

A. ALSABRY, M. ALGABRI

**ITERATIVE TUNING OF TREE-ENSEMBLE-BASED MODELS' PARAMETERS USING BAYESIAN OPTIMIZATION FOR BREAST CANCER PREDICTION**

*Alsabry A., Algabri M. Iterative Tuning of Tree-Ensemble-Based Models' parameters Using Bayesian Optimization for Breast Cancer Prediction.*

**Abstract.** The study presents a method for iterative parameter tuning of tree ensemble-based models using Bayesian hyperparameter tuning for states prediction, using breast cancer as an example. The proposed method utilizes three different datasets, including the Wisconsin Diagnostic Breast Cancer (WDBC) dataset, the Surveillance, Epidemiology, and End Results (SEER) breast cancer dataset, and the Breast Cancer Coimbra dataset (BCCD), and implements tree ensemble-based models, specifically AdaBoost, Gentle-Boost, LogitBoost, Bag, and RUSBoost, for breast cancer prediction. Bayesian optimization was used to tune the hyperparameters of the models iteratively, and the performance of the models was evaluated using several metrics, including accuracy, precision, recall, and f1-score. Our results show that the proposed method significantly improves the performance of tree ensemble-based models, resulting in higher accuracy, precision, recall, and f1-score. Compared to other state-of-the-art models, the proposed method is more efficient. It achieved perfect scores of 100% for Accuracy, Precision, Recall, and F1-Score on the WDBC dataset. On the SEER BC dataset, the method achieved an accuracy of 95.9%, a precision of 97.6%, a recall of 94.2%, and an F1-Score of 95.9%. For the BCCD dataset, the method achieved an accuracy of 94.7%, a precision of 90%, a recall of 100%, and an F1-Score of 94.7%. The outcomes of this study have important implications for medical professionals, as early detection of breast cancer can significantly increase the chances of survival. Overall, this study provides a valuable contribution to the field of breast cancer prediction using machine learning.

**Keywords:** iterative tuning, tree ensemble-based models, bayesian optimization, breast cancer, machine learning.

**1. Introduction.** Machine learning (ML) has a crucial role in predicting breast cancer (BC) and offers several benefits, including early detection and diagnosis, improved accuracy, personalized risk assessment, handling complex interactions, reducing false positives and negatives, and enabling continuous learning and improvement. By analyzing a vast amount of medical data, including mammograms, MRI scans, and patient health records, ML algorithms can identify patterns that might indicate the early stages of BC, leading to more effective treatment and improved patient outcomes [1].

Traditional methods of BC prediction, such as the BC Risk Assessment Tool (BCRAT) and Breast and Ovarian Analysis of Disease Incidence and Carrier Estimation Algorithm (BOADICEA) models, have limitations in their predictive accuracy [2, 3]. However, ML models can achieve higher accuracy rates, which are significantly higher than those of traditional models [4].

Furthermore, ML models have the ability to consider a wide range of risk factors, such as genetic data, lifestyle factors, and medical history, providing personalized risk assessments for individuals. This can help stratify prevention strategies and customize clinical management for each patient. In addition, ML algorithms can identify complex interactions among multiple heterogeneous risk factors, capturing nonlinear relationships and interactions that traditional models may overlook. ML models also have the potential to reduce false positives and negatives in BC diagnoses, preventing unnecessary treatments for those wrongly diagnosed and ensuring timely treatment for those with the disease. Finally, ML models can continuously learn and improve over time as they are exposed to more data, which can result in improved predictive accuracy as they analyze more patient data and learn from previous predictions [5]. Tree ensemble-based models, such as AdaBoost, Gentle-Boost, LogitBoost, Bag, and RUSBoost, are powerful ML tools that can be used for a variety of tasks, including predicting BC. These models work by creating decision trees and making predictions based on iteratively improving the predictions [6].

Traditionally, hyperparameters are tuned using methods like grid search or random search, which involve trying out many different combinations of hyperparameters and selecting the one that performs best on a validation set. However, these methods can be computationally expensive and do not guarantee finding the optimal set of hyperparameters [7].

Bayesian hyperparameter tuning is a more sophisticated approach that treats hyperparameter tuning as a Bayesian optimization problem. It builds a probabilistic model of the objective function (i.e., the validation error as a function of the hyperparameters) and uses this model to select the most promising hyperparameters to try next. This approach can be more efficient than grid search or random search because it uses information from previous evaluations to make smarter decisions about what hyperparameters to try next [8].

**1.1. Authors Contributions.** This study makes a significant contribution to the field of BC prediction across different datasets. By applying Bayesian hyperparameter tuning to tree ensemble-based models through several iterations, the study aims to enhance the performance of the models and generalization capabilities for BC prediction in diverse datasets. The challenges of model adaptability and robustness are tackled in this study through systematic evaluation and assessment of various datasets. The findings of this study can provide valuable insights into the effectiveness and transferability of the proposed approach across various

BC datasets, contributing to the development of more reliable and versatile prediction models.

In the following sections, we will review relevant literature, describe the methodology employed in this study, present the experimental results, discuss the implications of our findings, and a comparative analysis is conducted to compare the results obtained from the proposed method to those of state-of-the-art models and to the findings of a literature review. By the end of this research, we will have provided valuable insights into the iterative tuning of tree ensemble-based models using Bayesian hyperparameter tuning for BC prediction.

**2. Review of Literature.** This literature review aims to investigate prior research on using ML for BC prediction, with a particular emphasis on tree-ensemble-based models. The review will also cover various tree ensemble-based models such as AdaBoost, GentleBoost, LogitBoost, Bag, and RUSBoost and their applications in BC prediction. Additionally, current approaches to hyperparameter tuning, such as grid search and Bayesian hyperparameter tuning, will be discussed. The objective of this review is to identify the most efficient tree-ensemble-based models and parameter tuning methods for BC prediction.

Table 1 serves as a comprehensive summary of the related works, providing a clear and concise overview of the studies analyzed in this research.

**2.1. Previous studies on breast cancer prediction.** The research objectives of previous studies regarding BC prediction were diverse. Some studies aimed to predict the presence or absence of BC, using the BCCD dataset. Other studies focused on classifying breast tumors as benign or malignant, utilizing the WDBC dataset. Additionally, some studies aimed to predict patient survival or death, and the SEER dataset was used for this purpose. In this section, we delve into the studies carried out for each of these datasets.

**2.1.1. Studies Utilizing the WDBC Dataset in Prior Research.** Numerous studies have utilized the WDBC dataset to assess various machine learning (ML) algorithms and techniques for binary classification. These studies have employed a diverse range of classification methods, including Support Vector Machine (SVM), Random Forest (RF), Extreme Learning Machine (ELM), and Naive Bayes. In some of these studies, optimization techniques were utilized to enhance the performance of the classification algorithms.

In one such study, [9] achieved the highest accuracy of 99.3% by utilizing an optimized SVM with Bayesian hyperparameter optimization.

This study exemplified the effectiveness of leveraging a well-established optimization technique to boost the classification algorithm's performance.

Table 1. Summary of the related works

Ref.	Year	Dataset	Methods / Techniques	Accuracy (%)
[10]	2019	WDBC	WQPSO with smooth SVM	98.42
[12]	2019	WDBC	SVM with 10 selected features	96.72
[13]	2020	WDBC	Optimized FSTBSVM with Jaya optimization techniques	94.36
[11]	2021	WDBC	Cloud-based ELM	98.68
[14]	2021	WDBC	SVM and RF	96.5
[9]	2022	WDBC	Optimized SVM with bayesian hyperparameter optimization	99.3
[15]	2019	SEER BC	Gradient Boosting with Genetic Algorithm	75.03
[16]	2020	SEER BC	Improved Random Forest - based rule extraction	80.5
[17]	2020	SEER BC	J48	93
[18]	2022	SEER BC	RF	94.6
[19]	2020	BCCD	GA and Gradient Boosting Classifier	79
[20]	2021	BCCD	Adaboost Classifier	80
[9]	2022	DCCD	polynomial SVM	76.9

Study [10] achieved an accuracy of 98.42% using Water Quality Prediction using Particle Swarm Optimization (WQPSO) with smooth SVM, indicating that the algorithms used in these studies are effective for the WDBC dataset.

Study [11] achieved an accuracy of 98.68% using cloud-based ELM, which is slightly higher than the accuracy achieved by [10]. ELM is a relatively new algorithm that has been shown to be effective for classification tasks, and this study demonstrated its usefulness for the WDBC dataset.

Study [12] achieved an accuracy of 96.72% using SVM with 10 selected features, which is slightly lower than the other studies. However, this study used feature selection techniques to identify the most relevant features, which can reduce the computational complexity of the classification models and improve their performance.

Study [13] achieved an accuracy of 94.36% using optimized FSTBSVM with Jaya optimization techniques, which is lower than the other studies. However, this study explored a relatively new technique for

classification and demonstrated its effectiveness in achieving high accuracy.

Study [14] examined the performances of six different classification algorithms and achieved an accuracy of 96.5% using SVM and RF. While this study did not achieve the highest accuracy, it provided a comprehensive evaluation of different algorithms' performances on the WDBC dataset.

The studies included in this comparison exhibited high accuracy in classifying the WDBC dataset. The performance of the classification model was considerably influenced by the selection of algorithm, technique, and feature selection. While study [9] achieved the highest accuracy by utilizing optimized SVM with Bayesian hyperparameter optimization, indicating its efficacy in classifying the WDBC dataset, other studies also attained high accuracy using different algorithms and techniques. These findings demonstrate the significance of exploring various methods for classification tasks.

### **2.1.2. Studies Utilizing the SEER BC Dataset in Prior Research.**

Several other studies have concentrated on improving ML techniques to develop models for predicting the survival of BC patients using the SEER BC dataset. These studies employed different algorithms and techniques for classification, such as Gradient Boosting, RF, and J48 decision tree.

Study by [18] achieved the highest accuracy of 94.64% using RF, indicating the effectiveness of this algorithm for the SEER BC dataset. RF is a well-established algorithm for classification tasks, and its success in this study further emphasizes its utility for BC prediction tasks. Similarly, study by [17] achieved an accuracy of 93.02% using the J48 decision tree algorithm, demonstrating the effectiveness of Decision Tree (DT) algorithms for the SEER BC dataset.

In contrast, study [15] achieved the lowest accuracy of 75.03% using Gradient Boosting with Genetic Algorithm (GA). While this study demonstrated the potential of using optimization techniques to improve the performance of classification algorithms, it was not as effective as other studies in achieving high accuracy for the SEER BC dataset.

Study [16] explored a novel approach for rule extraction and classification, achieving an accuracy of 80.45%, which is lower than the accuracies achieved by the other studies. However, this study's approach has the potential to improve the accuracy of classification models, demonstrating the importance of exploring novel techniques in the field of BC prediction using ML.

In general, the studies presented in this revision achieved varying levels of accuracy in classifying the SEER BC dataset. The choice of

algorithm significantly affected the performance of the classification model. Studies [18] and [17] achieved high accuracy using well-known algorithms such as RF and J48, while study [16] explored a novel approach for rule extraction and classification.

### **2.1.3. Studies Utilizing the BCCD Dataset in Prior Research.**

Similarly, several studies have explored the use of ML techniques in the development of models for BC diagnosis using the BCCD dataset. These studies have used different algorithms such as Adaboost, Gradient Boosting, and SVM, among others. They have also employed feature selection and optimization techniques to improve model performance.

Study [20] achieved the highest accuracy of 80% using the Adaboost Classifier. This study demonstrated the effectiveness of using Adaboost for the BCCD dataset, which is a well-known algorithm for classification tasks.

Study [19] achieved an accuracy of 79% using the Gradient Boosting Classifier with the Genetic Algorithm for feature selection. This study demonstrated the effectiveness of using feature selection techniques to identify the most relevant features for classification, which can reduce the computational complexity of the classification models and improve their performance.

Study [9] achieved an accuracy of 76.9% using a polynomial SVM, which is lower than the other studies. However, this study explored a different algorithm than Adaboost and Gradient Boosting and demonstrated the potential of using a polynomial SVM for the BCCD dataset.

Overall, the studies presented in this comparison achieved varying levels of accuracy in classifying the BCCD dataset. The choice of algorithm and technique significantly affected the performance of the classification model. Studies [20] and [19] achieved high accuracy using Adaboost and Gradient Boosting with GA for feature selection, respectively. Study [9] explored a different algorithm and achieved lower accuracy but demonstrated the potential of using a polynomial SVM for the BCCD dataset.

**2.2. Existing tree ensemble-based models.** This section focuses on reviewing the tree-ensemble-based models that exist, including AdaBoost, GentleBoost, LogitBoost, Bag, and RUSBoost, and their applications in breast cancer (BC) prediction. Each of these models possesses unique characteristics that can be effective for different datasets and objectives. A detailed description of each model and its algorithm will be presented. Additionally, we will examine the applications of these models in BC prediction, including their performance on different datasets and feature selection. The objective of this section is to offer insights into the strengths

and weaknesses of each model and identify the most effective models for BC prediction.

**2.2.1. Bagged Trees.** It is an ML ensemble meta-algorithm designed to improve the stability and accuracy of ML algorithms used in statistical classification and regression. The algorithm was first introduced by Breiman in 1996 and has since been widely used in various applications such as text classification, image classification, and bioinformatics [21].

The basic idea behind bagging is to generate multiple versions of a predictor and use these to get an aggregated predictor. The aggregation averages usually over the predictions for regression problems and does a majority vote for classification problems. The Bagged Trees algorithm has several advantages. Firstly, it can reduce overfitting and improve the generalization performance of the model. Secondly, it is robust to noise and outliers in the data. Finally, it can handle high-dimensional feature spaces and large datasets. However, Bagged Trees have some limitations. One of the main limitations is that it can be computationally expensive, especially when the number of trees in the ensemble is large. Additionally, the interpretability of the model is reduced as the number of trees increases. Finally, the quality of the predictions can be affected by the choice of hyperparameters such as the number of trees, the depth of each tree, and the size of the bootstrap samples [22, 23].

The process of Bagged Trees can be described as follows [22].

---

**Algorithm 1.** Bagging algorithm when applied to decision trees for a classification problem

---

**Initialize:** Determine the number of bootstrap samples,  $B$ , to be created.

**For  $b = 1$  to  $B$ , repeat steps 1-3:**

**Step 1. Bootstrap Sampling:** Create a bootstrap sample by randomly selecting  $N$  instances from the original dataset with replacement, where  $N$  is the size of the dataset.

**Step 2. Tree Building:** Build a decision tree based on the bootstrap sample. Grow the tree to maximum size and do not perform any pruning.

**Step 3. End of the loop:** Return to step 2 and repeat the process until  $B$  trees have been grown.

**Prediction:** For a new data point, make a prediction with each of the  $B$  trees. The final prediction is the class that gets the most votes among the  $B$  trees.

---

Several studies have investigated the efficacy of the bagged trees algorithm for BC classification. However, there are variations in the datasets utilized and the accuracy achieved by these studies.

One study [24] applied the SMOTE technique for oversampling the data acquired from Shengjing Hospital of China Medical University. The study used the Bagged Tree algorithm and achieved an accuracy of 70.3%.

Another study [25] investigated a supervised learning technique for classifying BC using four different classifiers, namely Boosted Tree, Bagged Tree, Logistic Regression (LR), and Artificial Neural Networks (ANN). The ANN outperformed the other classifiers with an accuracy of 97.56%, while the bagged tree achieved the second-best accuracy. This study highlights the effectiveness of the ANN and bagged tree in classifying BC and demonstrates the importance of comparing multiple classifiers to identify the best-performing one.

In a third study [26], the performance of the bagged trees algorithm was evaluated on a dataset of 23 attributes containing 575 samples obtained from the Mizoram State Cancer Institute of Aizawl, Mizoram, India. An accuracy of 82.5% was achieved, which is higher than the first study but lower than the second study. However, the study was limited by the small size of the dataset, which may affect the generalization of the results.

**2.2.2. Adaboost Trees.** It is a variant of AdaBoost, which uses DT as a weak classifier. In each iteration, a DT is trained on the weighted samples, and the weights are updated based on the misclassification rate. The final prediction is made by combining the predictions of all the DTs, typically by taking the weighted average.

Studies have widely used the Adaboost algorithm for BC classification. For example, in [27], the performance of DT and Adboost was evaluated on an imbalanced dataset such as WDBC. Both models achieved high accuracy, with DT achieving 88.8% and Adboost achieving 92.5%. The study highlights the importance of selecting appropriate models for imbalanced datasets, such as Adboost, which is designed to handle such datasets, and demonstrates its efficacy in classifying BC.

In another study [28], ten models, including Adboost, RF, Tree, Gradient Boosting, KNN, ANN, Naïve Bayes, SVM, LR, and SGD, were compared for their performance in BC classification. Adboost achieved the best performance with an accuracy of 98.3%, an f1-score of 98.3%, a precision of 98.4%, a recall of 98.3%, and an AUC of 99.9%. The other models achieved varying levels of accuracy, with RF achieving 88.7%, Tree achieving 89.0%, Gradient Boosting achieving 86.3%, KNN achieving 77.3%, ANN achieving 74.7%, Naive Bayes achieving 71.7%, SVM achieving 73.7%, LR achieving 73.0%, and SGD achieving 71.3%. The study demonstrates the importance of comparing multiple models and selecting the best-performing one for BC classification.



The algorithm for AdaBoost classification is described as follows [29].

---

**Algorithm 2.** Real AdaBoost

---

**Initialize:** Start with weights  $w_i = 1/N$ ,  $i = 1, 2, \dots, N$ .

**For  $m = 1, 2, \dots, M$ , repeat steps 1-3:**

**Step 1:** Fit the classifier to obtain a class probability estimate  $p_m(x) = p_w^{\wedge}(y=I/x) \in [0,1]$ , using weights  $w_i$  on the training data.

**Step 2:** Set  $f_m(x) \leftarrow \frac{1}{2} \log p_m(x) / (1-p_m(x)) \in R$ .

**Step 3:** Set  $w_i \leftarrow \exp[-y_i f_m(x_i)]$ ,  $i = 1, 2, \dots, N$ , and renormalize so that  $\sum_i w_i = 1$ .

**Output the classifier**  $\text{sign} \left[ \sum_{m=1}^M f_m(x) \right] = 1$

---

**2.2.3. GentleBoost Trees.** It is an ML method used to improve the performance of DTs on binary classification problems. GentleBoost is known for its robustness and simplicity, and it is particularly effective when dealing with noisy data or outliers.

The GentleBoost algorithm works by iteratively adding weak classifiers (in this case, decision trees) to the model in a way that minimizes the overall error.

The algorithm for GentleBoost classification is described as follows [29].

---

**Algorithm 3.** GentleBoost

---

**Initialize:** Start with weights  $w_i = 1/N$ ,  $i = 1, 2, \dots, N$ ,  $F(x) = 0$ .

**For  $m = 1, 2, \dots, M$ , repeat steps 1-3:**

**Step 1:** Fit the regression function  $f_m(x)$  by *weighted least-squares* of  $y_i$  to  $x_i$  with weight  $w_i$ .

**Step 2:** Update  $F(x) \leftarrow F(x) + f(x)$ .

**Step 3:** Update  $w_i \leftarrow w_i \exp[-y_i f_m(x_i)]$  renormalize.

**Output the classifier**  $\text{sign} [F(x)] = \text{sign} \left[ \sum_{m=1}^M f_m(x) \right]$ .

---

**2.2.4. LogitBoost Trees.** It is a boosting algorithm used for binary classification problems. It was introduced by Jerome Friedman, Trevor Hastie, and Robert Tibshirani in 1998 [29]. The algorithm is based on additive logistic regression and uses decision trees as base learners. The main idea behind LogitBoost is to iteratively fit a simple model (like a decision stump) to the current residuals, then add this model to the ensemble, and update the residuals. The process is repeated until a stopping criterion is met.

The LogitBoost algorithm has several advantages. Firstly, it can handle noisy and complex datasets and achieve high accuracy. Secondly, it is robust to overfitting and can generalize well to new data. Finally, it is computationally efficient and can handle large datasets. However, LogitBoost also has some limitations. One of the main limitations is that it can be sensitive to outliers in the data. Additionally, the quality of the predictions can be affected by the choice of hyperparameters such as the number of weak classifiers and the learning rate.

The LogitBoost algorithm was utilized for BC classification and showed significant results compared to other methods; i.e., Study [30] compared the performance of several ML models in classifying tumors as metastatic or non-metastatic using two different datasets (Vijver dataset and Wang dataset). The study evaluated the performance of Logitboost, LR, SVM, Tree, Adaboost, and RF models. The results showed that the performance of the models varied depending on the dataset used. With the Vijver's dataset, the models achieved moderate to good accuracy, with Logitboost achieving the highest accuracy of 79% and an AUC of 0.810. SVM attained commendable results in terms of accuracy and AUC values, achieving an accuracy rate of 77.1% and an AUC of 0.806. Adaboost also performed well, achieving an accuracy of 77.7% and an AUC of 0.782. However, the accuracy and AUC values of the other models were relatively lower. With the Wang dataset, the models achieved higher accuracy and AUC values, with Logitboost achieving the highest accuracy of 89.7% and an AUC of 0.923. RF achieved high accuracy and AUC values, with an accuracy of 87.6% and an AUC of 0.915. Adaboost performed well, achieving an accuracy of 86.3% and an AUC of 0.893. SVM and Tree also achieved moderate to good accuracy and AUC values, while LR achieved relatively lower accuracy and AUC values. The results suggest that Logitboost, SVM, RF, and Adaboost are effective models for the Wang dataset, while Logitboost, SVM, and Adaboost are effective models for the Vijver dataset. However, it is important to consider the limitations of the study, such as the relatively small sample sizes and limited number of features used in the datasets.

The algorithm for LogitBoost classification is described as follows [29].

---

**Algorithm 4.** LogitBoost

---

**Initialize:** Start with weights  $w_i = 1/N$ ,  $i = 1, 2, \dots, N$ ,  $F(x) = 0$  and probability estimates  $p(x_i) = \frac{1}{2}$ .

**For  $m = 1, 2, \dots, M$ , repeat steps 1-3:**

**Step 1:** Compute the working response and weights

$$Z_i = \frac{(y_i - P(x_i))}{P(x_i)(1-P(x_i))}$$

**Step 2:** Fit the function  $f(x)$  by a weighted least-square regression of  $Z_i$  to  $x_i$  using weights  $w_i$ .

**Step 2:** update  $F(x) \leftarrow F(x) + \frac{1}{2} f_m(x)$  and  $P(x) \leftarrow (e^{F(x)}) / (e^{F(x)} + e^{-F(x)})$ .

**Output the classifier**  $\text{sign}[F(x)] = \text{sign}[\sum_{m=1}^M f_m(x)]$ .

---

**2.2.5. RUSBoost Trees.** It is a hybrid ML algorithm that combines Random Under-Sampling (RUS) and AdaBoost to handle imbalanced data classification problems. It was proposed in study [31] in 2010. The algorithm is designed to improve the performance of AdaBoost on imbalanced datasets by integrating a data sampling strategy. The RUSBoost algorithm exhibits several strengths. Primarily, it is capable of managing imbalanced datasets and achieving exceptional accuracy for the minority class. Moreover, it is resilient to overfitting and can generalize effectively to novel data. Lastly, it is computationally efficient and can handle sizable datasets. Despite these advantages, RUSBoost has certain drawbacks. Foremost among these is its susceptibility to noise and outliers in the data. Additionally, the quality of its predictions can be affected by hyperparameter selection, such as the number of weak classifiers and the size of the randomly selected negative class samples.

RUSBoost and SMOTE are used by several studies to handle imbalanced datasets; i.e., Study [32] aimed to examine the performance of two methods, RUSBoost and SMOTE-Boosted C5.0, for handling the problem of an imbalanced WDBC dataset for the classification of BC. The results showed that RUSBoost outperformed SMOTE-boosted C5.0 in terms of accuracy, sensitivity, and specificity. With RUSBoost, the study achieved an accuracy of 94.4%, a sensitivity of 93%, and a specificity of 95.4%. On the other hand, with SMOTE-Boosted C5.0, the study achieved

an accuracy of 92.5%, a sensitivity of 93.9%, and a specificity of 91.15%. These results suggest that RUSBoost is a more effective method for handling the imbalanced dataset in this context.

The algorithm for RUSBoost classification is described as follows [31].

---

**Algorithm 5.** RUSBoost

---

**Given:** Set  $S$  of examples  $(x_1, y_1), \dots, (x_m, y_m)$  with minority class  $y^f \in Y, |Y| = 2$

Weak learner, *WeakLearn*

Number of iterations,  $T$

The desired percentage of total instances to be represented by the minority class,

$N$

---

**Initialize:**  $D_1(i) = \frac{1}{m}$  for all  $i$ .

**Do For**  $t = 1, 2, \dots, T$ , **repeat steps 1-7:**

**Step 1:** Create a temporary training dataset  $S'_t$  with distribution  $D'_t$  using random undersampling.

**Step 2:** Call *WeakLearn*, providing it with example  $S'_t$  and their weights  $D'_t$ .

**Step 3:** Get back a hypothesis  $h_t: X \times Y \rightarrow [0, 1]$ .

**Step 4:** Calculate the pseudo-loss (for  $S$  and  $D_t$ ):

$$\epsilon_t = \sum_{(i,y): y_i \neq y} D_t(i)(1 - h_t(x_i, y_i) + h_t(x_i, y_i)).$$

**Step 5:** Calculate the weight update parameter:

$$\alpha_t = \frac{\epsilon_t}{1 - \epsilon_t}.$$

**Step 6:** Update  $D_t$ :

$$D_{t+1} = D_t(i) \alpha_t^{\frac{1}{2} (1 + h_t(x_i, y_i) - h_t(x_i, y_i; y_i \neq y))}$$

**Step 7:** Normalize  $D_{t+1}$ : Let  $Z_t = \sum_i D_{t+1}(i)$ .

$$D_{t+1}(i) = \frac{D_{t+1}(i)}{Z_t}$$

**Output the final hypothesis:**  $H(x) = \operatorname{argmax}_{y \in Y} \sum_{t=1}^T h_t(x_i, y_i) \log \frac{1}{\alpha_t}$ .

---

**2.3. Current methods of parameter tuning.** Parameter tuning is a crucial step in the process of building an ML model. It involves selecting the optimal values for the parameters of a model to improve its performance. The current methods of parameter tuning can be broadly categorized into Grid search and Bayesian optimization.

**2.3.1. Grid Search.** It is a traditional method for hyperparameter tuning. It involves specifying a subset of the hyperparameter space as a grid, and then systematically checking every point in the grid. For each combination of parameters, the model is trained, and its performance is measured. The main disadvantage of grid search is that it can be computationally expensive, especially for models with a large number of parameters. The following presents several studies that investigated the use of grid search to improve the performance of various ML models [33 – 36].

In study by [33], the authors employed grid search to fine-tune the hyperparameters of nine ML models, including Naive Bayes, LR, SVM, LASSO, DT, KNN, RF, AdaBoost, and XGBoost. The objective of the study was to identify which algorithms perform best for both balanced and imbalanced datasets. The results indicated that RF and XGBoost outperformed the other algorithms when the data was less balanced, whereas SVM, LR, and LASSO performed better than the other algorithms when the data was balanced. This finding highlights the importance of selecting the appropriate ML algorithm based on the dataset's balance or imbalance.

Another study [34] utilized grid search to optimize the hyperparameters of the SVM algorithm for BC classification. The authors compared the performance of SVM with and without grid search and found that grid search significantly improved the recall and precision of the SVM algorithm. The recall and precision were 83% and 61%, respectively, without grid search, while they were 95% and 95%, respectively, with grid search. This result suggests that hyperparameter tuning using grid search can enhance the performance of SVM for BC classification.

Similarly, study [35] employed grid search to optimize the hyperparameters of the RF algorithm for BC classification. The authors compared the performance of RF with and without grid search and found that grid search improved the recall, precision, and F1 score of the RF algorithm. The recall, precision, and F1 scores were 96% without grid search, while they were 97% with grid search. This result supports the effectiveness of hyperparameter tuning using grid search in enhancing the performance of ML algorithms in various applications.

Finally, study [36] used grid search to optimize the hyperparameters of the KNN algorithm for BC classification. The authors compared the performance of KNN with grid search and default tuning and found that grid search significantly improved the accuracy of the KNN algorithm. The accuracy was 94.35% with grid search, while it was 90.10% with default tuning. This result emphasizes the importance of hyperparameter tuning

using grid search in improving the performance of KNN for BC classification.

These studies demonstrate the effectiveness of hyperparameter tuning using grid search in enhancing the performance of ML algorithms for BC classification. The results highlight the importance of selecting the appropriate algorithm and tuning the hyperparameters for the specific dataset.

**2.3.2. Bayesian Optimization.** It is a more advanced method for hyperparameter tuning. It builds a probabilistic model of the function mapping from hyperparameter values to the objective evaluated on a validation set. By using this model, the algorithm can choose the most promising hyperparameters to evaluate in the true objective function. This method is more efficient than grid search and random search, especially for high-dimensional hyperparameter spaces [7, 37, 38].

Several studies have investigated the use of Bayesian optimization to enhance the performance of various ML models. In one study [7], a comprehensive comparative analysis was conducted on different ML models using various hyperparameter optimization methods, including Bayesian, grid search, and random search optimization. The findings revealed that the Bayesian hyperparameter optimization method was more stable than grid search and random search methods. Additionally, the XGBoost algorithm achieved a high accuracy of 94.74% and a sensitivity of 93.69%. In another study [37], a hybrid feature selection approach was implemented along with Bayesian hyperparameter tuning, resulting in the Extra tree classifier algorithm achieving the best accuracy of 96.2%. In a third study [38] a performance comparison was conducted on several ML algorithms, including SVM, DT, Naive Bayes, KNN, and Ensemble Classifiers, and the Bayesian optimization algorithm was applied to all classifiers to maximize the prediction accuracy. The results showed that the Bayesian optimization-based KNN algorithm outperformed the other ML algorithms, achieving an accuracy of 95.833%. Overall, these studies demonstrate the importance of selecting the appropriate optimization method and tuning hyperparameters to improve the performance of ML algorithms.

**2.4. Research gap.** The literature has shown that the performance of machine learning models heavily relies on the selection of appropriate hyperparameters. While several studies have investigated the use of various optimization methods to tune these hyperparameters, there is a research gap in exploring the potential benefits of Bayesian hyperparameter optimization for iterative tuning of Tree-Ensemble-Based machine learning models.

Tree-Ensemble-Based models, such as AdaBoost, Gentle-Boost, LogitBoost, Bag, and RUSBoost, are commonly used in various applications, including classification and regression tasks. However, the optimal hyperparameters for these models are not always known and can be challenging to determine given the large number of possible combinations.

Bayesian optimization is a promising approach for hyperparameter tuning that has been shown to outperform other optimization techniques in various applications. However, to the best of our knowledge, there is no investigation in the literature that explores the use of Bayesian hyperparameter optimization for iterative tuning of Tree-Ensemble-Based machine learning models.

Therefore, the research gap in the literature is the lack of studies that investigate the potential benefits and limitations of using Bayesian hyperparameter optimization for iterative tuning of Tree-Ensemble-Based machine learning models, such as AdaBoost, Gentle-Boost, LogitBoost, Bag, and RUSBoost. This research gap highlights the need for further exploration of this approach to improve the performance of these models in various applications.

**3. Methodology.** The aim of this study is to develop an iterative machine learning approach based on tree ensemble-based models with Bayesian hyperparameter tuning. The methodology involves the following steps.

**3.1. Data collection and preparation.** This study utilized three BC datasets, namely the WDBC, BCCD and the SEER BC dataset. The WDBC, BCCD, and SEER BC datasets are distinct from one another and have been utilized for different classification purposes, rather than being employed for the same classification task. Therefore, these datasets do not intersect. In the case of the WDBC dataset, the target class is labeled as "classification" and pertains to determining whether a tumor is malignant or benign, as presented in Table 2. On the other hand, the BCCD dataset assigns the target class as "Diagnosis," indicating the presence or absence of breast cancer, as specified in Table 3. Lastly, the SEER breast cancer dataset employs a target class called "STATUS," which indicates whether the patient is alive or deceased, as described in Table 4.

WDBC dataset is a well-known dataset used for breast cancer classification tasks. It contains 569 samples, each of which corresponds to a breast mass detected in a patient. Each sample is described by 30 different features, which provide information about the characteristics of the mass [39]. Table 2 shows a brief description of each feature in the dataset.

Table 2. WDBC dataset description

No.	Feature	Description
1	radius_mean	The mean radius of the mass
2	texture_mean	The mean texture of the mass.
3	perimeter_mean	The mean perimeter of the mass
4	area_mean	The mean area of the mass
5	smoothness_mean	The mean smoothness of the mass
6	compactness_mean	The mean compactness of the mass
7	concavity_mean	The mean concavity of the mass
8	concave points_mean	The mean number of concave points on the mass
9	symmetry_mean:	The mean symmetry of the mass
10	fractal_dimension_mean	The mean fractal dimension of the mass
11	radius_se	The standard error of the radius of the mass
12	texture_se	The standard error of the texture of the mass
13	perimeter_se	The standard error of the perimeter of the mass
14	area_se	The standard error of the area of the mass
15	smoothness_se	The standard error of the smoothness of the mass
16	compactness_se	The standard error of the compactness of the mass
17	concavity_se	The standard error of the concavity of the mass
18	concave points_se	The standard error of the number of concave points on the mass
29	symmetry_se	The standard error of the symmetry of the mass
20	fractal_dimension_se	The standard error of the fractal dimension of the mass
21	radius_worst	The worst (largest) radius of the mass
22	texture_worst	The worst (most irregular) texture of the mass
23	perimeter_worst	The worst (largest) perimeter of the mass
24	area_worst	The worst (largest) area of the mass
25	smoothness_worst	The worst (least smooth) smoothness of the mass
25	compactness_worst	The worst (most compact) compactness of the mass
26	concavity_worst	The worst (most severe) concavity of the mass
27	concave points_worst	The worst (most severe) number of concave points on the mass
28	symmetry_worst	The worst (least symmetrical) symmetry of the mass
29	fractal_dimension_worst	The worst (most irregular) fractal dimension of the mass
30	Classification	Malignant (cancerous) or benign (non-cancerous)

BCCD is a dataset used for BC classification tasks. It contains 116 (64 patients and 52 healthy controls) samples, each of which corresponds to



a breast mass detected in a patient. Each sample is described by 10 different features, which provide information about the characteristics of the mas [40]. Table 3 shows a brief description of each feature in the dataset.

Table 3. BCCD dataset description

No.	Feature	Description
1	Age	The age of the patient at the time of diagnosis
2	BMI	The body mass index of the patient
3	Glucose	The fasting plasma glucose concentration of the patient
4	Insulin	The fasting serum insulin level of the patient
5	HOMA	The homeostasis model assessment (HOMA) index, which estimates insulin resistance
6	Leptin	The concentration of leptin, a hormone involved in regulating energy balance
7	Adiponectin	The concentration of adiponectin, a hormone involved in regulating glucose levels
8	Adiponectin	The concentration of resistin, a hormone involved in regulating insulin sensitivity
9	MCP-1	The concentration of monocyte chemoattractant protein-1, a cytokine involved in inflammation and immune response
10	Diagnosis	Presence or absence of breast cancer

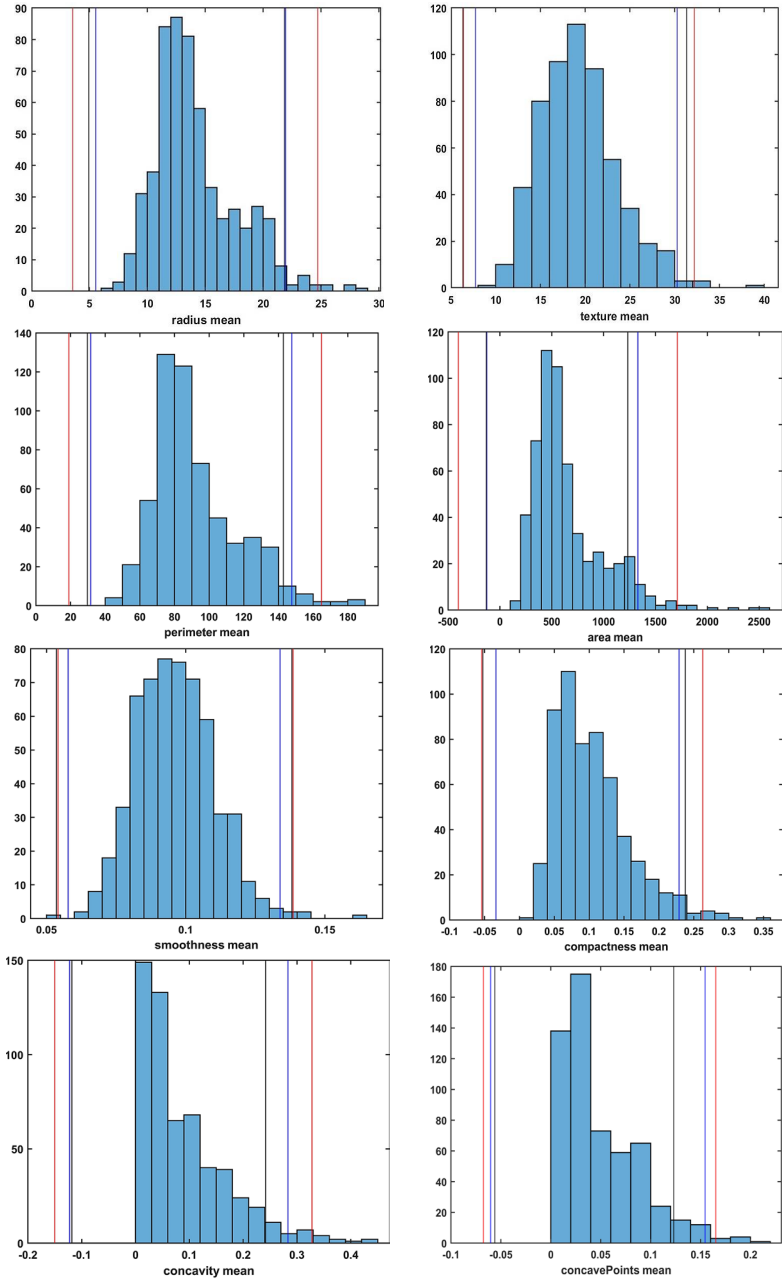
The SEER BREAST CANCER dataset is a dataset used for survival analysis tasks of breast cancer patients. It contains information on patients diagnosed with breast cancer between 2006 and 2010 and includes 4024 instances, of which 3408 are alive and 616 are deceased. Each instance is described by 15 different features, which provide information about the characteristics of the patients and their cancer [41]. Table 4 shows a brief description of each feature in the dataset:

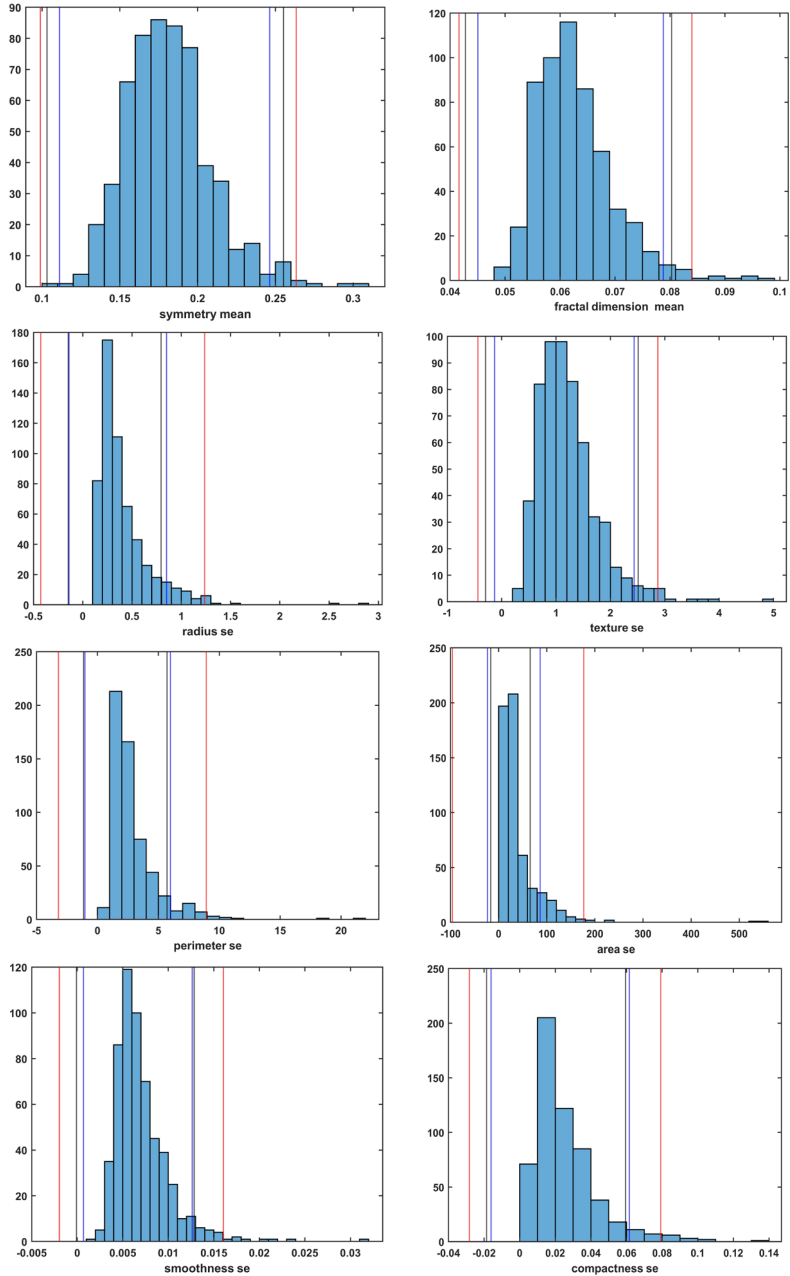
To prepare the datasets for analysis, the study used several preprocessing techniques. One of the preprocessing techniques used is the Synthetic Minority Over-sampling Technique (SMOTE) to address the imbalance problem that was obvious in the three datasets. SMOTE generates synthetic samples for the minority class to balance the dataset and improve the performance of the classification models.

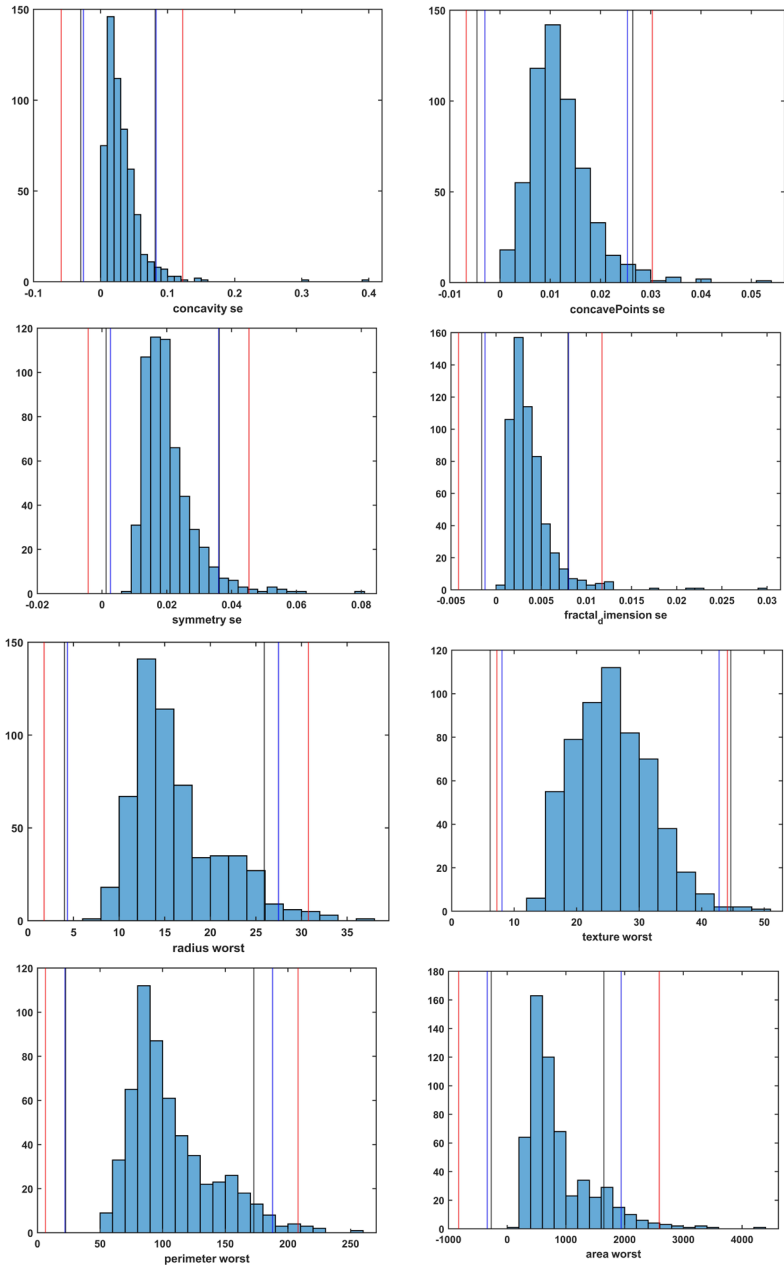
Table 4. SEER BC dataset description

No.	Feature	Type
1	AGE	The age of the patient at the time of diagnosis
2	RACE	The race of the patient
3	MARITAL STATUS	The marital status of the patient
4	T STAGE	The size and extent of the primary tumor at diagnosis
5	N STAGE	The spread of cancer to nearby lymph nodes at diagnosis
6	6TH STAGE	The stage of the cancer at diagnosis, based on the TNM system
7	GRADE	The grade of the tumor, indicating how abnormal the cancer cells look under a microscope
8	A STAGE	The stage of the cancer at diagnosis, based on a different staging system
9	TUMOR SIZE	The size of the tumor at diagnosis, in millimeters
10	ESTROGEN STATUS	The status of the estrogen receptor in the tumor
11	PROGESTERONE STATUS	The status of the progesterone receptor in the tumor
12	REGIONAL NODES EXAMINED	The number of lymph nodes examined during surgery
13	REGIONAL NODES POSITIVE	The number of lymph nodes with cancer cells found during surgery
14	SURVIVAL MONTHS	The number of months between diagnosis and last follow-up or death
15	STATUS (classification)	Alive or Dead

Additionally, the study removed outliers from the WDBC and SEER BC datasets using the method of three standard deviations (3 SD) above and below the mean. This method removes extreme data points that may skew the analysis or modeling results. Figures 1, 2 display a comparative analysis of three distinct outlier detection techniques. The first technique employed in the analysis is the three standard deviations (3-SD) above and below the mean, which are depicted in red. The second technique involves 1.5 times the interquartile range (IQR) above or below the third and first quartiles and is represented by a blue color. The third technique is based on three scaled median absolute deviations above and below the median and is displayed in black.







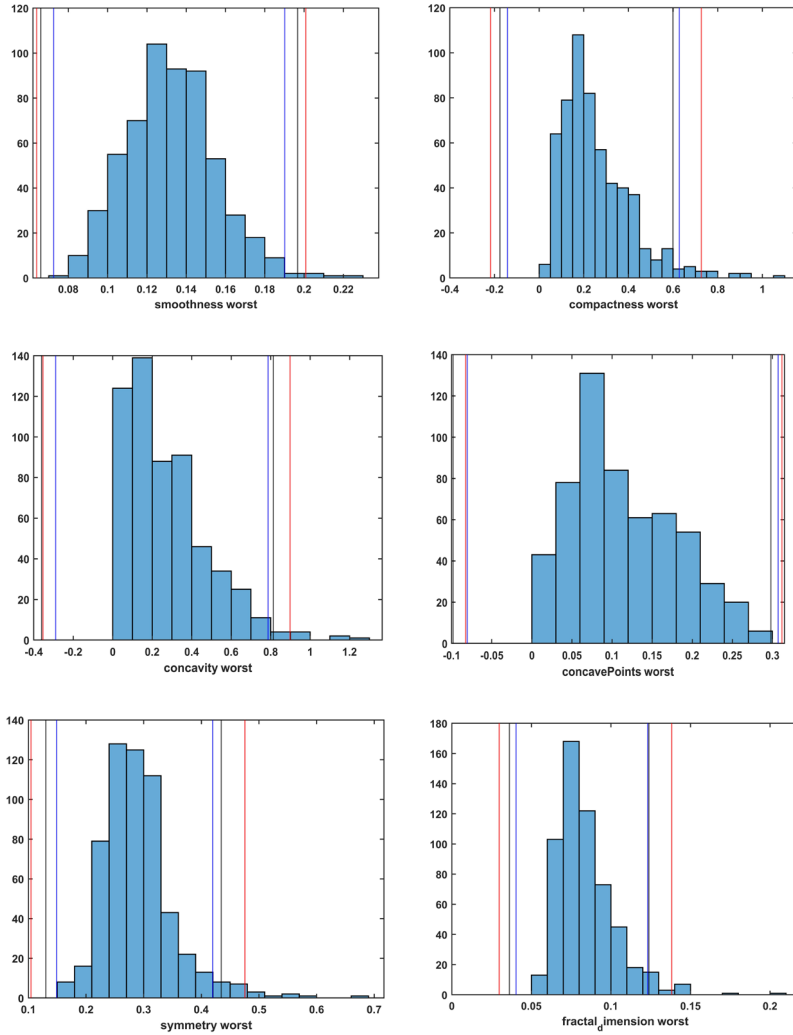


Fig. 1. Outlier detection techniques for the WDBC dataset

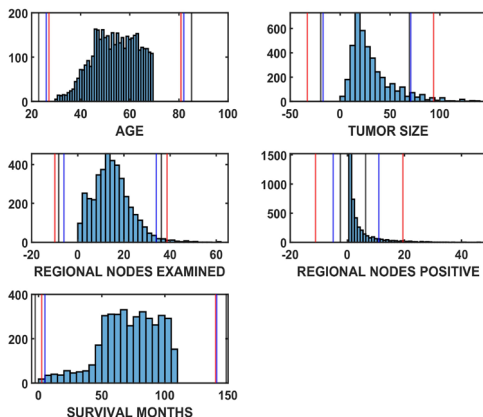


Fig. 2. Outlier detection techniques for the SEER dataset

**3.2. The proposed iterative process.** The study used five tree ensemble-based models: AdaBoost, GentleBoost, LogitBoost, Bag, and RUSBoost. Each model was trained with the default hyperparameters and with iterative tuning using Bayesian hyperparameters tuning. Figure 3 shows the workflow of the proposed iterative training process while the iterative tuning process involved the following steps.

---

**Algorithm 6.** Steps of iterative tuning process

---

**Let**  $i$  be the number of the models  $i$  [1,2,3,4,5].

**Let**  $N$  be the number of iterations.

---

**Step 1.** Split the dataset into training and validation sets.

**Step 2.** Train the model $_i$  with the default hyperparameters on the training set and evaluate its performance on the validation set.

**Step 3.** Use Bayesian hyperparameters tuning to select the best hyperparameters for the model based on the performance on the validation set.

**Step 4.** Train the model with the selected hyperparameters on the training set and evaluate its performance on the validation set.

**Step 5.** Repeat the above steps until the performance on the validation set no longer improves or a maximum number of iterations is reached.

**Step 6.** Repeat the above steps until the performance on the validation set no longer improves or a maximum number of iterations is reached.

**Step 7.** If the performance of model $_i$  best than model $_{i-1}$  then set the best result = the performance of model $_i$ .

**Step 8.** Repeat steps 1-7 for all  $i$ .

**Step 9.** Repeat steps 1-7 for  $N$  iteration.

---

**Output** The final prediction result including the Method name, best performance metrics, and the optimal hyperparameters.

---

**3.3. Evaluation.** The performance evaluation of each ML model in the study is conducted based on their effectiveness in predicting the target class. To assess their performance, a range of metrics, including accuracy, precision, recall, and F1-score, are employed. These metrics provide a comprehensive evaluation of the predictive capabilities and overall performance of the ML models. The calculation details of these evaluation metrics can be found in Table 5, which illustrates how each metric is computed and provides further insights into the model performance.

Table 5. Evaluation Metrics Equations

Measure	Equation
Accuracy	$(TP+TN)/(TP+TN+FN+FP)$
Precision	$TP/(TP+FP)$
Recall	$TP/(TP+FN)$
F1 score	$(2 \times \text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall})$

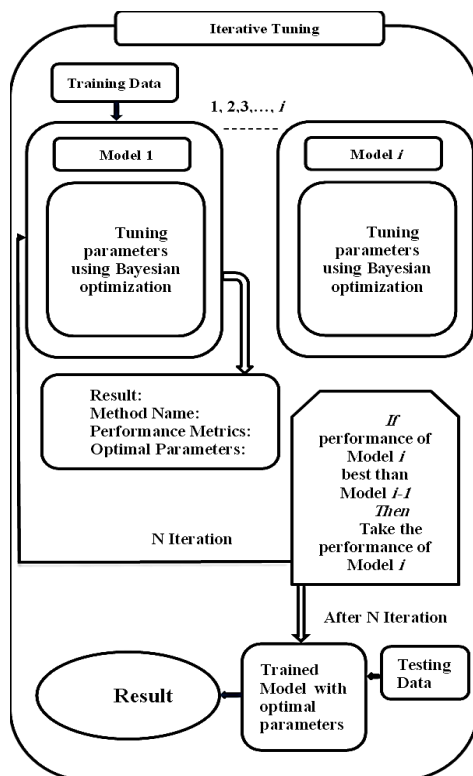


Fig. 3. Iterative tuning workflow



**4. Result and Discussion.** The experiments were conducted on three different datasets: WDBC, SEER BC, and BCCD. In the case of the WDBC dataset, the results showed that Gentle-Boost and AdaBoost achieved the highest accuracy of 100% across multiple iterations. These algorithms outperformed other models such as LogitBoost, which achieved an accuracy of 99.1%, Bagged trees with 98.2% accuracy, and RUSBoost with 95.5% accuracy. The detailed results can be found in Table 6. Moving on to the SEER BC dataset, Gentle-Boost demonstrated superior performance compared to the other models in all experiments with varying iterations. It consistently outperformed the rest and achieved the highest accuracy of 96% with 100 iterations. These findings are presented in Table 8. For the BCCD dataset, the Bagged trees algorithm stood out by achieving the highest performance. It attained an accuracy of 94.7% in the case of 60 iterations. The detailed results for this dataset can be found in Table 10.

**4.1. Discussion of the results obtained by implementing the proposed framework on the WDBC.** Table 6 shows the best accuracy achieved by the proposed iterative tuning of the tree ensemble-based model using Bayesian hyperparameter tuning. The results displayed in Table 7 show the performance of several tree ensemble-based algorithms applied to the WDBC dataset for different numbers of iterations. The results show that the various tree ensemble-based algorithms achieve very high accuracy, precision, recall, and F1-score values, indicating that they are generally effective in classifying the WDBC dataset. The Gentle-Boost algorithm appears to be the most effective, achieving the highest performance in six of the 12 cases, including the highest accuracy and F1-score values for 10, 30, 40, 60, 90, 110, and 100 iterations. AdaBoost also performed well, achieving the highest performance for 20 and 80 iterations. LogitBoost achieved the highest performance over 70 iterations, whereas RUSBoost achieved the highest performance over 50 iterations. Finally, the Bagged trees algorithm achieved the highest performance for 120 iterations. The performance of the different algorithms varied depending on the number of iterations. For example, AdaBoost achieved the highest performance for 20 and 80 iterations, but its performance was not as good for other numbers of iterations. Similarly, RUSBoost achieved the highest performance for 50 iterations, but its performance dropped sharply for higher or lower numbers of iterations. Overall, the results suggest that the Gentle-Boost algorithm is robust and effective for classifying the WDBC dataset. However, the choice of algorithm may depend on the specific application and the number of iterations required.

Table 6. The accuracy achieved in different numbers of iterations of the WDBC dataset

No. Iteration	AdaBoost	GentleBoost	LogitBoost	Bagged trees	RUSBoost
10 Iteration	-	100	-	-	-
20 Iteration	100	-	-	-	-
30 Iteration	-	99.1	-	-	-
40 Iteration	-	99.1	-	-	-
50 Iteration	-	-	-	-	95.5
60 Iteration	-	99.1	-	-	-
70 Iteration	-	-	99.1	-	-
80 Iteration	99.1	-	-	-	-
90 Iteration	-	99.1	-	-	-
100 Iteration	-	100	-	-	-
110 Iteration	-	100	-	-	-
120 Iteration	-	-	-	98.2	-

Table 7. Evaluation of the performance of the proposed framework applied to the WDBC dataset

No. Iteration	Model	Evaluation Metrix			
		Accuracy	Precision	Recall	F1-Score
10 Iteration	GentleBoost	100	100	100	100
20 Iteration	AdaBoost	100	100	100	100
30 Iteration	GentleBoost	99.1	100	98.3	99.1
40 Iteration	GentleBoost	99.1	100	98.3	99.1
50 Iteration	RUSBoost	95.5	94.3	96.6	95.4
60 Iteration	GentleBoost	99.1	100	98.3	99.1
70 Iteration	LogitBoost	99.1	100	98.3	99.1
80 Iteration	AdaBoost	99.1	100	98.3	99.1
90 Iteration	GentleBoost	99.1	100	98.3	99.1
100 Iteration	GentleBoost	100	100	100	100
110 Iteration	GentleBoost	100	100	100	100
120 Iteration	Bagged trees	98.2	100	96.6	98.3

**4.2. Discussion of the results obtained by implementing the proposed framework on the SEER BC dataset.** Table 8 shows the best accuracy, whereas Table 9 displays the performance achieved by the proposed methodology applied to the SEER BC dataset for different numbers of iterations. The results show that the Gentle-Boost algorithm achieves high accuracy, precision, recall, and F1-score values, indicating that it is effective for classifying the SEER BC dataset. It is interesting to note that the performance of the Gentle-Boost algorithm is consistently high across all different numbers of iterations. In particular, the algorithm achieved the highest performance in all 12 cases, with an accuracy ranging from 90.1% to 96%, a precision ranging from 96.4% to 98%, a recall ranging from 93.8% to 94.2%, and an F1-score ranging from 95% to 96%. Compared with the results obtained for the WDBC dataset, the performance of the Gentle-Boost algorithm for the SEER BC dataset was generally lower. This is likely due to the fact that the SEER BC dataset is more complex and noisier than the WDBC dataset. In general, the results suggest that the Gentle-Boost algorithm is effective for classifying the SEER BC dataset, and that its performance is consistent across different numbers of iterations.

Table 8. The accuracy achieved in different numbers of iterations of the SEER BC dataset

No. Iteration	AdaBoost	GentleBoost	LogitBoost	Bagged trees	RUSBoost
10 Iteration	-	90.1	-	-	-
20 Iteration	-	95.3	-	-	-
30 Iteration	-	95.8	-	-	-
40 Iteration	-	95.9	-	-	-
50 Iteration	-	95.6	-	-	-
60 Iteration	-	95.7	-	-	-
70 Iteration	-	95.7	-	-	-
80 Iteration	-	95.8	-	-	-
90 Iteration	-	95.8	-	-	-
100 Iteration	-	96	-	-	-
110 Iteration	-	96	-	-	-
120 Iteration	-	95.6	-	-	-

Table 9. Evaluation of the performance of the proposed framework applied to the SEER dataset

No. Iteration	Model	Evaluation Metric			
		Accuracy	Precision	Recall	F1-Score
10 Iteration	GentleBoost	90.1	96.4	93.8	95
20 Iteration	GentleBoost	95.3	96.9	93.8	95.3
30 Iteration	GentleBoost	95.8	97.4	94.2	95.8
40 Iteration	GentleBoost	95.9	97.6	94.2	95.9
50 Iteration	GentleBoost	95.6	97.4	93.8	95.6
60 Iteration	GentleBoost	95.7	97.4	94	95.7
70 Iteration	GentleBoost	95.7	97.4	94	95.7
80 Iteration	GentleBoost	95.8	97.4	94.2	95.8
90 Iteration	GentleBoost	95.8	97.4	94.2	95.8
100 Iteration	GentleBoost	96	98	94	96
110 Iteration	GentleBoost	96	97.4	94.2	95.7
120 Iteration	GentleBoost	95.6	97.3	94	95.6

**4.3. Discussion of the results obtained by implementing the proposed framework on the BCCD.** Table 10 presents the accuracy achieved in different numbers of iterations of the BCCD dataset. The results in Table 11 show the performance of several tree ensemble-based algorithms applied to the BCCD dataset for different numbers of iterations. Among all the algorithms, the Bagged trees algorithm achieved the highest performance in the case of 60 iterations, with an accuracy of 94.7%, a precision of 90%, a recall of 100%, and an F1-score of 94.7%. The results suggest that the Gentle-Boost algorithm is generally effective in classifying the BCCD dataset, achieving the highest performance in six of the 12 cases. However, the performance of the Gentle-Boost algorithm is not consistent across different numbers of iterations. For example, the algorithm achieved high performance in cases with 10, 20, 50, and 100 iterations, but its performance decreased in cases with 70, 80, and 120 iterations. Other tree ensemble-based algorithms, such as AdaBoost, LogitBoost, and RUSBoost, also achieved high performance in some cases; however, their performance was generally less consistent than that of Gentle-Boost. For example, AdaBoost achieved the highest performance in the cases of 40 and 90 iterations, but its performance was not as good in other cases. Similarly, LogitBoost and RUSBoost achieved the highest performance in the cases of 30 and 110 iterations, respectively; however, their performance dropped off in other cases. Overall, the results suggest that the bag algorithm is

effective for classifying the BCCD dataset. However, compared to the results obtained for the SEER BC dataset and the WDBC dataset, the performance of the tree ensemble-based algorithms for the BCCD dataset was generally lower.

Table 10. The accuracy achieved in different numbers of iterations of the BCCD dataset

No. Iteration	AdaBoost	GentleBoost	LogitBoost	Bag	RUSBoost
10 Iteration	-	78.9	-	-	-
20 Iteration		78.9	-	-	-
30 Iteration	-		84.2	-	-
40 Iteration	89.5		-	-	-
50 Iteration	-	68.4	-	-	
60 Iteration	-		-	94.7	-
70 Iteration	-	84.2		-	-
80 Iteration		84.2	-	-	-
90 Iteration	78.9		-	-	-
100 Iteration	-	68.4	-	-	-
110 Iteration	-		-	-	84.2
120 Iteration	84.2	-	-		-

Table 11. Evaluation of the performance of the proposed framework applied to BCCD

No. Iteration	Model	Evaluation Metrix			
		Accuracy	Precision	Recall	F1-Score
10 Iteration	GentleBoost	78.9	70	88.9	78.3
20 Iteration	GentleBoost	78.9	70	88.9	78.3
30 Iteration	LogitBoost	84.2	70	100	82.4
40 Iteration	AdaBoost	89.5	90	88.9	89.4
50 Iteration	GentleBoost	68.4	70	66.7	68.3
60 Iteration	Bag	94.7	90	100	94.7
70 Iteration	GentleBoost	84.2	90	77.8	83.5
80 Iteration	GentleBoost	84.2	90	77.8	83.5
90 Iteration	AdaBoost	78.9	70	88.9	78.3
100 Iteration	GentleBoost	68.4	70	66.7	68.3
110 Iteration	RUSBoost	84.2	70	100	82.4
120 Iteration	GentleBoost	84.2	90	77.8	83.5

**5. Comparison of the results.** This section presents a comparative analysis of the performance of multiple machine learning models for predicting BC using publicly available datasets such as WDBC, BCCD, and SEER BC. We evaluate the performance of both state-of-the-art models and the proposed framework. The performance of each model is assessed based on accuracy, precision, recall, and F1-score, and the best-performing model for BC prediction is identified. Additionally, we compare the performance of the proposed framework to the results reported in the literature for BC prediction.

**5.1. Comparative Analysis of the Performance of Various Machine Learning Models in Predicting Breast Cancer.** Table 12 showcases the results obtained from several machine learning models that were applied to three distinct datasets: WDBC, SEER, and BCCD, with a training and testing ratio of (80:20), (80:20), and (85:15), respectively. The experimental setup for these datasets is identical to the one mentioned in Section 3.1. The performance of the machine learning models was evaluated using metrics such as accuracy, precision, recall, and F1-Score. The metrics used to evaluate the performance of these models were accuracy, precision, recall, and F1-Score.

The best-performing model across all datasets was the proposed model, with perfect scores on the WDBC dataset and impressive results on the SEER and BCCD datasets. The proposed model's F1-Score, a measure that balances precision and recall, is particularly high, indicating strong performance in both identifying positive cases and limiting false positives.

The Cubic SVM and the Narrow, Wide, and Bilayered Neural Networks also achieved perfect scores on the WDBC dataset. However, their performance on the SEER and BCCD datasets is not as strong as that of the proposed model.

The Fine, Medium, and Coarse Trees, as well as the Linear SVM, showed consistent performance across all datasets; however, their scores were generally lower than those of the aforementioned models. The fine trees performed slightly better than the Medium and Coarse Trees, indicating that a more complex decision boundary might be beneficial for these datasets.

The Gaussian SVMs and KNN models exhibited varied performance. For instance, the Fine Gaussian SVM had high recall but lower precision, indicating a higher rate of false positives. The Course KNN, on the other hand, had high precision but low recall on the BCCD dataset, indicating a higher rate of false negatives.

The Neural Networks showed strong performance, particularly the Wide Neural Network and the Bilayered Neural Network. These models

achieved perfect scores on the WDBC dataset, and performed well on the SEER and BCCD datasets.

Table 12. The performance of various ML models

ML Model Name	Dataset	Accuracy	Precision	Recall	F1-Score
Fine Tress	WDBC	94.6	92.5	96.6	94.5
	SEER	91.9	94	89.8	91.9
	BCCD	84.2	70	100	82.4
Medium Tree	WDBC	94.6	92.5	96.6	94.5
	SEER	88	90.3	85.8	88
	BCCD	84.2	70	100	82.3
Coarse Tree	WDBC	92	86.8	96.6	91.4
	SEER	82.7	96.7	68.6	80.3
	BCCD	87.9	70	88.9	78.3
Linear SVM	WDBC	97.3	100	94.9	97.4
	SEER	80.1	83.2	77	80
	BCCD	84.2	100	66.7	80
Quadratic SVM	WDBC	99.1	100	98.3	99.1
	SEER	86	92.3	79.7	85.5
	BCCD	78.9	80	77.8	78.9
Cubic SVM	WDBC	100	100	100	100
	SEER	89.2	92.5	85.9	89.1
	BCCD	73.7	80	66.7	72.7
Fine Gaussian SVM	WDBC	92	83	100	90.7
	SEER	90.9	86.3	95.4	90.6
	BCCD	63.2	40	88.9	55.2
Medium Gaussian SVM	WDBC	99.1	98	97.3	97.6
	SEER	87.7	92.3	83	87.4
	BCCD	84.2	80	88.9	84.2
Coarse Gaussian SVM	WDBC	97.3	100	94.9	97.4
	SEER	79.7	82.7	76.6	79.5
	BCCD	68.4	90	44.4	59.5
Fine KNN	WDBC	96.4	92.5	100	96.1
	SEER	94	90.9	97.1	93.9
	BCCD	78.9	80	77.8	78.9
Medium KNN	WDBC	96.4	94.3	98.3	96.3
	SEER	85.9	88	83.8	85.8
	BCCD	78.9	90	66.7	76.6

Continuation of Table 12

ML Model Name	Dataset	Accuracy	Precision	Recall	F1-Score
Coarse KNN	WDBC	95.5	96.2	94.9	95.5
	SEER	78.8	86.3	71.4	78.1
	BCCD	57.9	100	11.1	19.9
Cosine KNN	WDBC	95.5	98.1	93.2	95.6
	SEER	84.3	87	81.6	84.2
	BCCD	73.3	90	55.6	68.7
Cubic KNN	WDBC	97.3	96.2	98.3	97.2
	SEER	84.2	84.1	84.3	84.2
	BCCD	84.2	90	77.8	83.5
Weighted KNN	WDBC	98.2	98.1	98.3	98.2
	SEER	91.9	89.1	94.7	91.8
	BCCD	78.9	90	66.7	76.6
Narrow Neural Network	WDBC	100	100	100	100
	SEER	87.3	90.3	84.3	87.2
	BCCD	68.4	80	55.6	65.6
Medium Neural Network	WDBC	99.1	98.1	100	99.0
	SEER	88.1	87.8	88.5	88.1
	BCCD	73.3	80	66.7	72.7
Wide Neural Network	WDBC	100	100	100	100
	SEER	91.8	90.5	93.1	91.8
	BCCD	78.9	90	66.7	76.6
Bilayered Neural Network	WDBC	100	100	100	100
	SEER	87.3	87	87.6	87.3
	BCCD	78.9	80	77.8	78.9
Trilayered Neural Network	WDBC	100	100	100	100
	SEER	88.8	89.8	87.8	88.8
	BCCD	73.7	80	66.7	72.7
<b>The proposed</b>	<b>WDBC</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>
	<b>SEER</b>	<b>95.9</b>	<b>97.6</b>	<b>94.2</b>	<b>95.9</b>
	<b>BCCD</b>	<b>94.7</b>	<b>90</b>	<b>100</b>	<b>94.7</b>

**5.2. Comparison with previous studies.** According to Table 13, hyperparameter tuning plays a crucial role in improving the performance of ML models. The proposed framework utilizes the power of Bayesian optimization and Tree ensemble-based models in an iterative process to achieve the best possible results. Table 13 indicates that the proposed



framework outperforms the results reported in the literature for the three commonly used datasets.

Table 13. Comparison with previous studies

Ref.	Year	Dataset	Methods/Techniques	Accuracy
[10]	2019	WDBC	WQPSO with smooth SVM	98.42
[15]	2019	SEER BC	Gradient Boosting with Genetic Algorithm	75.03
[13]	2020	WDBC	Optimized FSTBSVM with Jaya optimization techniques	94.36
[16]	2020	SEER BC	Improved Random Forest - based rule extraction	80.5
[19]	2020	BCCD	GA and Gradient Boosting Classifier	79
[14]	2021	WDBC	SVM, LR, KNN, DT, Naive Bayes, and RF	SVM and RF = 96.5
[9]	2022	WDBC	Optimized SVM with Bayesian hyperparameter optimization	99.3
[18]	2022	SEER BC	RF	94.6
[9]	2022	DCCD	Optimized SVM with Bayesian hyperparameter optimization	76.9
<b>The proposed</b>	<b>2023</b>	<b>WDBC</b>	<b>Tree ensemble-based models with iterative tuning of Bayesian optimization</b>	<b>100</b>
		<b>SEER</b>		<b>95.9</b>
		<b>BCCD</b>		<b>94.7</b>

**6. Conclusion.** This paper presents a method for the iterative tuning of tree-ensemble-based model parameters using Bayesian hyperparameter tuning for BC prediction. The proposed method achieved perfect scores on the WDBC dataset and impressive results on the SEER and BCCD datasets. The results show that the proposed method can significantly improve the performance of tree ensemble-based models for BC prediction. By utilizing Bayesian hyperparameter tuning, we were able to identify the optimal hyperparameter values for the models, resulting in a higher accuracy. The optimized tree ensemble-based models' high F1-Score indicates their effectiveness in identifying positive cases and limiting false positives, making them a robust and reliable option for generalization. Compared with various state-of-the-art models, the proposed method is more efficient. The outcomes of this study have important implications for medical professionals, as the early detection of BC can significantly increase the chances of survival. By utilizing ML models, such as tree ensemble-based models, and optimizing their hyperparameters using Bayesian hyperparameter tuning, medical professionals can improve their ability to

detect BC in its early stages, leading to better patient outcomes. Overall, this study provides a valuable contribution to the field of BC prediction using ML, and its proposed method can be extended to other domains where tree ensemble-based models are used.

## References

1. Rabiei R., Ayyoubzadeh S.M., Sohrabei S., Esmaceli M., Atashi A. Prediction of breast cancer using machine learning approaches. *Journal of Biomedical Physics and Engineering*. 2022. vol. 12. no. 3. pp. 297–308. DOI: 10.31661/jbpe.v0i0.2109-1403.
2. Visvanathan K., Fabian C.J., Bantug E., Brewster A.M., Davidson N.E., DeCensi A., et al. Use of endocrine therapy for breast cancer risk reduction: ASCO clinical practice guideline update. *Journal of clinical oncology*. 2019. vol. 37. no. 33. pp. 3152–3165.
3. Thirthagiri E., Lee S.Y., Kang P., Lee D.S., Toh G.T., Selamat S., Yoon S.Y., Mohd Taib N.A., Nhong M.K., Yip C.H., Teo S.H. Evaluation of BRCA1 and BRCA2 mutations and risk-prediction models in a typical Asian country (Malaysia) with a relatively low incidence of breast cancer. *Breast Cancer Research*. 2008. vol. 10. no. 4. pp. 1–12. DOI: 10.1186/bcr2118.
4. Monirujjaman Khan M., Islam S., Sarkar S., Ayaz F.I., Kabir M.M., Tazin T., Albraikan A.A., Almalki F.A. Machine learning based comparative analysis for breast cancer prediction. *Journal of Healthcare Engineering*. 2022. vol. 2022. DOI: 10.1155/2022/4365855.
5. Sharma D., Kumar R., Jain A. Breast Cancer Patient Classification from Risk Factor Analysis Using Machine Learning Classifiers. *Emergent Converging Technologies and Biomedical Systems: Select Proceedings of ETBS 2021*. 2022. pp. 491–504.
6. Ali S., Majid A., Javed S.G., Sattar M. Can-CSC-GBE: Developing Cost-sensitive Classifier with Gentleboost Ensemble for breast cancer classification using protein amino acids and imbalanced data. *Computers in biology and medicine*. 2016. vol. 73. pp. 38–46.
7. Gao L., Ding Y. Disease prediction via Bayesian hyperparameter optimization and ensemble learning. *BMC research notes*. 2020. vol. 13. pp. 1–6.
8. Sharma S.K., Vijayakumar K., Kadam V.J., Williamson S. Breast cancer prediction from microRNA profiling using random subspace ensemble of LDA classifiers via Bayesian optimization. *Multimedia Tools and Applications*. 2022. vol. 81. no. 29. pp. 41785–41805.
9. Rasool A., Bunterngchit C., Tiejian L., Islam M.R., Qu Q., Jiang Q. Improved machine learning-based predictive models for breast cancer diagnosis. *International journal of environmental research and public health*. 2022. vol. 19(6). no. 3211. DOI: 10.3390/ijerph19063211.
10. Latchoumi T.P., Ezhilarasi T.P., Balamurugan K. Bio-inspired weighed quantum particle swarm optimization and smooth support vector machine ensembles for identification of abnormalities in medical data. *SN Applied Sciences*. 2019. vol. 1(10). no. 1137. DOI: 10.1007/s42452-019-1179-8.
11. Lahoura V., Singh H., Aggarwal A., Sharma B., Mohammed M.A., Damasevicius R., Kadry F., Cengiz K. Cloud computing-based framework for breast cancer diagnosis using extreme learning machine. *Diagnostics*. 2021. vol. 11(2). no. 241. DOI: 10.3390/diagnostics11020241.
12. Showrov M.I.H., Islam M.T., Hossain M.D., Ahmed M.S. Performance comparison of three classifiers for the classification of breast cancer dataset. in *4th International*

- conference on electrical information and communication technology (EICT). 2019. pp. 1–5.
13. Sheth P.D., Patil S.T., Dhore M.L. Evolutionary computing for clinical dataset classification using a novel feature selection algorithm. *Journal of King Saud University-Computer and Information Sciences*. 2022. vol. 34. pp. 5075–5082.
  14. Ara S., Das A., Dey A. Malignant and benign breast cancer classification using machine learning algorithms. in *2021 International Conference on Artificial Intelligence (ICAI)*. 2021. pp. 97–101.
  15. Lu H., Wang H., Yoon S.W. A dynamic gradient boosting machine using genetic optimizer for practical breast cancer prognosis. *Expert Systems with Applications*. 2019. vol. 116. pp. 340–350.
  16. Wang S., Wang Y., Wang D., Yin Y., Wang Y., Jin Y. An improved random forest-based rule extraction method for breast cancer diagnosis. *Applied Soft Computing*. 2020. vol. 86. no. 105941. DOI: 10.1016/j.asoc.2019.105941.
  17. Ozkan G.Y., Gunduz S.Y. Comparison of Classification Algorithms for Survival of Breast Cancer Patients. *Innovations in Intelligent Systems and Applications Conference (ASYU)*. 2020. pp. 1–4.
  18. Haque M.N., Tazin T., Khan M.M., Faisal S., Ibraheem S.M., Algethami H., Almalki F.A. Predicting characteristics associated with breast cancer survival using multiple machine learning approaches. *Computational and Mathematical Methods in Medicine*. 2022. vol. 2022. pp. 1–12. DOI: 10.1155/2022/1249692.
  19. Mishra A.K., Roy P., Bandyopadhyay S. Genetic algorithm based selection of appropriate biomarkers for improved breast cancer prediction. *Intelligent Systems and Applications: Proceedings of the Intelligent Systems Conference (IntelliSys)*. 2020. vol. 2. pp. 724–732.
  20. Mishra A.K., Roy P., Bandyopadhyay S. Binary particle swarm optimization based feature selection (bpso-fs) for improving breast cancer prediction. *Proceedings of International Conference on Artificial Intelligence and Applications: ICAIA 2020*. 2021. pp. 373–384.
  21. Breiman L. Bagging predictors. *Machine learning*. 1996. vol. 24. pp. 123–140.
  22. Harrou F., Saidi A., Sun Y. Wind power prediction using bootstrap aggregating trees approach to enabling sustainable wind power integration in a smart grid. *Energy Conversion and Management*. 2019. vol. 201. no. 112077. DOI: 10.1016/j.enconman.2019.112077.
  23. Buhlmann P., Yu B. Analyzing bagging. *The annals of Statistics*. 2002. vol. 30. no. 4. pp. 927–961.
  24. Liu C., Zhao Z., Gu X., Sun L., Chen G., Zhang H., Jiang Y., Zhang Y., Cui X., Liu C. Establishment and verification of a bagged-trees-based model for prediction of sentinel lymph node metastasis for early breast cancer patients. *Frontiers in Oncology*. 2019. vol. 9. no. 282. DOI: 10.3389/fonc.2019.00282.
  25. Deepa R., Gnanadesigan R., Ranjith D., Nithishkumar K., Dinesh A., Moorthy C. Performance Analysis of the Classification of Breast Cancer. *Third International Conference on Inventive Research in Computing Applications (ICIRCA)*. 2021. pp. 1–6.
  26. Dawngliani M., Chandrasekaran N., Lalmuanawma S., Thangkhanhau H. Prediction of breast cancer recurrence using ensemble machine learning classifiers. *Security with Intelligent Computing and Big-Data Services 2019: Proceedings of the 3rd International Conference on Security with Intelligent Computing and Big-data Services (SICBS)*. 2020. pp. 232–244.
  27. Assegie T.A., Tulasi R.L., Kumar N.K. Breast cancer prediction model with decision tree and adaptive boosting. *IAES International Journal of Artificial Intelligence*. 2021. vol. 10(1). no. 184.

28. Okagbue H.I., Adamu P.I., Oguntunde P.E., Obasi E.C., Odetunmbi O.A. Machine learning prediction of breast cancer survival using age, sex, length of stay, mode of diagnosis and location of cancer. *Health and Technology*. 2021. vol. 11. pp. 887–893.
29. Friedman J., Hastie T., Tibshirani R. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics*. 2000. vol. 28. no. 2. pp. 337–407.
30. Wu J., Gan M., Jiang R. A genetic algorithm for optimizing subnetwork markers for the study of breast cancer metastasis. *Seventh International Conference on Natural Computation*. 2011. pp. 1578–1582.
31. Seiffert C., Khoshgoftaar T.M., Van Hulse J., Napolitano A. RUSBoost: A hybrid approach to alleviating class imbalance. *IEEE transactions on systems, man, and cybernetics-part A: systems and humans*. 2009. vol. 40. pp. 185–197.
32. Tian J.-X., Zhang J. Breast cancer diagnosis using feature extraction and boosted C5.0 decision tree algorithm with penalty factor. *Math Biosci Eng*. 2022. vol. 19. no. 3. pp. 2193–2205.
33. Jiang X., Xu C. Deep learning and machine learning with grid search to predict later occurrence of breast Cancer metastasis using clinical data. *Journal of clinical medicine*. 2022. vol. 11(19). no. 5772.
34. Deshwal V., Sharma M. Breast cancer detection using SVM classifier with grid search technique. *International Journal of Computer Applications*. 2019. vol. 975. no. 8887.
35. Buttan Y., Chaudhary A., Saxena K. An improved model for breast cancer classification using random forest with grid search method. *Proceedings of Second International Conference on Smart Energy and Communication: ICSEC 2020*. 2021. pp. 407–415.
36. Assegie T.A. An optimized K-Nearest Neighbor based breast cancer detection. *Journal of Robotics and Control (JRC)*. 2021. vol. 2. no. 3. pp. 115–118.
37. Mate Y., Somai N. Hybrid feature selection and Bayesian optimization with machine learning for breast cancer prediction. *7th International Conference on Advanced Computing and Communication Systems (ICACCS)*. 2021. pp. 612–619.
38. Ceylan Z. Diagnosis of breast cancer using improved machine learning algorithms based on bayesian optimization. *International Journal of Intelligent Systems and Applications in Engineering*. 2020. vol. 8. no. 3. pp. 121–130.
39. kaggle Repository. Breast Cancer Wisconsin Diagnostic Dataset. Available at: <https://www.kaggle.com/code/karan1210/breast-cancer/data>. (accessed 07.06.2022).
40. kaggle Repository. Breast Cancer Coimbra Dataset. Available at: <https://www.kaggle.com/datasets/yasserhessein/breast-cancer-coimbra-data-set>. (accessed 07.06.2022).
41. Surveillance, Epidemiology, and End Results (SEER) program. SEER breast cancer data. Available at: <https://ieee-dataport.org/open-access/seer-breast-cancer-data>. (accessed 07.06.2022).

**Alsabry Ayman** — Graduate student, Department of computer science, Sana'a University. Research interests: computer science, artificial intelligence, machine learning. The number of publications — 7. [aymanalsabri@su.edu.ye](mailto:aymanalsabri@su.edu.ye); Sana'a University St., Sana'a, Yemen; office phone: +967(774)265-262.

**Algabri Malek** — Ph.D., Associate Professor, Head of the department, Department of computer science, faculty of computing and information technology, Sana'a University. Research interests: computer science, MANETs, MPLS, artificial intelligence, machine learning. The number of publications — 17. [malekye@su.edu.ye](mailto:malekye@su.edu.ye); Sana'a University St., Sana'a, Yemen; office phone: +967(777)662-909.

А. АЛСАБРИ, М. АЛЬГАБРИ

## ИТЕРАТИВНАЯ НАСТРОЙКА ПАРАМЕТРОВ МОДЕЛЕЙ НА ОСНОВЕ ДРЕВОВИДНЫХ АНСАМБЛЕЙ С ИСПОЛЬЗОВАНИЕМ БАЙЕСОВСКОЙ ОПТИМИЗАЦИИ ДЛЯ ПРОГНОЗИРОВАНИЯ РАКА МОЛОЧНОЙ ЖЕЛЕЗЫ

*Алсабри А., Альгабри М. Итеративная настройка параметров моделей на основе древовидных ансамблей с использованием байесовской оптимизации для прогнозирования рака молочной железы.*

**Аннотация.** Представлен метод итеративной настройки параметров моделей на основе ансамблей деревьев с использованием настройки байесовских гиперпараметров для прогнозирования состояний на примере рака молочной железы. Предлагаемый метод использует три различных набора данных, в том числе набор данных по диагностическому раку молочной железы Висконсина (WDBC), набор данных по надзору, эпидемиологии и конечным результатам (SEER) по раку молочной железы и набор данных по раку молочной железы в Коимбре (BCCD), а также реализует набор данных на основе древовидных ансамблей. Модели, в частности AdaBoost, Gentle-Boost, LogitBoost, Bag и RUSBoost, для прогнозирования рака молочной железы. Байесовская оптимизация использовалась для итеративной настройки гиперпараметров моделей, а производительность моделей оценивалась с использованием нескольких показателей, включая точность, прецизионность, полноту и оценку f1. Наши результаты показывают, что предложенный метод значительно повышает производительность моделей на основе ансамблей деревьев, что приводит к более высокой точности, прецизионности, полноте и оценке f1. По сравнению с другими современными моделями предлагаемый метод более эффективен. Он достиг 100% идеальных результатов по точности, прецизионности, полноте и оценке F1 в наборе данных WDBC. В наборе данных SEER BC точность метода составила 95,9%, прецизионность 97,6%, полнота 94,2% и оценка F1 95,9%. Для набора данных BCCD метод достиг точности 94,7%, прецизионности 90%, полноты 100% и оценки F1 94,7%. Результаты этого исследования имеют важное значение для медицинских работников, поскольку раннее выявление рака молочной железы может значительно повысить шансы на выживание. В целом, это исследование вносит ценный вклад в область прогнозирования рака молочной железы с использованием машинного обучения.

**Ключевые слова:** итеративная настройка, модели на основе древовидных ансамблей, байесовская оптимизация, рак молочной железы, машинное обучение.

### Литература

1. Rabiei R., Ayyoubzadeh S.M., Sohrabei S., Esmaceli M., Atashi A. Prediction of breast cancer using machine learning approaches. *Journal of Biomedical Physics and Engineering*. 2022. vol. 12. no. 3. pp. 297–308. DOI: 10.31661/jbpe.v0i0.2109-1403.
2. Visvanathan K., Fabian C.J., Bantug E., Brewster A.M., Davidson N.E., DeCensi A., et al. Use of endocrine therapy for breast cancer risk reduction: ASCO clinical practice guideline update. *Journal of clinical oncology*. 2019. vol. 37. no. 33. pp. 3152–3165.
3. Thirthagiri E., Lee S.Y., Kang P., Lee D.S., Toh G.T., Selamat S., Yoon S.Y., Mohd Taib N.A., Nhong M.K., Yip C.H., Teo S.H. Evaluation of BRCA1 and BRCA2 mutations and risk-prediction models in a typical Asian country (Malaysia) with a

- relatively low incidence of breast cancer. *Breast Cancer Research*. 2008. vol. 10. no. 4. pp. 1–12. DOI: 10.1186/bcr2118.
4. Monirujjaman Khan M., Islam S., Sarkar S., Ayaz F.I., Kabir M.M., Tazin T., Albraikan A.A., Almalki F.A. Machine learning based comparative analysis for breast cancer prediction. *Journal of Healthcare Engineering*. 2022. vol. 2022. DOI: 10.1155/2022/4365855.
  5. Sharma D., Kumar R., Jain A. Breast Cancer Patient Classification from Risk Factor Analysis Using Machine Learning Classifiers. *Emergent Converging Technologies and Biomedical Systems: Select Proceedings of ETBS 2021*. 2022. pp. 491–504.
  6. Ali S., Majid A., Javed S.G., Sattar M. Can-CSC-GBE: Developing Cost-sensitive Classifier with Gentleboost Ensemble for breast cancer classification using protein amino acids and imbalanced data. *Computers in biology and medicine*. 2016. vol. 73. pp. 38–46.
  7. Gao L., Ding Y. Disease prediction via Bayesian hyperparameter optimization and ensemble learning. *BMC research notes*. 2020. vol. 13. pp. 1–6.
  8. Sharma S.K., Vijayakumar K., Kadam V.J., Williamson S. Breast cancer prediction from microRNA profiling using random subspace ensemble of LDA classifiers via Bayesian optimization. *Multimedia Tools and Applications*. 2022. vol. 81. no. 29. pp. 41785–41805.
  9. Rasool A., Bunterngchit C., Tiejian L., Islam M.R., Qu Q., Jiang Q. Improved machine learning-based predictive models for breast cancer diagnosis. *International journal of environmental research and public health*. 2022. vol. 19(6). no. 3211. DOI: 10.3390/ijerph19063211.
  10. Latchoumi T.P., Ezhilarasi T.P., Balamurugan K. Bio-inspired weighed quantum particle swarm optimization and smooth support vector machine ensembles for identification of abnormalities in medical data. *SN Applied Sciences*. 2019. vol. 1(10). no. 1137. DOI: 10.1007/s42452-019-1179-8.
  11. Lahoura V., Singh H., Aggarwal A., Sharma B., Mohammed M.A., Damasevicius R., Kadry F., Cengiz K. Cloud computing-based framework for breast cancer diagnosis using extreme learning machine. *Diagnostics*. 2021. vol. 11(2). no. 241. DOI: 10.3390/diagnostics11020241.
  12. Showrov M.I.H., Islam M.T., Hossain M.D., Ahmed M.S. Performance comparison of three classifiers for the classification of breast cancer dataset. in *4th International conference on electrical information and communication technology (EICT)*. 2019. pp. 1–5.
  13. Sheth P.D., Patil S.T., Dhore M.L. Evolutionary computing for clinical dataset classification using a novel feature selection algorithm. *Journal of King Saud University-Computer and Information Sciences*. 2022. vol. 34. pp. 5075–5082.
  14. Ara S., Das A., Dey A. Malignant and benign breast cancer classification using machine learning algorithms. in *2021 International Conference on Artificial Intelligence (ICAI)*. 2021. pp. 97–101.
  15. Lu H., Wang H., Yoon S.W. A dynamic gradient boosting machine using genetic optimizer for practical breast cancer prognosis. *Expert Systems with Applications*. 2019. vol. 116. pp. 340–350.
  16. Wang S., Wang Y., Wang D., Yin Y., Wang Y., Jin Y. An improved random forest-based rule extraction method for breast cancer diagnosis. *Applied Soft Computing*. 2020. vol. 86. no. 105941. DOI: 10.1016/j.asoc.2019.105941.
  17. Ozkan G.Y., Gunduz S.Y. Comparison of Classification Algorithms for Survival of Breast Cancer Patients. *Innovations in Intelligent Systems and Applications Conference (ASYU)*. 2020. pp. 1–4.
  18. Haque M.N., Tazin T., Khan M.M., Faisal S., Ibraheem S.M., Algethami H., Almalki F.A. Predicting characteristics associated with breast cancer survival using multiple

- machine learning approaches. *Computational and Mathematical Methods in Medicine*. 2022. vol. 2022. pp. 1–12. DOI: 10.1155/2022/1249692.
19. Mishra A.K., Roy P., Bandyopadhyay S. Genetic algorithm based selection of appropriate biomarkers for improved breast cancer prediction. *Intelligent Systems and Applications: Proceedings of the Intelligent Systems Conference (IntelliSys)*. 2020. vol. 2. pp. 724–732.
  20. Mishra A.K., Roy P., Bandyopadhyay S. Binary particle swarm optimization based feature selection (bpso-fs) for improving breast cancer prediction. *Proceedings of International Conference on Artificial Intelligence and Applications: ICAIA 2020*. 2021. pp. 373–384.
  21. Breiman L. Bagging predictors. *Machine learning*. 1996. vol. 24. pp. 123–140.
  22. Harrou F., Saïdi A., Sun Y. Wind power prediction using bootstrap aggregating trees approach to enabling sustainable wind power integration in a smart grid. *Energy Conversion and Management*. 2019. vol. 201. no. 112077. DOI: 10.1016/j.enconman.2019.112077.
  23. Buhlmann P., Yu B. Analyzing bagging. *The annals of Statistics*. 2002. vol. 30. no. 4. pp. 927–961.
  24. Liu C., Zhao Z., Gu X., Sun L., Chen G., Zhang H., Jiang Y., Zhang Y., Cui X., Liu C. Establishment and verification of a bagged-trees-based model for prediction of sentinel lymph node metastasis for early breast cancer patients. *Frontiers in Oncology*. 2019. vol. 9. no. 282. DOI: 10.3389/fonc.2019.00282.
  25. Deepa R., Gnanadesigan R., Ranjith D., Nithishkumar K., Dinesh A., Moorthy C. Performance Analysis of the Classification of Breast Cancer. *Third International Conference on Inventive Research in Computing Applications (ICIRCA)*. 2021. pp. 1–6.
  26. Dawngliani M., Chandrasekaran N., Lalmuanawma S., Thangkhanhau H. Prediction of breast cancer recurrence using ensemble machine learning classifiers. *Security with Intelligent Computing and Big-Data Services 2019: Proceedings of the 3rd International Conference on Security with Intelligent Computing and Big-data Services (SICBS)*. 2020. pp. 232–244.
  27. Assegie T.A., Tulasi R.L., Kumar N.K. Breast cancer prediction model with decision tree and adaptive boosting. *IAES International Journal of Artificial Intelligence*. 2021. vol. 10(1). no. 184.
  28. Okagbue H.L., Adam P.I., Oguntunde P.E., Obasi E.C., Odetunmbi O.A. Machine learning prediction of breast cancer survival using age, sex, length of stay, mode of diagnosis and location of cancer. *Health and Technology*. 2021. vol. 11. pp. 887–893.
  29. Friedman J., Hastie T., Tibshirani R. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics*. 2000. vol. 28. no. 2. pp. 337–407.
  30. Wu J., Gan M., Jiang R. A genetic algorithm for optimizing subnetwork markers for the study of breast cancer metastasis. *Seventh International Conference on Natural Computation*. 2011. pp. 1578–1582.
  31. Seiffert C., Khoshgoftaar T.M., Van Hulse J., Napolitano A. RUSBoost: A hybrid approach to alleviating class imbalance. *IEEE transactions on systems, man, and cybernetics-part A: systems and humans*. 2009. vol. 40. pp. 185–197.
  32. Tian J.-X., Zhang J. Breast cancer diagnosis using feature extraction and boosted C5.0 decision tree algorithm with penalty factor. *Math Biosci Eng*. 2022. vol. 19. no. 3. pp. 2193–2205.
  33. Jiang X., Xu C. Deep learning and machine learning with grid search to predict later occurrence of breast Cancer metastasis using clinical data. *Journal of clinical medicine*. 2022. vol. 11(19). no. 5772.

34. Deshwal V., Sharma M. Breast cancer detection using SVM classifier with grid search technique. *International Journal of Computer Applications*. 2019. vol. 975. no. 8887.
35. Buttan Y., Chaudhary A., Saxena K. An improved model for breast cancer classification using random forest with grid search method. *Proceedings of Second International Conference on Smart Energy and Communication: ICSEC 2020*. 2021. pp. 407–415.
36. Assegie T.A. An optimized K-Nearest Neighbor based breast cancer detection. *Journal of Robotics and Control (JRC)*. 2021. vol. 2. no. 3. pp. 115–118.
37. Mate Y., Somai N. Hybrid feature selection and Bayesian optimization with machine learning for breast cancer prediction. *7th International Conference on Advanced Computing and Communication Systems (ICACCS)*. 2021. pp. 612–619.
38. Ceylan Z. Diagnosis of breast cancer using improved machine learning algorithms based on bayesian optimization. *International Journal of Intelligent Systems and Applications in Engineering*. 2020. vol. 8. no. 3. pp. 121–130.
39. kaggle Repository. Breast Cancer Wisconsin Diagnostic Dataset. Available at: <https://www.kaggle.com/code/karan1210/breast-cancer/data>. (accessed 07.06.2022).
40. kaggle Repository. Breast Cancer Coimbra Dataset. Available at: <https://www.kaggle.com/datasets/yasserhessein/breast-cancer-coimbra-data-set>. (accessed 07.06.2022).
41. Surveillance, Epidemiology, and End Results (SEER) program. SEER breast cancer data. Available at: <https://ieee-dataport.org/open-access/seer-breast-cancer-data>. (accessed 07.06.2022).

**Алсабри Айман** — аспирант, кафедра информатики, Университет Саны. Область научных интересов: информатика, искусственный интеллект, машинное обучение. Число научных публикаций — 7. [aymanalsabri@su.edu.ye](mailto:aymanalsabri@su.edu.ye); Университетская улица Саны, Сана, Йемен; р.т.: +967(774)265-262.

**Альгабри Малек** — Ph.D., доцент, руководитель отдела, кафедра информатики, факультет вычислительной техники и информационных технологий, Университет Саны. Область научных интересов: информатика, MANETs, MPLS, искусственный интеллект, машинное обучение. Число научных публикаций — 17. [malekye@su.edu.ye](mailto:malekye@su.edu.ye); Университетская улица Саны, Сана, Йемен; р.т.: +967(777)662-909.