



Comparative Study of Machine Learning Algorithms Using Bagging and XGBoost Techniques for Breast Cancer Classification

Rully Pramudita¹, Dwi Ismiyana Putri², Bambang Kriswantara³, Vina Zahrotun Nazah⁴, Rahmat Budiarto⁵

^{1,4,5}Dept. of Informatics Engineering, Faculty of Informatics and Design, Bina Insani University, Indonesia

²Dept. of Software Engineering, Faculty of Informatics and Design, Bina Insani University, Bekasi, Indonesia

³Dept. of Informatics Management, Faculty of Informatics and Design, Bina Insani University, Bekasi, Indonesia

⁵Dept. of Computer Science, Faculty of Computing and Information, Al-Baha University, Al Baha, Saudi Arabia

*rullypramudita@binainsani.ac.id

Abstract

Machine learning (ML) has become an important data-driven approach for classification and prediction, including applications in medical diagnosis. Ensemble methods can improve classifier performance by combining complementary learning mechanisms. However, systematic evidence on the sequential use of Bagging and XGBoost across different classifier architectures remains limited. This study develops a staged ensemble framework in which five classifiers—Support Vector Machine (SVM), Neural Network (NN), Logistic Regression (LR), Decision Tree (DT), and K-Nearest Neighbours (KNN)—are first optimized through Bagging and subsequently enhanced using XGBoost. The experiments were conducted on the Breast Cancer Wisconsin (Diagnostic) dataset under a consistent 70:30 train-test protocol. Performance was assessed using accuracy, confusion matrices, ROC curves, and Area Under the Curve (AUC), while repeated experiments were used to examine statistical significance. The results show that the staged Bagging-XGBoost approach improves both predictive accuracy and class discrimination across the evaluated classifier types. Neural Network achieved the largest improvement, with mean accuracy increasing from 93.1% to 97.0% across repeated experiments. The findings demonstrate that the sequential framework can improve non-tree-based as well as tree-based classifiers, providing empirical evidence for broader use of staged ensemble integration in breast cancer classification.

Keywords: bagging; breast cancer diagnosis; ensemble learning; machine learning; XGBoost

How to Cite: R. Pramudita, D. I. Putri, B. Krisdiwantara, V. Z. Nazah, and R. Budiarto, "Comparative Study of Machine Learning Algorithms Using Bagging and XGBoost Techniques for Breast Cancer Classification", *J. RESTI (Rekayasa Sist. Teknol. Inf.)*, vol. 10, no. 4, pp. 1174–1182, Aug. 2026, doi: [10.29207/resti.v10i4.7216](https://doi.org/10.29207/resti.v10i4.7216).

Permalink/DOI: <https://doi.org/10.29207/resti.v10i4.7216>

Received: September 29, 2025

Accepted: June 21, 2026

Available Online: August 25, 2026

*This is an open-access article under the [CC BY 4.0 License](https://creativecommons.org/licenses/by/4.0/)
Published by [Ikatan Ahli Informatika Indonesia](https://www.iaii.or.id/)*

1. Introduction

Machine Learning (ML) is currently one of the most popular approaches in modern data processing. This technology enables systems to learn from patterns in data, which can then be used to make predictions and classifications automatically. The important role of ML is clearly seen in classification tasks, which is the process of determining the label or category of data based on its features. This approach has been widely applied in various fields, ranging from healthcare, finance, education, automotive, energy, environment, to cybersecurity. The integration of machine learning techniques has emerged as a promising approach, aiding healthcare professionals in the complex task of

breast cancer diagnosis[1]. However, the performance of each algorithm is highly dependent on the type of data, the level of complexity of the problem, and the parameter configuration. To overcome the limitations of a single model, ensemble learning is presented as a solution that can reduce variance and bias, as well as improve generalization capabilities. The two most widely used approaches are Bagging and Boosting, with Extreme Gradient Boosting (XGBoost) being one of the most prominent modern boosting variants due to its efficiency and accuracy. Comparative analysis shows that the XGBoost model established has better effects, and its recall, precision, accuracy, and F1-score [2].

In the field of healthcare, ensemble learning is widely used for disease diagnosis and prediction. Admojo and Poetro reported relatively stable Bagging performance across the validation folds, indicating that the method provided consistent results for breast cancer classification. Nasab et al. [4] investigated risk factors for primary invasive breast cancer in the Iranian population using several machine learning models, including Random Forest (RF), Neural Networks (NN), Bagged CART, and XGBoost. Their XGBoost Bagged CART model achieved an AUC of 0.89, whereas the standalone XGBoost model obtained an AUC of 0.78. Pramudita et al. [5] evaluated SVM, NN, LR, DT, and KNN for breast cancer diagnosis and reported that the boosted Logistic Regression model achieved 97.1% accuracy with an AUC of 99.7%. Pramudita et al. [6] further reported accuracy gains of 0.6% for Decision Tree and 1.3% for KNN following the application of Bagging. These findings suggest that ensemble-based approaches have strong potential for improving predictive reliability in medical diagnosis tasks. Zizaan et al. [7] reported that Bagging and Boosting ensembles generally achieved better performance than their corresponding base learners and individual models. The highest-performing configuration was Gradient Boosting with DenseNet201 as the feature extractor and 200 trees, which achieved 85.50% accuracy. Rahman et al. [8] evaluated ten machine learning algorithms for breast cancer prediction and found that MLP-LR achieved the highest accuracy among the individual models at 98.25%, followed by SVM at 97.08%. Among the ensemble approaches, the Voting classifier obtained the highest accuracy of 99.42%. Dutta et al. [9] developed an interpretable breast cancer classification approach based on XGBoost, SHAP, and LIME. Among the evaluated models, XGBoost achieved the highest accuracy, reaching 99.30%. Le et al. [10] developed XGBoost classifiers incorporating LDA dimensionality reduction for breast cancer prediction. Their approach achieved a root mean squared error of 0.115, corresponding to an error level at least 2.6% lower than those obtained by the other evaluated methods. Prasetyo [11] evaluated several ensemble techniques for breast cancer diagnosis and identified XGBoost as the best-performing approach, achieving an accuracy of 94.737%. The reported performance was associated with the scalability, computational efficiency, robustness, gradient-boosting mechanism, and regularization characteristics of XGBoost. Avlani [12] evaluated Bagging and Boosting algorithms using accuracy and precision to assess model performance. The study also incorporated explainable AI techniques, including SHAP and LIME, to provide greater insight into feature importance and individual predictions. The combination of machine learning classifiers, hyperparameter tuning, and explainability techniques was presented as a potential approach for developing more interpretable breast cancer screening systems.

Asselman et al. [13] developed a new ensemble learning-based Performance Factors Analysis (PFA)

approach, and XGBoost proved to be much more accurate than the original PFA. Deng et al. [14] combined Bagging and XGBoost for forecasting electricity load during extreme weather, resulting in more reliable predictions and smaller errors. Lukas et al. [15] compared Bagging and AdaBoost on multi-class data and found Bagging to be slightly superior, although not significantly. Malek et al. [16] examined class imbalance in water quality data, with Gradient Boosting without resampling producing the best performance, while Random Forest with random under-sampling excelled in balanced accuracy. Niazkar et al. [17] found in their literature review that 74% of publications reported XGBoost or XGBoost-based hybrids as the best models in water resource engineering. Li et al. [18] also showed that XGBoost was able to optimize PM2.5 and O3 pollutant simulations with more accurate results than WRF-Chem models and other algorithms. Plaia et al. [19] expanded decision trees and ensembles for ranking data, with the boosting method remaining superior to bagging. Zhao et al. [20] proposed new methods called Boosting Fine-tuning (BF) and Bagging and Boosting Fine-tuning (BBF), which proved to be flexible and effective on 46 real datasets. Demir and Sahin [21] combined feature selection (Boruta, RFE, Stepwise Regression) with tree-based ensembles, including XGBoost, and found that the combination of XGBoost + Boruta provided the most stable predictions in cases of soil liquefaction. Recent studies have further demonstrated the effectiveness of ensemble machine learning approaches for breast cancer classification [22], [23]. Other ensemble-based studies have further explored breast cancer classification by integrating feature selection and image-processing strategies, indicating the continued development of ensemble approaches in this domain [24].

Having reviewed 28 prior studies, it is evident that Bagging and XGBoost consistently enhance model performance across various domains, including healthcare, education, automotive, energy, environment, and cybersecurity. Bagging has been widely recognized for improving the stability of high-variance algorithms such as Decision Tree and KNN, while XGBoost has demonstrated strong capability in handling complex data patterns and reducing bias to achieve higher predictive precision [25]. Recent studies have also reported the effectiveness of ensemble-based approaches for breast cancer detection and classification [26]. However, existing studies predominantly apply these ensemble techniques independently or restrict their analysis to specific model types, without systematically investigating their sequential integration across heterogeneous base classifiers within a unified evaluation setting. Therefore, this study proposes a staged hybrid ensemble framework that integrates baseline model training, Bagging-based variance reduction, and subsequent enhancement using XGBoost. Rather than merely comparing model performance, this research

systematically examines the generalizability of the sequential Bagging–XGBoost strategy across five machine learning algorithms—SVM, NN, LR, DT, and KNN—under a consistent experimental protocol. By evaluating both predictive accuracy and class-separation capability through confusion matrix analysis, ROC curves, and AUC, this study aims to provide methodological insight into how sequential ensemble integration contributes to performance improvement in medical classification tasks, particularly breast cancer diagnosis.

2. Methods

This study uses the publicly available Breast Cancer Wisconsin (Diagnostic) dataset obtained from the Kaggle platform. The dataset is commonly employed for breast cancer classification and contains numerical measurements describing characteristics of cell nuclei along with their corresponding diagnostic categories. It consists of 569 observations and 30 numerical features, with the diagnosis represented by two classes: malignant and benign. Further details regarding the dataset composition and class distribution are presented in the Results and Discussion section. The dataset was selected as the experimental basis for evaluating machine learning models for breast cancer classification.

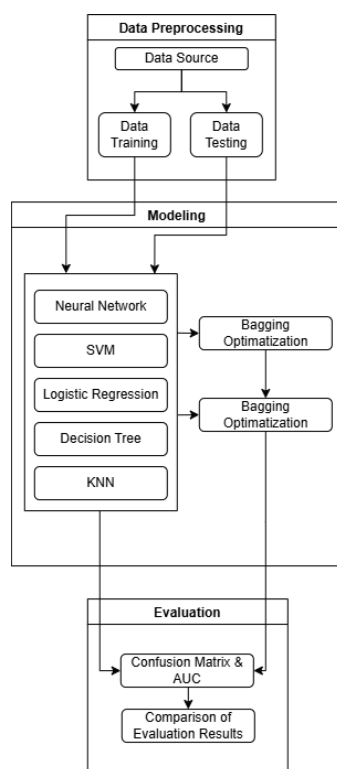


Figure 1. Research Stages of the Study

The research process is divided into three main stages as illustrated in Figure 1, namely Data Preprocessing; Modelling; and Evaluation. During data preparation, the dataset was partitioned into training and testing subsets using a 70:30 ratio. The training portion was used for model development, whereas the remaining

observations were reserved for evaluating model performance. This partitioning procedure established a consistent evaluation setting for all classifiers.

2.1 Modeling

The next stage is modelling, which is the core of the research. Five classification algorithms were selected as the basic models, namely SVM, NN, LR, DT, and KNN. These five algorithms were chosen because they have diverse characteristics in handling data and have been widely applied in classification studies in the medical field. Each classifier was first fitted using the training subset and subsequently evaluated on the held-out test subset to establish its baseline performance. The classifiers were then processed through the proposed staged ensemble configuration, in which Bagging was applied first and XGBoost was used as the subsequent enhancement stage.

Bagging was used to obtain more stable predictions by aggregating multiple fitted estimators, whereas the subsequent XGBoost stage was used to learn additional patterns from the resulting representation and address remaining prediction errors. Each base model was combined individually with Bagging and Boosting (XGBoost), so that it could be analyzed to what extent these ensemble techniques improved accuracy and performance compared to a single model.

2.2 Evaluation

Model performance was assessed for both the initial classifiers and the corresponding models obtained after the Bagging–XGBoost stages. Performance assessment incorporated confusion-matrix analysis, Receiver Operating Characteristic (ROC) curves, and the Area Under the Curve (AUC). Confusion matrices were examined to identify correct and incorrect classifications for each diagnostic class and to derive the corresponding classification measures. ROC analysis characterizes classifier behavior across different true-positive and false-positive rates, while AUC summarizes the model's ability to discriminate between the two diagnostic categories.

The resulting metrics were compared across the baseline, Bagging, and staged Bagging–XGBoost configurations to quantify the performance changes associated with each ensemble stage. This comparison focused not only on accuracy but also on class separation quality as reflected in the AUC value. This evaluation provided a consistent basis for examining the contribution of the ensemble stages to breast cancer classification performance.

To improve experimental reliability and minimize randomness bias, each experiment was repeated 10 times using different random seeds ranging from 1 to 10. For every repetition, stratified train–test splitting with a ratio of 70:30 was applied to preserve class distribution consistency. Accuracy results from the repeated runs were summarized using the mean and

standard deviation and then subjected to a paired t-test at a significance level of 0.05.

3. Results and Discussions

The dataset contains no missing values across its attributes; therefore, no imputation procedure was required before the processing and modelling stages. The data were subsequently partitioned into training and testing subsets, comprising 70% and 30% of the observations, respectively. A random split was applied while preserving the proportion of the diagnostic classes to reduce potential sampling bias. The resulting subsets were then used consistently to train and evaluate both the individual classifiers and the ensemble-based models.

Modelling is performed by applying five classification algorithms—both in the form of single models and with optimization through the Bagging and XGBoost methods. The performance of each model is assessed based on accuracy, confusion matrices, ROC curves, and AUC values. The evaluation results are presented in tables and graphs to facilitate comparative analysis of model performance.

3.1 Data Preprocessing

The initial stage of the experiment begins with the collection of data from Kaggle platform that formed the basis for analysis via `sklearn.datasets.load_breast_cancer`. The classification labels consist of two categories, namely malignant and benign. The dataset contains complete records with no missing values; therefore, no additional data-cleaning procedure was required before the modelling stage.

Once the dataset was confirmed to be suitable for analysis, it was partitioned into two subsets, with 70% allocated for training and the remaining 30% for testing. The training subset was used to develop the classification models, while the testing subset was reserved for evaluating their predictive performance. This data partitioning served as the foundation for the subsequent modelling stage.

The experiments were conducted in Python using the scikit-learn, XGBoost, Matplotlib, and Seaborn libraries on a high-performance computer equipped with 64 GB RAM, an Intel Core i9-13900K processor, and 500 GB SSD storage. The base classifiers were configured as follows: SVM with a linear kernel and probability estimation enabled; NN/MLP with hidden layers of (100, 50) and a maximum of 1,000 iterations; Logistic Regression with a maximum of 10,000 iterations; Decision Tree with a maximum depth of 3; and KNN with three nearest neighbours. The Bagging models were configured with 10 estimators and a fixed random state of 42. For XGBoost, the selected parameters included a maximum tree depth ranging from 3 to 4, a learning rate of 0.1, and 100–150 estimators, with logloss specified as the evaluation metric.

3.2 Modelling the Data

The initial modelling stage was carried out by training each basic classification algorithm using the prepared training data. This process was carried out separately for each algorithm. Each classifier learns the underlying patterns from the training data and establishes a classification function that is subsequently used to generate predictions on the test data. The training procedure was implemented in Python. Figure 2 illustrates the corresponding workflow, where all classifiers receive the same training data as input and independently generate their respective predictive models. The models formed at this stage are then evaluated to measure their classification performance in the next stage.

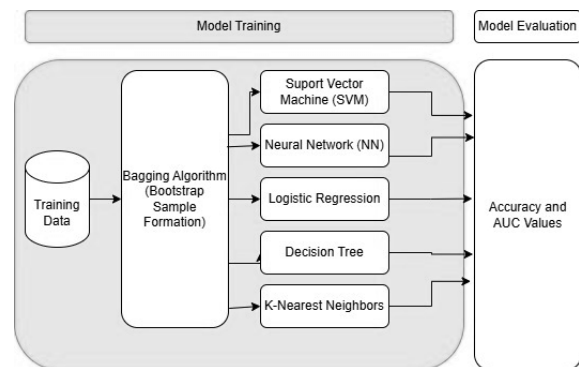


Figure 2. Training Flow of Base Classifiers with Bagging Technique

After all baseline models have been trained using the Bagging technique as described in Figure 2, the process continues to the next optimization stage using XGBoost. Figure 3 illustrates how the Bagging model results are used as input for XGBoost training, demonstrating the continuity of the process from variance reduction (Bagging) to increased focus on errors (Boosting). This hybrid approach allows for sequential utilization of the advantages of both techniques.

After all basic algorithm models were trained using the Bagging technique in the first modelling stage, the next stage involved further optimization by applying the XGBoost algorithm. This stage was intended to enhance the Bagging model by applying a boosting mechanism that focuses on patterns associated with previously misclassified or challenging instances. Training at this stage is carried out by using the Bagging model output as the basis for the training process using XGBoost. The training flow is visualized in Figure 3, which illustrates the integration between the Bagging model results and the adaptive weighting mechanism characteristic of XGBoost during the model formation process.

In the proposed staged Bagging–XGBoost framework, each base classifier is first trained using the Bagging technique to obtain more stable prediction results and reduce model variance. After the Bagging model is trained, prediction outputs are generated for both the training and testing datasets. These prediction outputs

are then appended as an additional feature to the original dataset, resulting in a hybrid feature

representation that combines the original attributes with the knowledge learned by the Bagging model.

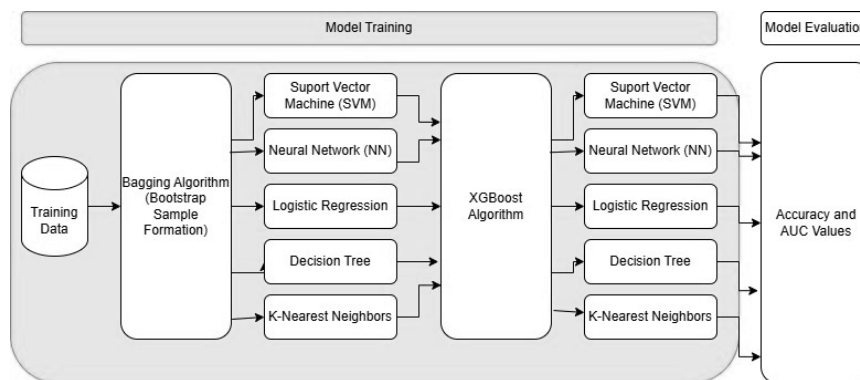


Figure 3. Optimization of Bagging results with XGBoost

Subsequently, XGBoost is trained using this augmented feature set to further improve classification performance by identifying residual patterns and correcting prediction errors that may remain after the Bagging stage. Through this sequential optimization process, Bagging serves as a variance-reduction mechanism, while XGBoost acts as a performance enhancement stage, enabling the framework to exploit the complementary strengths of both ensemble techniques within a unified classification pipeline.

The implementation of the proposed staged Bagging–XGBoost framework begins by training a Bagging ensemble for each selected base classifier using the training dataset. After training, the Bagging model generates prediction outputs for both the training and testing datasets. These prediction outputs are then appended to the corresponding original feature sets as an additional feature, producing hybrid training and testing datasets. XGBoost is subsequently trained using the hybrid training dataset, while the hybrid testing dataset is used to generate the final classification predictions. The final model performance is assessed using accuracy, Receiver Operating Characteristic (ROC) curves, and Area Under the Curve (AUC) metrics. This sequential implementation ensures that the Bagging stage is completed before the XGBoost stage and that the prediction information generated by Bagging is explicitly incorporated into the subsequent XGBoost learning process.

3.3 Evaluation

The experimental findings presented in Table 1 show that the addition of XGBoost generally resulted in higher performance than Bagging across the evaluated algorithms. Neural Network, for example, increased from an accuracy of 95.3% in Bagging to 98.2% in XGBoost, with an almost perfect AUC of 99.9%. Decision Tree, which tends to be prone to overfitting, also experienced an improvement, with accuracy increasing from 94.7% to 95.3% and AUC from 98.3% to 99.0%.

Even K-Nearest Neighbours showed a stable increase from 95.3% accuracy to 95.9% and AUC from 99.4%

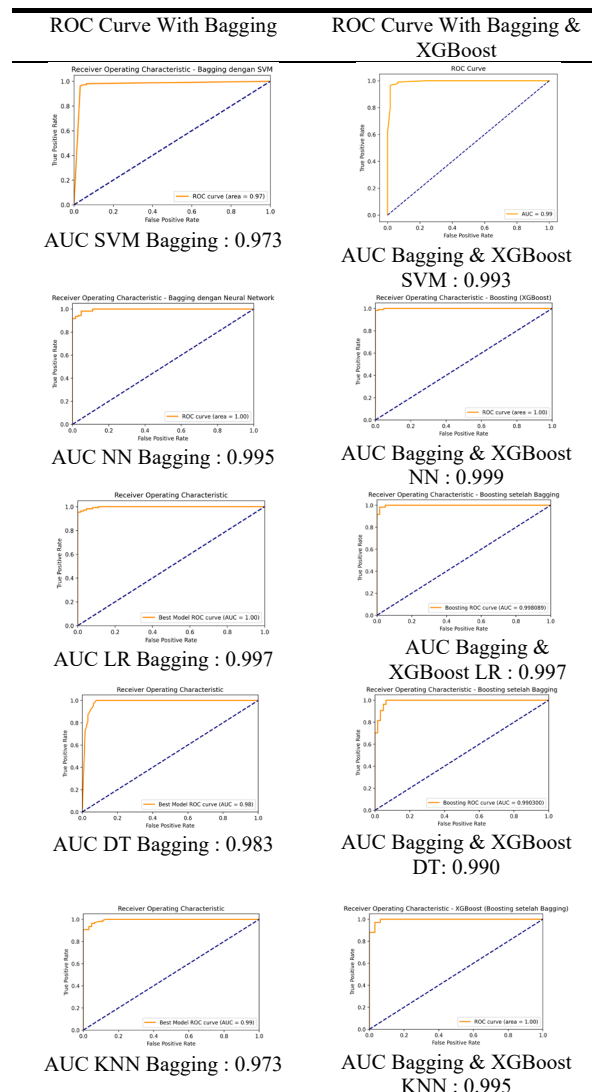
to 99.5%. Thus, the application of the ensemble method showed a significant improvement in performance for each classification algorithm.

Table 1. Accuracy Comparison

Accuracy of Bagging	Accuracy of Bagging & XGBoost
Confusion Matrix - Bagging dengan SVM SVM Bagging Accuracy: 0.959	Confusion Matrix-Bagging & Boosting XGBoost (SVM) SVM Bagging & XGBoost Accuracy: 0.964
Confusion Matrix - Bagging dengan Neural Network NN Bagging Accuracy: 0.953	Confusion Matrix - Bagging & Boosting XGBoost (NN) NN Bagging & XGBoost Accuracy: 0.982
Confusion Matrix - Bagging dengan Logistic Regression LR Bagging Accuracy: 0.965	Confusion Matrix - Bagging & Boosting XGBoost (LR) LR Bagging & XGBoost Accuracy: 0.976
Confusion Matrix - Bagging dengan Decision Tree DT Bagging Accuracy: 0.947	Confusion Matrix - Bagging & Boosting XGBoost (DT) DT Bagging & XGBoost Accuracy: 0.953
Confusion Matrix - Bagging dengan k-Nearest Neighbors KNN Bagging Accuracy: 0.953	Confusion Matrix - Bagging & Boosting XGBoost (KNN) KNN Bagging & XGBoost Accuracy: 0.959

The evaluation process was continued by comparing models optimized using Bagging with models that were subsequently combined with the Boosting technique through XGBoost. Table 2 presents the ROC curve comparison between the baseline models and those enhanced through Bagging and XGBoost. When compared to previous studies using AdaBoost, a significant difference can be seen in the scope of algorithms that can be optimized.

Table 2. Comparison of Bagging vs. XGBoost ROC Curves



With AdaBoost, performance gains were primarily observed for linear and decision-tree-based classifiers, including Logistic Regression, Support Vector Machine, and Decision Tree. In contrast, Neural Networks and KNN showed limited improvement because the conventional AdaBoost weighting mechanism is less suited to these classifiers. In contrast, this study shows that with XGBoost, the performance of all algorithms tested—including Neural Network and KNN—can be consistently improved, both in terms of accuracy and AUC. Thus, the results of this study reinforce the evidence that XGBoost is superior to AdaBoost, not only in terms of improved accuracy and

AUC, but also in the range of algorithms that can be optimized.

Table 3 summarizes the accuracy and AUC obtained from the staged Bagging–XGBoost framework across the five evaluated classifiers. The results show that the addition of XGBoost improved accuracy for all classifiers, with the largest gain observed for the Neural Network, increasing from 95.3% to 98.2%. The AUC values also increased for SVM, Neural Network, Decision Tree, and K-Nearest Neighbours, while Logistic Regression maintained a high AUC of 99.7%. These results indicate that the proposed sequential framework improves both predictive accuracy and class-separation capability across different classifier architectures.

Table 3. Accuracy and AUC Calculation Results of Xgboost

Model	Accuracy		AUC	
	Bagging	Bagging & XGBoost	Bagging	Bagging & XGBoost
SVM	95.9%	96.4%	97.3%	99.3%
NN	95.3%	98.2%	99.5%	99.9%
LR	96.5%	97.6%	99.7%	99.7%
DT	94.7%	95.3%	98.3%	99.0%
KNN	95.3%	95.9%	97.3%	99.5%

For comparison with the previous Bagging–AdaBoost approach, Table 4 presents the reported accuracy and AUC results for the same set of base classifiers [5]. The comparison shows that Bagging combined with AdaBoost could be evaluated for SVM, Logistic Regression, and Decision Tree, whereas Neural Network and K-Nearest Neighbours were reported as not compatible with the AdaBoost implementation. These results provide a reference point for assessing whether the proposed Bagging–XGBoost framework offers broader applicability across different classifier architectures.

Table 4. Accuracy and AUC Calculation Results of Adaboost

Model	Accuracy		AUC	
	Bagging	Bagging & AdaBoost	Bagging	Bagging & AdaBoost
SVM	95.90%	97.0%	97.30%	97.0%
NN	95.30%	Not Compatible	99.50%	Not Compatible
LR	96.50%	97.1%	99.70%	99.7%
DT	94.70%	95.3%	98.30%	98.9%
KNN	95.30%	Not Compatible	97.30%	Not Compatible

In contrast, the current study demonstrates that XGBoost can enhance all five algorithms tested, including Neural Network and KNN. For instance, Neural Network improved from 95.3% to 98.2% accuracy with an almost perfect AUC of 99.9%, while KNN improved from 95.3% to 95.9% accuracy and AUC from 99.4% to 99.5%. These results indicate that XGBoost has broader applicability and greater robustness compared to AdaBoost, particularly when applied to non-linear or non-tree-based algorithms.

Another notable difference is in AUC stability. With AdaBoost, SVM experienced a slight decrease in AUC (from 97.3% to 97.0%) despite accuracy gains. By

contrast, under XGBoost, SVM not only improved its accuracy (from 95.9% to 96.4%) but also achieved a substantial increase in AUC (from 97.3% to 99.3%). These findings suggest that XGBoost provides a more consistent performance improvement across both accuracy and class-separation capability.

Taken together, the findings suggest that while AdaBoost remains effective for tree-based and linear models, XGBoost offers superior generalization and flexibility, making it a stronger candidate for medical classification tasks where diverse algorithms are often required.

In addition to performance comparison obtained from single experimental runs presented in previous tables, statistical analysis was conducted using repeated experiments to verify whether the observed improvements were statistically significant. Each experiment was repeated 10 times using different random seeds, and the aggregated results are summarized in Table 5.

Table 5. Statistical Significance Test Results (Paired t-test)

Model	Mean Accuracy (Bagging)	Mean Accuracy (Bagging + XGBoost)	p-value	Significance
SVM	0.956 ± 0.010	0.972 ± 0.012	0.0003	Significant
NN	0.931 ± 0.014	0.970 ± 0.012	0.0001	Significant
LR	0.952 ± 0.004	0.971 ± 0.007	<0.0001	Significant
DT	0.949 ± 0.020	0.954 ± 0.017	0.0150	Significant
KNN	0.933 ± 0.010	0.965 ± 0.012	<0.0001	Significant

The statistical analysis suggests that the observed performance improvements are unlikely to be caused by random variation. As shown in Table 5, all evaluated models achieved statistically significant improvements ($p < 0.05$) after applying the staged Bagging–XGBoost framework. Neural Networks demonstrated the largest performance increase, suggesting that models with higher training variability may benefit more substantially from sequential ensemble integration, whereas models such as Decision Trees exhibited relatively smaller but still statistically meaningful improvements.

The substantial improvement observed in Neural Networks can be attributed to their sensitivity to training instability and local minima, where Bagging helps reduce variance while XGBoost enhances difficult-to-classify instances through iterative error correction. In contrast, algorithms such as Logistic Regression, which are inherently more stable and less sensitive to variance, exhibit smaller performance gains. This indicates that the effectiveness of the staged ensemble framework is influenced by the intrinsic

learning characteristics of each algorithm, particularly their sensitivity to variance and bias.

These observations provide important insight into the applicability of ensemble strategies beyond their conventional use in tree-based models. While XGBoost is inherently designed for decision tree structures, the findings demonstrate that its integration within a staged hybrid framework can also enhance non-tree-based models such as Neural Networks and Logistic Regression. This observation is relevant because non-tree models rely on different optimization mechanisms and are not naturally aligned with boosting-based iterative error correction. The observed improvement suggests that the sequential combination of variance reduction through Bagging and bias-focused enhancement using XGBoost may provide a more model-agnostic optimization strategy.

In addition to predictive performance evaluation, computational analysis was conducted to assess the overhead introduced by the proposed staged Bagging–XGBoost framework. Training time comparison was performed to quantify the additional computational cost associated with sequential ensemble integration. The computational analysis results are summarized in Table 6.

Table 6. Computational Cost Comparison Between Bagging and Staged Bagging–XGBoost Framework

Model	Bagging Training Time (s)	Bagging + XGBoost Training Time (s)	Additional Time (s)
SVM	34.821	35.336	+0.515
NN	3.425	3.461	+0.036
LR	3.867	3.996	+0.129
DT	0.042	0.125	+0.083
KNN	0.019	0.114	+0.095

The computational analysis indicates that the proposed staged Bagging–XGBoost framework introduces additional training overhead across all evaluated models. This increase is expected because the framework performs sequential ensemble processing, where Bagging outputs must first be generated before subsequent boosting stages are applied. Although additional runtime is observed across all models, the absolute increase remains relatively small compared to the predictive performance improvements achieved. These findings suggest that the computational overhead introduced by the proposed framework remains practical for medical classification scenarios where predictive reliability is prioritized. Taken together, these findings indicate that the proposed staged framework provides a practical trade-off between computational cost and predictive performance improvement across diverse classification models.

4. Conclusions

The evaluation of five classifiers, Support Vector Machine (SVM), Neural Network (NN), Logistic

Regression (LR), Decision Tree (DT), and K-Nearest Neighbours (KNN) shows that the proposed staged ensemble approach can improve predictive performance and class discrimination across different model architectures. The sequential application of Bagging followed by XGBoost provides a structured optimization process in which the first stage supports prediction stability and the subsequent stage further enhances the discriminatory capability of the resulting models.

The findings reveal that performance improvement is not limited to tree-based models, but extends to non-tree-based classifiers such as Neural Networks and Support Vector Machines. This indicates that the effectiveness of XGBoost within the proposed framework is not confined to its traditional association with decision-tree structures, thereby extending its applicability across different learning paradigms. These findings suggest that sequential ensemble integration may extend beyond conventional model-specific optimization strategies.

Furthermore, the consistent improvement in both accuracy and AUC across models suggests that the staged Bagging–XGBoost strategy enhances not only predictive precision but also robustness in class separation. This dual improvement is particularly relevant in medical diagnosis contexts, where reliable discrimination between malignant and benign cases directly impacts clinical decision-making.

Overall, the results indicate that the proposed staged ensemble framework offers a systematic means of examining how sequential ensemble techniques contribute to predictive performance across different classification models. The findings provide practical and methodological considerations for developing ensemble strategies in medical classification, particularly for breast cancer diagnosis, and establish a basis for further evaluating the applicability of hybrid ensemble approaches across other high-stakes domains.

Despite the promising results, this study has several limitations. First, the experiments relied on a single benchmark dataset, which may restrict the applicability of the findings to medical datasets with greater complexity or class imbalance. Second, the proposed staged Bagging–XGBoost framework introduces additional computational overhead due to sequential ensemble integration, resulting in increased training time and higher memory requirements due to sequential ensemble processing and intermediate feature generation. Future studies could explore computationally efficient optimization strategies that preserve predictive performance while reducing processing requirements, as well as assess the proposed framework using larger and more heterogeneous datasets.

References

- [1] Q. H. Hoang, L. M. Duong, P. L. L. Bui, A. V. Tran, T. A. Nguyen, and V. D. Nguyen, "A Comparative Study of Machine Learning Algorithms for Breast Cancer Classification," in *2023 International Conference on Advanced Technologies for Communications (ATC)*, 2023, pp. 409–414. doi: [10.1109/ATC58710.2023.10318887](https://doi.org/10.1109/ATC58710.2023.10318887).
- [2] H. Chen, N. Wang, X. Du, K. Mei, Y. Zhou, and G. Cai, "Classification Prediction of Breast Cancer Based on Machine Learning," *Comput. Intell. Neurosci.*, vol. 2023, no. 1, 2023, doi: [10.1155/2023/6530719](https://doi.org/10.1155/2023/6530719).
- [3] Fadhila Tangguh Admojo and B. S. Waluyo Poetro, "Comparative Study on the Performance of the Bagging Algorithm in the Breast Cancer Dataset," *International Journal of Artificial Intelligence in Medical Issues*, vol. 1, no. 1, pp. 36–44, 2023, doi: [10.56705/ijaimi.v1i1.87](https://doi.org/10.56705/ijaimi.v1i1.87).
- [4] M. Dianati-nasab and K. Salimifard, "Machine learning algorithms to uncover risk factors of breast cancer : insights from a large case-control study," *Front Oncol*, no. February, pp. 1–13, 2024, doi: [10.3389/fonc.2023.1276232](https://doi.org/10.3389/fonc.2023.1276232).
- [5] R. Pramudita, N. Safitri, and V. Z. Nazah, "Studi Komparatif Algoritma Machine Learning dengan Teknik Bagging dan AdaBoost pada Klasifikasi Kanker Payudara," *Tematik*, vol. 12, no. 1, pp. 101–108, 2025, doi: [10.38204/tematik.v12i1.2435](https://doi.org/10.38204/tematik.v12i1.2435).
- [6] R. Pramudita, Solikin, and N. Safitri, "Optimization Analysis of Neural Network Algorithms Using Bagging Techniques on Classification of Date Fruit Types," in *Seventh International Conference on Informatics and Computing (ICIC)*, Denpasar: IEEE, 2022. [Online]. Available: <https://ieeexplore.ieee.org/document/10006986>
- [7] A. Zizaan and A. Idri, "Evaluating and comparing bagging and boosting of hybrid learning for breast cancer screening," *Sci. Afr.*, vol. 23, no. November 2023, p. e01989, 2024, doi: [10.1016/j.sciaf.2023.e01989](https://doi.org/10.1016/j.sciaf.2023.e01989).
- [8] M. Rahman, K. H. Kobir, S. Akther, and A. H. Kallol, "Ensemble Machine Learning for Enhanced Breast Cancer Prediction : A Comparative Study," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 15, no. 7, pp. 932–941, 2024, doi: [10.14569/IJACSA.2024.0150792](https://doi.org/10.14569/IJACSA.2024.0150792).
- [9] M. Dutta, K. M. Hasan, A. Akter, and H. Rahman, "An interpretable machine learning-based breast cancer classification using XGBoost , SHAP , and LIME," *Bulletin of Electrical Engineering and Informatics*, vol. 13, no. 6, pp. 4306–4315, 2024, doi: [10.11591/eei.v13i6.7866](https://doi.org/10.11591/eei.v13i6.7866).
- [10] M. Nhu, U. Le, J. Zhou, D. Phu, C. Le, and D. Wang, "Ensemble of XGBoost Classifiers Based on LDA Dimensionality Reduction for Predicting Breast Cancer," *Universal Journal of Public Health*, vol. 12, no. 3, pp. 434–440, 2024, doi: [10.13189/ujph.2024.120302](https://doi.org/10.13189/ujph.2024.120302).
- [11] S. Y. Prasetyo, "Synergizing Ensemble Techniques for Precision Breast Cancer Diagnosis: A Comparative Performance Analysis," *Proceeding of the IEEE International Conference on Smart Instrumentation, Measurement and Applications, ICSIMA*, no. 2024, pp. 297–302, 2024, doi: [10.1109/ICSIMA62563.2024.10675541](https://doi.org/10.1109/ICSIMA62563.2024.10675541).
- [12] D. T. Avlani, M. B. Abhijna, G. S. Sai Disha, A. Kodipalli, and T. Rao, "Comprehensive Methodologies for Breast Cancer Classification: Leveraging XAI LIME, SHAP Bagging Boosting, and Diverse Single Classifiers," *2024 4th Asian Conference on Innovation in Technology, ASIANCON 2024*, pp. 1–6, 2024, doi: [10.1109/ASIANCON62057.2024.10837893](https://doi.org/10.1109/ASIANCON62057.2024.10837893).
- [13] A. Asselman, M. Khaldi, and S. Aammou, "Enhancing the prediction of student performance based on the machine learning XGBoost algorithm," *Interactive Learning Environments*, vol. 31, no. 6, pp. 3360–3379, Aug. 2023, doi: [10.1080/10494820.2021.1928235](https://doi.org/10.1080/10494820.2021.1928235).
- [14] X. Deng et al., "Bagging–XGBoost algorithm based extreme weather identification and short-term load forecasting model," *Energy Reports*, vol. 8, pp. 8661–8674, 2022, doi: [10.1016/j.egyr.2022.06.072](https://doi.org/10.1016/j.egyr.2022.06.072).
- [15] O. Vigo, S. Lukas, D. Krisnadi, P. Widjaja, and P. Korespondensi, "Comparison Of Bagging And Adaboost Performance For Multi-Class Data Classification," *Journal Information System Development (ISD)*, vol. 7, no. 2, pp. 66–69, 2022, doi: [10.19166/isd.v7i2.547](https://doi.org/10.19166/isd.v7i2.547)

- [16] N. H. A. Malek, W. F. W. Yaacob, Y. B. Wah, S. A. Md Nasir, N. Shaadan, and S. W. Indratno, "Comparison of ensemble hybrid sampling with bagging and boosting machine learning approach for imbalanced data," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 29, no. 1, pp. 598–608, 2023, doi: [10.11591/ijeecs.v29.i1.pp598-608](https://doi.org/10.11591/ijeecs.v29.i1.pp598-608).
- [17] M. Niazkar et al., "Applications of XGBoost in water resources engineering: A systematic literature review (Dec 2018–May 2023)," *Environmental Modelling & Software*, vol. 174, p. 105971, 2024, doi: <https://doi.org/10.1016/j.envsoft.2024.105971>.
- [18] J. Li et al., "Application of XGBoost algorithm in the optimization of pollutant concentration," *Atmos. Res.*, vol. 276, p. 106238, 2022, doi: <https://doi.org/10.1016/j.atmosres.2022.106238>.
- [19] A. Plaia, S. Buscemi, J. Fürnkranz, and E. L. Mencía, "Comparing Boosting and Bagging for Decision Trees of Rankings," *J. Classif.*, vol. 39, no. 1, pp. 78–99, 2022, doi: [10.1007/s00357-021-09397-2](https://doi.org/10.1007/s00357-021-09397-2).
- [20] C. Zhao, R. Peng, and D. Wu, "Bagging and Boosting Fine-Tuning for Ensemble Learning," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 4, pp. 1728–1742, 2024, doi: [10.1109/TAI.2023.3296685](https://doi.org/10.1109/TAI.2023.3296685).
- [21] 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, 2023, doi: [10.1007/s00521-022-07856-4](https://doi.org/10.1007/s00521-022-07856-4).
- [22] T. Srinivas et al., "Novel Based Ensemble Machine Learning Classifiers for Detecting Breast Cancer," *Mathematical Problems in Engineering*, vol. 2022, no. 1, 2022, doi: [10.1155/2022/9619102](https://doi.org/10.1155/2022/9619102).
- [23] U. Naseem et al., "An Automatic Detection of Breast Cancer Diagnosis and Prognosis Based on Machine Learning Using Ensemble of Classifiers," *IEEE Access*, vol. 10, no. April, pp. 78242–78252, 2022, doi: [10.1109/ACCESS.2022.3174599](https://doi.org/10.1109/ACCESS.2022.3174599).
- [24] A. K. M. R. H. Rafid, S. Azam, S. Montaha, A. Karim, K. U. Fahim, and Z. Hasan, "An Effective Ensemble Machine Learning Approach to Classify Breast Cancer Based on Feature Selection and Lesion Segmentation Using Preprocessed Mammograms," *Biology*, vol. 11, no. 11, pp. 1964, 2022, doi: [10.3390/biology11111654](https://doi.org/10.3390/biology11111654).
- [25] T. R. Mahesh, V. V. Kumar, V. Muthukumaran, H. K. Shashikala, B. Swapna, and S. Guluwadi, "Performance Analysis of XGBoost Ensemble Methods for Survivability with the Classification of Breast Cancer," *Journal of Sensors*, vol. 2022, no. 1, 2022, doi: [10.1155/2022/4649510](https://doi.org/10.1155/2022/4649510).
- [26] M. Umer et al., "Breast Cancer Detection Using Convolved Features and Ensemble Machine Learning Algorithm," *Cancers (Basel)*, vol. 14, no. 23, pp. 1–18, 2022, doi: [10.3390/cancers14236015](https://doi.org/10.3390/cancers14236015).