of each feature to the prediction output of the best model, namely LightGBM (LGBM), optimized with Hyperband tuning. SHAP (Shapley Additive Explanations) is a game theory-based interpretability method that consistently and locally identifies the individual contribution of each feature to model predictions. The use of SHAP is important because it provides transparency over black-box models like ensemble methods (including LGBM) and helps identify the features that most influence model decisions quantitatively and visually.

In the plot, the horizontal axis shows the SHAP value, which indicates the impact of a feature on the classification probability (positive or negative), while the color of the dot indicates the original value of the feature (blue = low, red = high). Each dot represents a single observation in the testing dataset. The features at the top of the graph are those that generally contribute the most to model predictions. These results show that the URL_of_Anchor, SSLfinal_State, Prefix_Suffix, and Links_in_tags features have the most significant contributions to the model's prediction output. For example, high values for the URL_of_Anchor feature (marked in red) tend to push the model's output in a positive direction (higher risk), while low values (blue) lead to negative predictions (higher risk). A similar pattern is also seen for the Abnormal_URL, Request_URL, and SFH features, which also have a substantial influence on classification.

Thus, this SHAP analysis not only strengthens our understanding of how the model makes decisions but also opens up opportunities for model simplification, feature selection, and increased user confidence in the classification system particularly in the context of phishing detection. The identified key features can also form the basis for rule-based cyber defense systems or serve as a focus for future data collection.

 e-ISSN 2721-0251

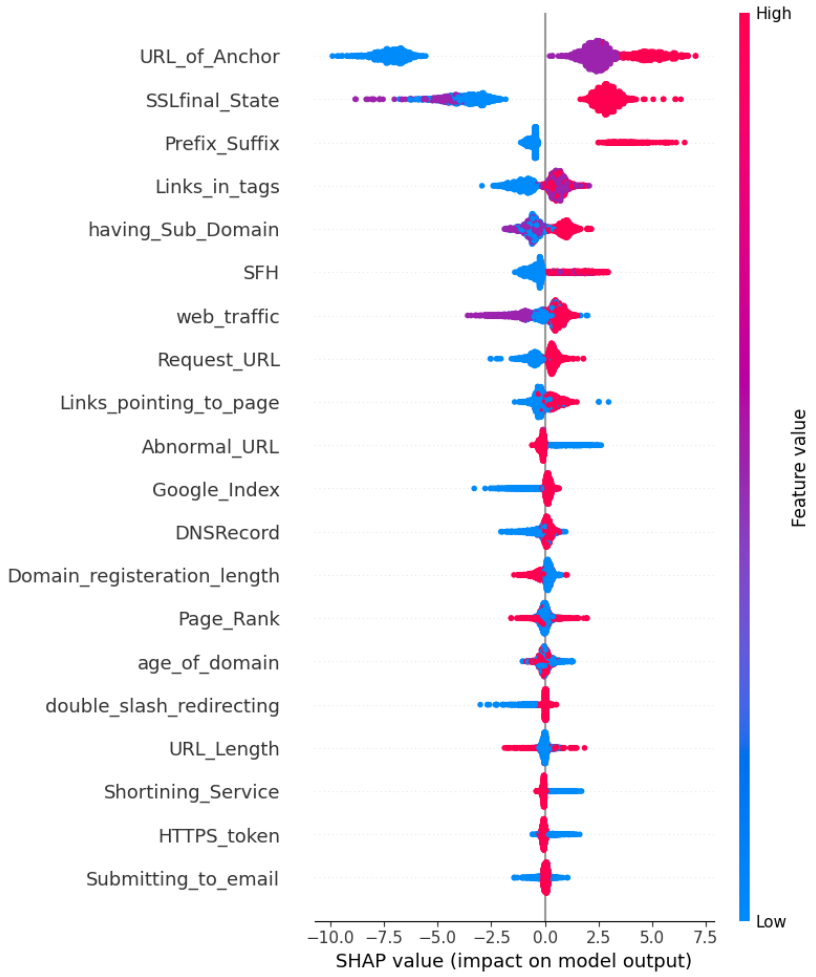

Figure 8. Visualisation SHAP summary plot

3.6 Confusion Matrix Analysis

The results of the analysis on the classification models are shown through confusion matrices for various methods, including Random Forest (RF), Extreme Gradient Boosting (XGB), Light Gradient Boosting Machine (LGBM), and the stacking technique. For the baseline, the RF model showed 907 true negatives and 1230 true positives, while XGB had 913 true negatives and 1228 true positives, and LGBM recorded 907 true negatives and 1226 true positives. The stacking method showed similar results with 910 true negatives and 1232 true positives. In the hyperband test, the RF model showed 896 true negatives and 1228 true positives, while XGB got 907 true negatives and 1224 true positives, and LGBM recorded 912 true negatives with 1238 true positives. The stacking model in the hyperband showed 917 true negatives and 1227 true positives, indicating that, despite variations in performance, all methods provide quite good classification results. This analysis illustrates the strength of different algorithms in identifying target categories with relatively high effectiveness.

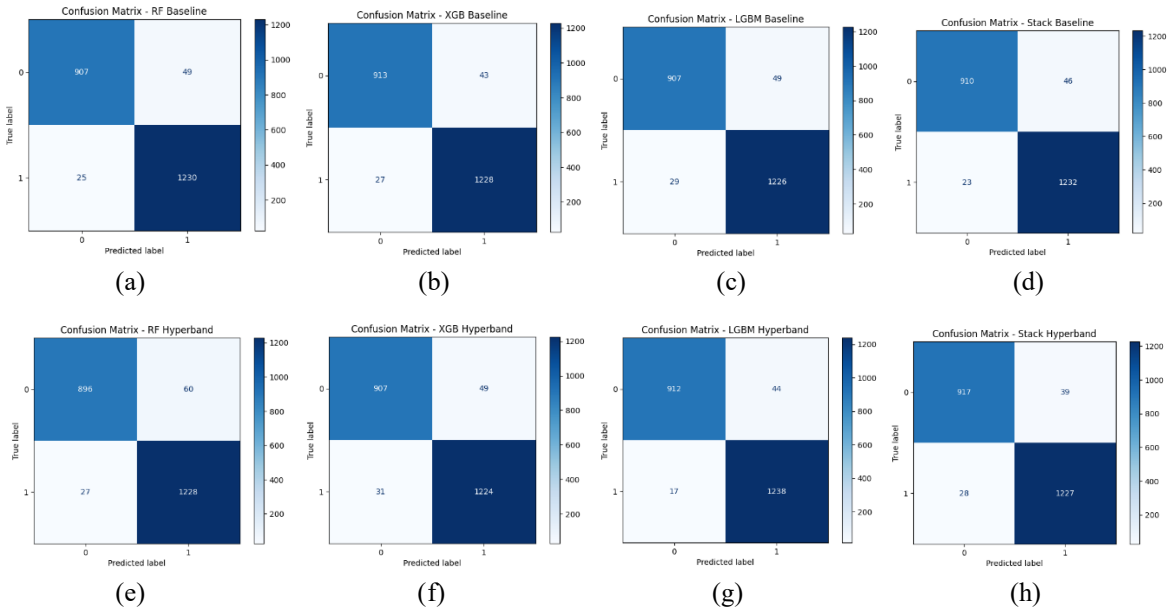

Figure 9. Confusion Matrix baseline and Hyperband all Model (a) Confussion matric RF Baseline, (b) Confussion matric XGB Baseline, (c) Confussion matric LGBM Baseline, (d) Confussion matric Stack Baseline, (e) Confussion matric RF Hyperband, (f) Confussion matric XGB Hyperband (g) Confussion matric LGBM Hyperband (h) Confussion matric Stack Hyperband.

3.7 Summary of Findings

This experiment yielded several important findings that offer an in-depth assessment of the performance of various classification models, both at the initial (baseline) stage and after hyperparameter optimization using HalvingRandomSearchCV. The Stacking Ensemble model consistently performed best, both at the baseline and after tuning. In the baseline phase, stacking achieved an accuracy of 96.88%, a recall of 98.17%, and an F1-score of 97.28%, outperforming the other individual models. Performance improved after tuning, reaching 96.97% accuracy, 97.77% recall, 97.34% F1-score, and 96.84% AUC-ROC value. Although the increase is relatively small, it demonstrates the stability and reliability of the stacking model in maintaining classification quality.

The individual models also responded positively to the tuning process. LightGBM recorded the most significant improvement, from 96.47% accuracy to 97.24%, and from 96.92% to 97.60% in the F1-score, indicating that the model is highly responsive to parameter adjustments. In contrast, Random Forest and XGBoost experienced performance fluctuations, with Random Forest showing a slight decrease in accuracy and AUC while maintaining high recall. This suggests that tuning does not always guarantee improved performance once the model is already near-optimal.

Overall, the tuning process proved to have a positive impact on model performance, particularly for LightGBM and the Stacking ensemble. The application of an ensembling approach that combines the strengths of each individual model yielded more balanced and stable classification results. The implications of these results are particularly important in real-world applications such as phishing detection or network security classification systems, where high recall and minimal misclassification are key priorities. With its ability to capture a significant proportion of incidents without significantly increasing the false positive rate, the Tuned Stacking model is reliable in error-critical classification systems.

This research has a number of shortcomings, although it achieved a high level of accuracy and interpretability on benchmark datasets. In the first place, the model was trained and tested on static datasets (UCI and public repositories), which might not be an accurate representation of the ever-changing nature of phishing techniques in real-world circumstances. Furthermore, the SHAP-based interpretability, although it is insightful, results in additional processing cost during post-analysis,

which may not be feasible in real-time detection systems. Due to the fact that its performance on different phishing datasets or multilingual URLs has not yet been investigated, the generalizability of the proposed model is also something that will require additional examination. These constraints could be addressed in subsequent work by verifying the approach on streaming data, testing it across various platforms, and including real-time model adaptation strategies.

4. Conclusion

This research successfully evaluated and compared the performance of several classification algorithms, namely Random Forest, XGBoost, LightGBM, and the Stacking ensemble model, both under baseline conditions and after hyperparameter tuning using HalvingRandomSearchCV. The evaluation results showed that the stacking ensemble approach consistently provided the best classification performance, with an accuracy of 96.97%, a recall of 97.77%, and an AUC-ROC of 96.84% after tuning. This model not only excelled in global accuracy but also demonstrated an optimal balance between precision and sensitivity, making it a strong candidate for application in real-world applications requiring reliable classification, such as phishing detection systems.The feature scaling and tuning processes proved to have a positive impact, particularly on the LightGBM model, which demonstrated significant improvements in almost all evaluation metrics. Further analysis using SHAP (SHapley Additive exPlanations) demonstrated that model interpretability can also be strengthened by identifying the features most influential in predictive decisions. Thus, the model is not only precise but also understandable and transparently auditable.Although the results are very promising, there are several possible future development directions. First, the integration of data balancing methods such as SMOTE or ADASYN can be explored to address imbalanced class distributions, which are common in real-world datasets. Second, exploring deep learning-based model architectures, such as LSTM or Transformer for sequential data, can be an alternative for cases with temporal patterns. Third, implementing and testing the model in a production environment or real-time pipeline is necessary to practically verify the system's scalability and efficiency. Finally, a combination of other explainable AI techniques, such as LIME or counterfactual explanations, can enrich the model's understanding and increase end-user confidence in the developed system.

References

[1] P. Singh, T. Hasija, and K. R. Ramkumar, “Integrated Machine Learning Approach to Phishing Detection: Comparing SVM, Random Forest, and XGBoost Models,” in *2024 4th International Conference on Technological Advancements in Computational Sciences (ICTACS)*, IEEE, Nov. 2024, pp. 739–744. doi: 10.1109/ICTACS62700.2024.10840493.

[2] N. F. Almujahid, M. A. Haq, and M. Alshehri, “Comparative evaluation of machine learning algorithms for phishing site detection,” *PeerJ Comput Sci*, vol. 10, p. e2131, Jun. 2024, doi: 10.7717/peerj-cs.2131.

[3] N. Q. Do, A. Selamat, O. Krejcar, E. Herrera-Viedma, and H. Fujita, “Deep Learning for Phishing Detection: Taxonomy, Current Challenges and Future Directions,” *IEEE Access*, vol. 10, pp. 36429–36463, 2022, doi: 10.1109/ACCESS.2022.3151903.

[4] M. Almousa, T. Zhang, A. Sarrafzadeh, and M. Anwar, “Phishing website detection: How effective are deep <scp>learning-based</scp> models and hyperparameter optimization?,” *SECURITY AND PRIVACY*, vol. 5, no. 6, Nov. 2022, doi: 10.1002/spy2.256.

[5] K. Ileri, “Comparative analysis of CatBoost, LightGBM, XGBoost, RF, and DT methods optimised with PSO to estimate the number of k-barriers for intrusion detection in wireless sensor networks,” *International Journal of Machine Learning and Cybernetics*, May 2025, doi: 10.1007/s13042-025-02654-5.

[6] S. Demir and E. K. Sahin, “An investigation of feature selection methods for soil liquefaction prediction based on tree-based ensemble algorithms using AdaBoost, gradient boosting, and XGBoost,” *Neural Comput Appl*, vol. 35, no. 4, pp. 3173–3190, Feb. 2023, doi: 10.1007/s00521-022-07856-4.

[7] T. Kavzoglu and A. Teke, “Predictive Performances of Ensemble Machine Learning Algorithms in Landslide Susceptibility Mapping Using Random Forest, Extreme Gradient Boosting (XGBoost) and Natural Gradient Boosting (NGBoost),” *Arab J Sci Eng*, vol. 47, no. 6, pp. 7367–7385, Jun. 2022, doi: 10.1007/s13369-022-06560-8.

[8] I. D. Mienye and Y. Sun, "A Survey of Ensemble Learning: Concepts, Algorithms, Applications, and Prospects," *IEEE Access*, vol. 10, pp. 99129–99149, 2022, doi: 10.1109/ACCESS.2022.3207287.

[9] V. Selvaraj and I. Vairavasundaram, "A Bayesian optimized machine learning approach for accurate state of charge estimation of lithium ion batteries used for electric vehicle application," *J Energy Storage*, vol. 86, p. 111321, May 2024, doi: 10.1016/j.est.2024.111321.

[10] N. Subaşı, "Comprehensive Analysis of Grid and Randomized Search on Dataset Performance," *European Journal of Engineering and Applied Sciences*, vol. 7, no. 2, pp. 77–83, Dec. 2024, doi: 10.55581/ejeas.1581494.

[11] B. K. Dedeturk and B. Akay, "A parallel hybrid approach integrating clonal selection with artificial bee colony for logistic regression in spam email detection," *Neural Comput Appl*, Dec. 2024, doi: 10.1007/s00521-024-10505-7.

[12] J. Wilson, S. Chaudhury, and B. Lall, "Successive Halving Based Online Ensemble Selection for Concept-Drift Adaptation," *IEEE Transactions on Artificial Intelligence*, pp. 1–15, 2025, doi: 10.1109/TAI.2025.3578305.

[13] F. Hutter, L. Kotthoff, and J. Vanschoren, Eds., *Automated Machine Learning*. Cham: Springer International Publishing, 2019. doi: 10.1007/978-3-030-05318-5.

[14] Arunraju Chinnaraju, "Explainable AI (XAI) for trustworthy and transparent decision-making: A theoretical framework for AI interpretability," *World Journal of Advanced Engineering Technology and Sciences*, vol. 14, no. 3, pp. 170–207, Mar. 2025, doi: 10.30574/wjaets.2025.14.3.0106.

[15] J. R, "Transparency in AI Decision Making: A Survey of Explainable AI Methods and Applications," *Advances in Robotic Technology*, vol. 2, no. 1, pp. 1–10, Jan. 2024, doi: 10.23880/art-16000110.

[16] L. Merrick and A. Taly, "The Explanation Game: Explaining Machine Learning Models Using Shapley Values," 2020, pp. 17–38. doi: 10.1007/978-3-030-57321-8_2.

[17] Z. Li, "GeoShapley: A Game Theory Approach to Measuring Spatial Effects in Machine Learning Models," *Ann Am Assoc Geogr*, vol. 114, no. 7, pp. 1365–1385, Aug. 2024, doi: 10.1080/24694452.2024.2350982.

[18] M. Li, H. Sun, Y. Huang, and H. Chen, "Shapley value: from cooperative game to explainable artificial intelligence," *Autonomous Intelligent Systems*, vol. 4, no. 1, p. 2, Feb. 2024, doi: 10.1007/s43684-023-00060-8.

[19] F. Yahya *et al.*, "Detection of Phising Websites using Machine Learning Approaches," in *2021 International Conference on Data Science and Its Applications (ICoDSA)*, IEEE, Oct. 2021, pp. 40–47. doi: 10.1109/ICoDSA53588.2021.9617482.

[20] K. Barik, S. Misra, and R. Mohan, "Web-based phishing URL detection model using deep learning optimization techniques," *Int J Data Sci Anal*, Feb. 2025, doi: 10.1007/s41060-025-00728-9.

[21] K. Kanathey, VishwaGupta, and F. Imam, "An Enhanced and Optimized Stacking Ensemble Framework for Phishing URLs Detection," in *2025 4th OPJU International Technology Conference (OTCON) on Smart Computing for Innovation and Advancement in Industry 5.0*, IEEE, Apr. 2025, pp. 1–6. doi: 10.1109/OTCON65728.2025.11070371.

[22] M. Adnan, M. O. Imam, M. F. Javed, and I. Murtza, "Improving spam email classification accuracy using ensemble techniques: a stacking approach," *Int J Inf Secur*, vol. 23, no. 1, pp. 505–517, Feb. 2024, doi: 10.1007/s10207-023-00756-1.