An Optimized Framework for Breast Cancer Prediction Using Classification and Regression Tree

Asma Agaal(✉) and Mansour Essgaer

Department of Artificial Intelligence, Faculty of Information Technology, Sebha University, Sebha, Libya
{asma.agaal,man.essgaer}@sebhau.edu.ly

Abstract. Several machine learning algorithms have been proposed in recent years to design accurate classification systems for a wide range of diseases such as cancers, hepatitis, and coronavirus. In this study, the Classification and Regression Tree (CART) is proposed to predict breast cancer in the early stage, later applied to real data collected from the Sebha oncology center. The study focuses on improving the CART accuracy through several methods: (1) cross-validation, (2) dimensionality reduction and (3) hyper-parameter tuning. However, two cross-validation strategies have been investigated namely: The K fold and stratified fold, followed by dimensionality reduction to determine the most effective features using two methods, namely: recursive feature elimination with cross-validation and principal component analysis, and lastly, investigating the most optimal CART parameters using two optimization algorithms, namely: grid search, and random search. The experimental results have shown that the best CART model which achieved 97% accuracy uses a stratified fold as a cross-validation strategy, recursive feature elimination with cross-validation as dimensionality reduction, and grid search as parameters tuning algorithm. Moreover, when compared to the original CART, the accuracy of the proposed CART has improved from 63% to 97%.

Keywords: Breast Cancer · CART · Sebha Oncology Center · Dimensionality reduction with cross-validation · Grid search · Hyper-parameter tuning

1 Introduction

These days, research into the use of Bio-Marker (BM) based approaches to identify Breast Cancer (BC) is expanding quickly. A biomarker is a quantifiable element, organ, or bodily function that can be used to identify the existence or frequency of a disease [1]. Blood pressure, temperature, red blood cells, blood platelets, and genomic information are only a few examples of biomarkers [2]. Breast cancer BM features were addressed in this study, which was collected from Sebha Oncology Center (SOC). These fundamental BM, which can be found in blood samples, will be used in this study to assess BC susceptibility in the southern region of Libya. This is because it is less time- and money-consuming than alternative examination techniques.

I G. P. Suta Wijaya et al. (Eds.): MIMSE-I-C 2022, ACSR 102, pp. 398–412, 2022.
https://doi.org/10.2991/978-94-6463-084-8_33

Although medical data mining typically extracts knowledge from big datasets, it can be challenging to do so from smaller amounts of data, such as the few patient records [3]. Where the data miner faces two basic challenges based on the number of features in the datasets and the classification procedures. The first issue is obtaining the most accurate model based on the available dataset, and the second is determining the effectiveness of the proposed model on samples that have not yet been seen. Moreover, model improvement seeks to identify one classifier's optimal level of complexity [4].

However, there is a challenge in getting higher accuracy with a limited amount of data, which can be overcome by utilizing Cross-Validation techniques (CV). Additionally, some models suffer from limitations such as the inability to generalization, high processing costs, and slow performance with large data, which can be solved by applying Dimensionality Reduction (DR) [3]. On the other hand, one of the criteria to assess any model's efficiency is how well it fits the data. However, as models become more complicated, with more parameters, they will inevitably produce models either under or overfitting the data. Overfitting occurs when a model accurately describes previously trained data but fails to generalize to new data. So the aim is to design machine learning models that have fewer generalization errors by balancing model complexity with the goodness of fit by adjusting hyper-parameters [5].

This study aims to obtain an accurate breast cancer prediction model based on CART, which is shown to be effective in medical fields [6, 7]. The CART model will be enhanced in this work by employing DR techniques to select the best optimal subset of the proposed model based on recursive feature elimination with CV and principal component analysis. In the same, a CV strategy will also be employed to better generalize the model based on the K fold and stratified fold. Furthermore, the CART algorithm's efficiency will be improved by fine-tuning hyper-parameters to achieve the optimum fit between training and test data based on grid search, and random search.

The rest of this paper is organized as follows: The related work is presented in Sect. 2, the materials and techniques are presented in Sect. 3, the experiments and discussion are presented in Sect. 4, and the conclusion is presented in Sect. 5.

2 Literature Review

In medical diagnostics, a common option when dealing with a large number of input features is to use specialized strategies to reduce the dimensions of the original problem [8]. Some authors [9] focused on the most significant and best BM for diagnosing BC by utilizing several feature selection methods. The results show that Glucose, Age, and Resistin are the most important feature. The K-NN classifier has the maximum classification accuracy when these three features are utilized. In contrast, according to a different study by [10] the top 5 diagnostic BMs are glucose, age, body mass index (BMI), Resistin, and MCP-1. In comparison to other machine-learning models, the SVM classifier had the highest classification accuracy.

Furthermore, there is a consensus among scientists that the Principal Component Analysis (PCA) is significant as a DR method, and it can improve the performance of many machine learning models [11]. Moreover, the most significant benefit of PCA as compared to other DR is the ability to capture the data's variability [12]. Furthermore,

some scholars [12] have highlighted the model generalization and fitting issues, and they point out that CV is a method for measuring generalization performance in this context. In addition [13] claims that CV can thus be used in three different contexts: performance estimation, model selection, and tuning model parameters. On the other hand, many scholars argue about the estimated value of K in K folds CV. Generally, [13, 14] indicates that the greater the value of k, the better the model's accuracy. However, [15] confirm that this can lead to model over-fitting and suggest that $k = 2$ is better when choosing between two models. This fact is supported by [16] who writes that it is commonly believed that the ideal k is between 5 and 10. Furthermore, other scholars [17–19] conclude that CV is better performed with $k = 10$ or $k = 20$ since these numbers have been empirically proved to give low test error.

Based on other prospective, [20] claims that maximum depth is the most important parameter to avoid overfitting. Furthermore [21] points out that the one of decision tree-based model's drawbacks relies on the default parameters settings, such as splitting criteria and min-weight, which are known as the most effective parameters on tree-based model performance. On the other hand, [22] argues that the total number of leaves, the depth of the tree, the total number of nodes, the depth of the tree, and the number of features employed are used as metrics to evaluate tree complexity.

Early studies, identify these parameters based on prior experiences or trial and error or based on the literature recommendations. Finding an appropriate set of values manually can be subjective and time-consuming depending on the amount of training time available [23]. To solve this issue, optimization techniques are frequently used to find an appropriate set of hyper-parameter settings. Furthermore, some scholars [24] have found that random search provides better outcomes than grid search. Whereas [25] stated that random search works similarly to grid search in tuning models with different combinations of parameter values. Therefore, in this study, the efficiency of DR, CV, and optimization of hyper-parameters that help in increasing CART performance will be examined.

3 Material and Method

The suggested framework for enhancing the CART algorithm's performance to more accurately predict BC is shown in Fig. 1.

3.1 Data Acquisition

- Data Collection: Data were collected from SOC in southern Libya. The data include 2435 cases gathered between 2015 and 2020 from various southern cities.
- Dataset Description: The dataset's features are presented in Table 1 and include the target, which may be benign or malignant, as well as 22 other features.

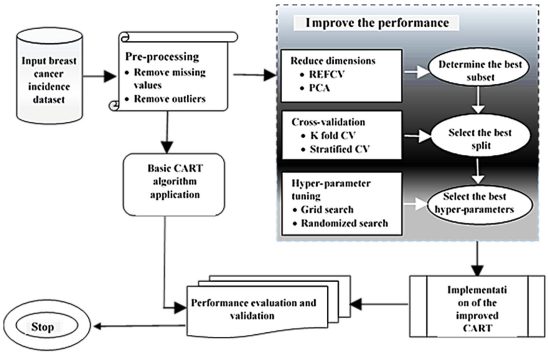

Fig. 1. Describes the proposed methodology

Table 1. Dataset characteristics

Feature	Feature description	Feature	Feature description
Sex	The patient's gender	ESR	Deposition of blood
Age	The patient's age	LDH	Lactate Dehydrogenizes
Address	The patient's address	Na	Sodium
FBS	Blood glucose	K	Potassium
Urea	Urea in blood	CL	Chloride acid
Creatinine	Creatinine in blood	CA15	Cancer antigen
ALB	Albumin	CEA	Carcinoma embryonic antigen
T-Ca	Total calcium in the blood	WBC	White blood cells
GPT	Liver functions ' gpt'	RBC	Red blood cells
GOT	Liver functions ' got '	**Class**	**Benign or Malignant**
ALP	Alkaline Phosphate		
HGB	Hemoglobin		
PLT	Blood platelets		

3.2 Data Preprocessing

The data were preprocessed as follows after being collected in raw form:

- Missing values handling: Average is used to replace all continuous numerical features that have missing values [26].
- Remove anomalies: Outliers were removed by applying the Z-Score technique [27].

3.3 Classification and Regression Tree (CART)

The CART is described as a top-down, recursive flowchart tree structure, similar to a binary decision, where each internal node stands in for a test on features, each branch

for a test result, and each leaf node for the class label. The best tree splitting in CART can also be found in various ways [26], where CART offers Gini Index, also known as Gini Impurity, is a measure of impurity that is used in place of entropy and is calculated as given in Eq. 1 as follows:

$$Gini(D) = 1 - \sum_{i=1}^{m} p^2 \tag{1}$$

The sum is computed over m classes, where Pi is the likelihood that a tuple in D connects to class Ci.

Selecting input variables and dividing points on those variables until an appropriate tree is created is the first step in creating a CART model. A greedy algorithm is used to optimize a cost function when deciding which input variable to utilize and which split or cut-point to employ. A preset stopping condition, such as a minimum number of training cases assigned to each leaf node of the tree, is used to cease tree construction [6].

3.4 Dimensional Reduction Techniques (DR)

A common option when dealing with a large number of input features is to utilize specialized strategies to minimize the dimensionality of the original problem which is known as dimension reduction techniques (DR). On the other hand, DR strategies, are commonly separated into Feature Selection (FS) and Feature Extraction (FE) methods [8]. The primary distinction is that FE combines the original characteristics and generates a set of new features, whereas FS selects a subset of the original features [28]. Some of these methods will be addressed further below:

- Recursive Feature Elimination with Cross Validation (RFECV): It's one of the most widely used methods for selecting features, and in this method, all the features are sorted in order of relevance using the model that uses all the features found in the first phase. A new determination of feature fit is made after the model has been completely reconstructed. Every iteration involves calculating the regenerator model's best performance as well as the fit of the high-performing features to the final model [29].
- Principal component analysis (PCA): It's a basic nonparametric method for extracting the most relevant features, by removing components that do not contribute significantly to the observed variability, PCA minimizes the number of original variables because it is a linear data transformation that reduces duplication (measured by covariance) while increasing information (measured by variance) [11].

3.5 Cross-Validation Techniques (CV)

A statistical method known as CV is used to assess and evaluate learning algorithms that separate data into two sections: one is used to train a model, and the other is used to validate it. In order for each data point to have a chance of being validated, the training and validation sets must intersect in successive rounds [30]. Some of these methods will be discussed in detail as follows:

- K-Fold CV: It can be used to improve the hold-out approach. This strategy ensures that the model's score is independent of how chose the training and testing sets, according to the following steps: (1) divide the entire dataset into k folds at random (subsets), and build a model on (k–1) folds, then the model be tested to on remaining (k) fold. (2) Repeat the first step until all k-folds have served as the test set and the model's performance metric is the average of recorded accuracy for each K fold [31].
- Stratified Fold CV: It can be difficult to use K-Fold on a classification task. Because randomly shuffling the data and then dividing it into folds, may end up with severely imbalanced folds, causing training to be biased. Therefore, uses stratification to create stratified folds so that each fold is a decent representation of the entire dataset while keeping the percentage of sample for each class. This eliminates the random sample issue that the K-Fold approach has [32].

3.6 Hyper-Parameter Tuning Techniques

It's the task of determining ideal hyper-parameters for a machine learning algorithm for a given dataset [33]. A Learning algorithm has two sorts of variables: hyper-parameters and parameters. Model parameters are learned throughout the training process for making predictions, whereas model hyper-parameters regulate the behavior of the model during the training phase and are specified before model training even begins [34]. There are several tuning strategies, some of which will be addressed below:

- Grid Search: Creates models of the values provided and chooses the model with the best performance. Although this method is utilized for automatic tuning, the algorithm's performance rapidly drops as the range of hyper-parameters being tuned grows, resulting in high processing costs [35].
- Random Search: It works similarly to Grid search, analyzing models with different combinations of values, but instead of performing an exhaustive search, it requires a random selection for each hyper-parameter. When compared to Grid Search, this method is more efficient in a high-dimensional space since not all hyper-parameters are equally relevant to tune [24].

4 Results and Discussion

This section focuses on the results of the proposed framework, where the SOC dataset was used for BC prediction in two stages: the first stage involved applying the fundamental CART algorithm, and the second stage involved several CART algorithm improvements based on various techniques, such as DR techniques, validation techniques, and finally hyper-parameter tuning techniques, to achieve the best possible CART performance. The experiments were carried out in the Python environment because it is a well-defined foundation for machine learning. The experiments can be summarized as follows:

- Applying the basic CART.
- Optimizing the CART algorithm, through several methods including (1) DR techniques, (2) CV methods, and (3) hyper-parameter tuning techniques.
- Comparison of the basic and improved CART performance to determine the percentage of enhancement.

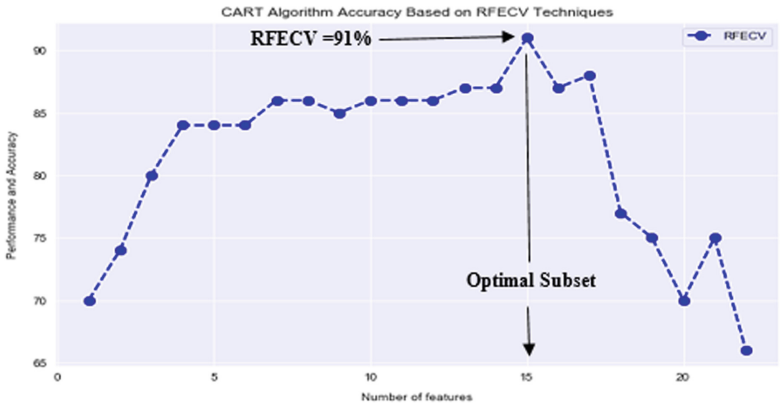

Fig. 2. The accuracy of each subset of features based on RFECV

4.1 Performance of Basic CART

In the beginning, the basic CART is utilized with all of the original parameters. The result was 63% accuracy, 0.63% precision, 0.63% recall, and 0.63% f1-score.

4.2 CART Performance Enhancement

Based on a set of criteria, a model is judged to be the "best" one. Two common strategies are generalization and goodness of fit. A variety of tests, which we summarize in more detail below, can explain why this only happens if the model is improved:

Applying the Dimensionality reduction techniques. To determine the best technique for DR, a comparison between the RFECV and the PCA technique will be conducted, the result is summarized below:

Regarding the performance of CART when utilizing a different set of features. The RFECV method produced the best results, with an accuracy of nearly 91%, using the 15 features listed in Fig. 2, namely: “Age,” “Address,” “FBS,” “UREA,” “ALB,” “T-Ca,” “GPT,” “ALP,” “CA15,” “CEA,” “WBC,” “RBC,” “PLT,” “ESR,” and “LDH”.

On the other hand, the next thing to consider while implementing PCA and optimization of CART performance is "How many core components will we use for the new subspace feature?" In methods of extracting features. A useful metric is what is called "explained variance," which indicates how much information (variance) can be assigned to each of the major components. Thus, the first principal component alone has explained the majority of the variance—0.49 of the variance, to be exact—as seen in Fig. 3. Information is still present in the second principal component 0.17 and is also present in the third and fourth principal components combined at 22%. Comparatively speaking, the fifth, sixth, seventh, and eighth components share less information than the other primary components, but since they account for about 10% of the data, this information cannot be disregarded. Furthermore, the first nine components comprise 89.99 of the information. In contrast, as they account for less than 1% of the variation,

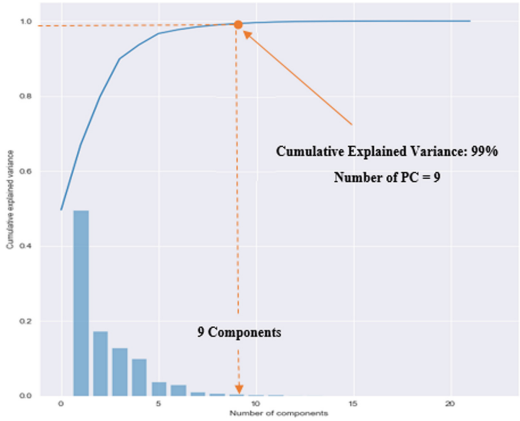

Fig. 3. Describes the cumulative explained variance of features using PCA

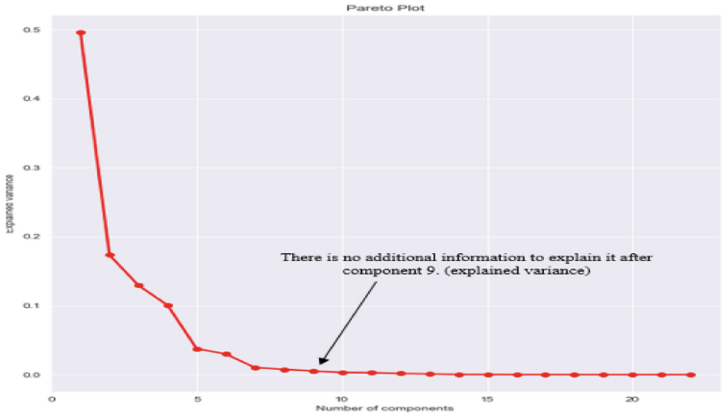

Fig. 4. Describes the explained variance of components using PCA

we may safely eliminate the remaining components without significantly affecting the data.

On the other hand, Fig. 4 shows that, no matter how many components we add, the variance reduces beyond component 9 and that it is, therefore, to avoid overfitting, the CART model only is trained on 9 components and achieved an accuracy of 83%.

Thus, based on the given results, we can conclude that the RFECV approach outperformed the PCA method with an accuracy of 91%. As a result, when trained on only 15 features, the DR technique increased the accuracy of the CART model from 63% to 91%.

Applying Cross-validation techniques. To employ CV approaches, the model must first be trained using N number of parameters. In this experiment, the range was determined with values ranging from 1 to 30 for the value of k-fold. This will help to find out the ideal value of k folds. The following experiments demonstrate this:

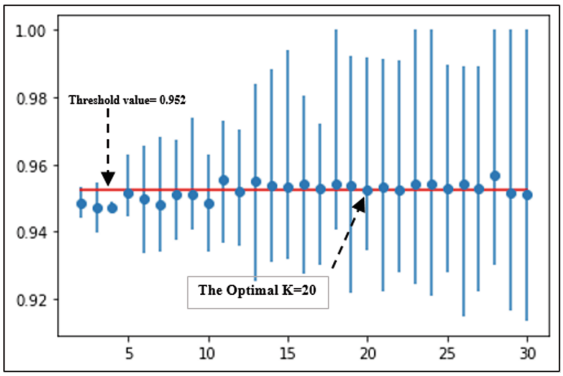

Fig. 5. Determine the optimal value of k

Table 2. THE K-Fold CV AND Stratified CV PERFORMANCE

	K-Fold CV			Stratified CV		
	Precision	recall	f1-score	Precision	recall	f1-score
Benign	0.96	0.95	0.96	0.96	0.97	0.96
Malignant	0.94	0.94	0.95	0.96	0.96	0.95
Weighted avg	0.95	0.95	0.95	0.96	0.96	0.96
Accuracy	0.95			0.96		

- Determining the optimal K:
  To calculate the ideal value of K, the threshold at which the model CART performs with the highest level of accuracy should be identified. So, the threshold value was set at 0.952 by CART, and the best value for K at this time was 20 as shown in Fig. 5.
- Comparison of the two cross-validation techniques
  Cross-validation methods like K-Fold CV and Stratified CV are utilized in this study to divide the data set into 20 folds while training the optimized CART model on 15 candidate features using the REFCV method. The final accuracy of the CART model is based on the average accuracy of 20 fields. Table 2 shows the result for the two CV where the Stratified approach outperformed the K-Fold by achieving an accuracy of 96%. Moreover, the model's accuracy has increased dramatically from 91% to 96%.
  Applying hyper-parameter tuning techniques. In this section CART parameters, such as max depth, max-leaf nodes, min-samples-split, and random state, will be tweaked as outlined below to improve the CART accuracy:
- Tunning the max depth of the tree
  As shown below, three methods based on mean squared error, model fit, and the ideal value of k fold were used to tunned the tree depth, where the model was trained using parameters ranging from 1 to 20 for max-depth:

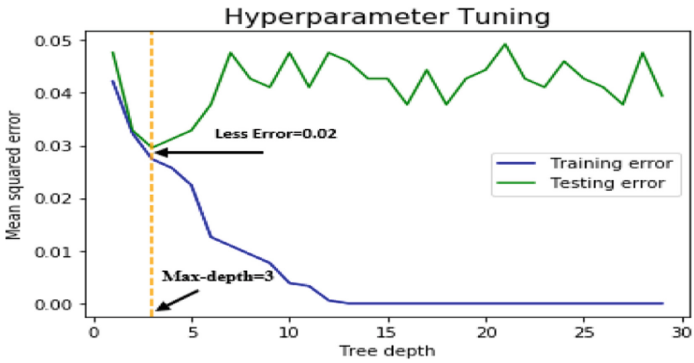

Fig. 6. Max-depth parameter based on mean square error

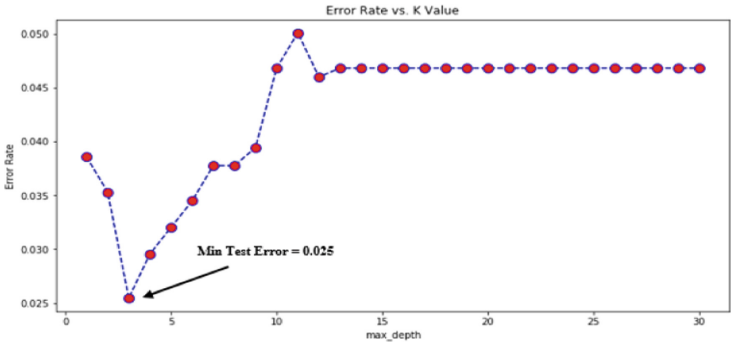

Fig. 7. Max-depth parameter based on the test set's maximum accuracy

- Based on mean squared error:
  Figure 6 shows the training and testing's mean squared error. When both the test error and training error values are too low, it is where the optimal tree depth parameter value is. As can be seen in Fig. 6, max-depth = 3 is the ideal value.
- Based on the least squared mean error of the test:
  On the other hand, by locating the test set's lowest error, it is feasible to establish the best depth for a tree. Figure 7 shows the model's least squared error 0.025 at tree depth 3.
- Based on the optimal value of k:
  The ideal value of the tree max-depth was looked at using a k-fold CV method. The experiments are set based on max-depth values ranging from1 to 30. Figure 8, shows the best CART accuracy of 97% at tree depth = 3, with the value of k = 20 as obtained from the previous experiment. Additionally, another interesting observation is that the model's accuracy starts to decline as the tree's depth rises.
- Based on the fitting of the training and testing:
  By examining the model's over-fitness, we can observe that it has the best tree depth at 3, where it fits the training and test data with a 97%. Once the training model has reached this depth, it starts to over-fit, as seen in Fig. 9.

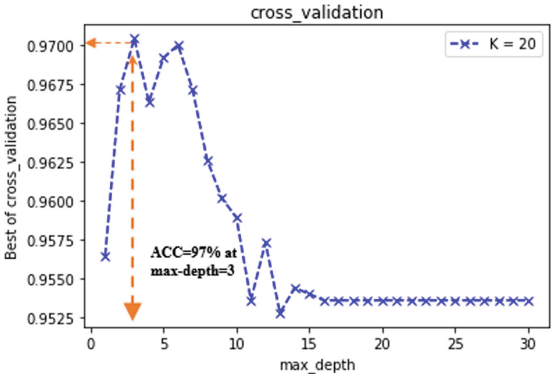

Fig. 8. Max-depth parameter based on the optimal value of k-fold

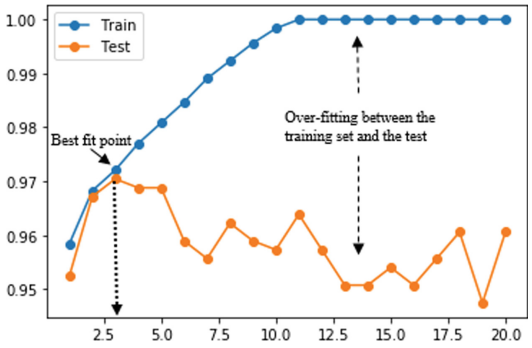

Fig. 9. Max-depth parameter based on Over-fitting

In a conclusion, the best value of the max depth of the tree was 3 since it increased the model's accuracy by up to 97%.

- hyper-parameters tuning techniques:
  Furthermore, to further verify and concentrate on fine-tuning some CART parameters, like maximum depth, minimal sample segmentation, accurate leaf samples, and maximum leaf nodes, and show how these can enhance CART model prediction. Grid search and random search methods were used to achieve this. These methods are used to tune values for the aforementioned parameters, as shown in Table 3:

  Therefore, when using hyper-parameters tuning techniques, the grid search outperforms the random search with an accuracy of 97.3% against 96.6%, as shown by the AUC curve in Fig. 10.

  However, regarding optimizing recall metrics. Using the grid search technique increased the false negative prediction rate from 98% to 99%, compared to the rate reached by the CART algorithm when only the tree depth parameter was tuned. This means that CART's performance has an error of 0.01 in determining actual positive

Table 3. Comparison of hyper-parameter tuning techniques for the CART algorithm

Hyper-parameter	The ideal value to tune	
	Grid Search	Random Search
max depth	3	7
max-leaf nodes	12	22
min samples leaf	2	12
min samples split	22	12

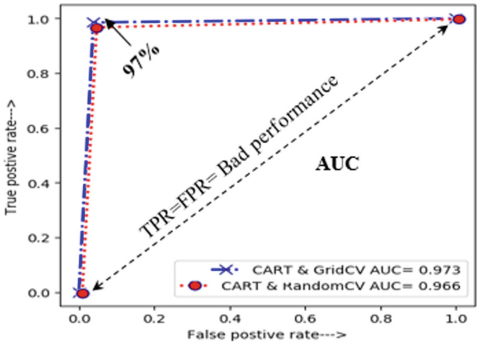

Fig. 10. The AUC curve shows the accuracy of the CART using tuning methods

Table 4. The original and improved cart performance

	Original CART			Improvement CART		
	Precision	recall	f1-score	Precision	recall	f1-score
Benign	0.68	0.65	0.66	0.99	0.96	0.97
Malignant	0.58	0.61	0.58	0.95	0.99	0.97
Weighted avg	0.63	0.63	0.63	0.97	0.97	0.97
Accuracy	0.63			0.97		

predictions, which boosts the model's efficiency in predicting malignant cases by using the grid search method.

4.3 Performance of the Original and Improved CART

The aforementioned experiments were conducted to enhance the performance of the original CART. The result is shown in Table 4, the weighted average and accuracy of the CART have both improved. Accuracy has enhanced by 34%, from 63% to 97%.

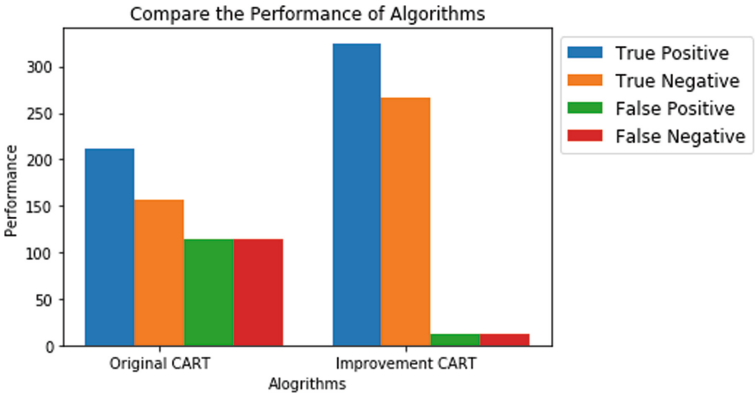

Fig. 11. The confusion matrix of the improved CART against original

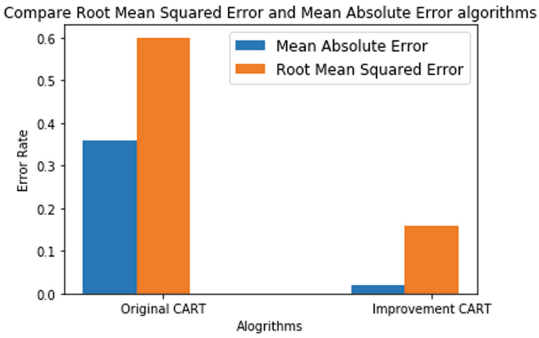

Fig. 12. The error rate of the improved CART vs. the original

Furthermore, Fig. 11 demonstrates that the highest percentage of real positive predictions was achieved using the improved CART. According to Fig. 12, the improved CART has the lowest MAE and RMSE when compared to the original CART.

5 Conclusion

Dimensionality reduction, cross-validation, and hyper-parameter tuning techniques are proposed in this study to enhance the performance of the CART. The study demonstrates how important these techniques are in improving the classifier's accuracy. Firstly, an accuracy of 91% is obtained using the recursive feature elimination with cross-validation as a dimensionality reduction technique, compared to an accuracy of 63% of the original CART. Additionally, the Stratified cross-validation method significantly improved the model's accuracy, bringing it to 96% after identifying the ideal value of k equal to 20. However, the results demonstrated that altering the number of folds does not necessarily produce accurate results. The increase can sometimes improve accuracy while other times it will merely increase the processing expense. When the hyperparameters were adjusted, the grid search technique surpassed the random search in increasing

CART accuracy by 97%. However, when only the tree depth parameter was adjusted, the CART method achieved the same accuracy, confirming the efficiency of this parameter in improving model accuracy when compared to other parameters.

Future work will include the addition of pre-and post-processing hyper-parameters that could make the tuning problem being addressed more complex, as well as expanding the experiments to include more different ML.

References

1. [1]Trovato, M.J.D.: Update on International Medical Taxonomies of Biomarkers and Their Applications in Management of Thyroid Cancers. 12(3), 662 (2022).
2. [2]Tabl, A.A., et al.: A machine learning approach for identifying gene biomarkers guiding the treatment of breast cancer. 10, 256 (2019).
3. Nematzadeh, Z., R. Ibrahim, and A. Selamat.: Comparative studies on breast cancer classifications with k-fold cross validations using machine learning techniques. In: 2015 10th Asian control conference (ASCC). IEEE (2015).
4. Kaklamanis, M.M. and M.E. Filippakis.: A comparative survey of machine learning classification algorithms for breast cancer detection. In: Proceedings of the 23rd Pan-Hellenic Conference on Informatics (2019).
5. Ying, X.: An overview of overfitting and its solutions. In: Journal of physics: Conference series. IOP Publishing (2019).
6. [6]Ghiasi, M.M., et al.: Decision tree-based diagnosis of coronary artery disease: CART model. 192, 105400 (2020).
7. Agaal, A. and M. Essgaer.: Biological and Tumor Markers in Early Prediction Phase of Breast Cancer UsingClassification and Regression Tree: Sebha Oncology Center as a Case study. In: 2022 IEEE 2nd International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering (MI-STA). (2022).
8. Remeseiro, B., V.J.C.i.b. Bolon-Canedo, and medicine.: A review of feature selection methods in medical applications. 112, 103375 (2019).
9. Singh, B.K.J.B. and B. Engineering.: Determining relevant biomarkers for prediction of breast cancer using anthropometric and clinical features: A comparative investigation in machine learning paradigm. 39(2), 393–409 (2019).
10. [10]Pham, H., et al.: A novel generalized logistic dependent model to predict the presence of breast cancer based on biomarkers. 32(1), e5467 (2020).
11. Chowdhary, C.L. and D.J.P.C.S. Acharjya.: Segmentation and feature extraction in medical imaging: a systematic review. 167, 26–36 (2020).
12. Jain, D., V.J.I.J.o.C. Singh, and Applications.: A two-phase hybrid approach using feature selection and adaptive SVM for chronic disease classification. 43(6), 524–536 (2021).
13. Marcot, B.G. and A.M.J.C.S. Hanea.: What is an optimal value of k in k-fold cross-validation in discrete Bayesian network analysis?. 36(3), 2009–2031 (2021).
14. Raschka, S.J.a.p.a.: Model evaluation, model selection, and algorithm selection in machine learning (2018).
15. Arlot, S. and A.J.S.s. Celisse.: A survey of cross-validation procedures for model selection. 4, 40–79 (2010).
16. [16]Jiang, G. and W.J.P.R. Wang.: Error estimation based on variance analysis of k-fold cross-validation. 69, 94-106 (2017).
17. James, G., et al.: Resampling methods. In: An introduction to statistical learning, pp. 175–201. Springer (2013).

18. Zhang, Y. and Y.J.J.o.E. Yang.: Cross-validation for selecting a model selection procedure. 187(1), 95–112 (2015).
19. Hamza, R. and M.J.I.I.P. Chtourou.: Design of fuzzy inference system for apple ripeness estimation using gradient method. 14(3), 561–569 (2020).
20. Alawad, W., M. Zohdy, and D. Debnath.: Tuning hyperparameters of decision tree classifiers using computationally efficient schemes. In: 2018 IEEE First International Conference on Artificial Intelligence and Knowledge Engineering (AIKE). IEEE (2018).
21. Mantovani, R.G., et al.: An empirical study on hyperparameter tuning of decision trees. (2018).
22. Rokach, L. and O. Maimon.: Decision trees. In: Data mining and knowledge discovery handbook, pp. 165–192. Springer (2005).
23. Mantovani, R.G., et al.: Hyper-parameter tuning of a decision tree induction algorithm. In: 2016 5th Brazilian Conference on Intelligent Systems (BRACIS). IEEE (2016).
24. Bergstra, J. and Y.J.J.o.m.l.r. Bengio.: Random search for hyper-parameter optimization. 13(2) (2012).
25. Bergstra, J., et al.: Algorithms for hyper-parameter optimization. 24 (2011).
26. Maniruzzaman, M., et al.: Accurate diabetes risk stratification using machine learning: role of missing value and outliers. 42(5), 1–17 (2018).
27. García, S., J. Luengo, and F. Herrera.: Data preprocessing in data mining. Springer (2015).
28. Zhao, Z.A. and H. Liu.: Spectral feature selection for data mining. Taylor & Francis (2012).
29. Misra, P. and A.S.J.I.J.E.T. Yadav.: Improving the classification accuracy using recursive feature elimination with cross-validation. 11(3), 659–665 (2020).
30. Refaeilzadeh, P., L. Tang, and H.J.E.o.d.s. Liu.: Cross-validation. 5, 532–538 (2009).
31. Ghojogh, B. and M.J.a.p.a. Crowley.: The theory behind overfitting, cross validation, regularization, bagging, and boosting: tutorial. (2019).
32. Bey, R., et al.: Fold-stratified cross-validation for unbiased and privacy-preserving federated learning. 27(8), 1244–1251 (2020).
33. [33]Probst, P., et al.: Hyperparameters and tuning strategies for random forest. 9(3), e1301 (2019).
34. Antal-Vaida, C.J.I.E.: Basic Hyperparameters Tuning Methods for Classification Algorithms. 25(2) (2021).
35. [35]Radzi, S.F.M., et al.: Hyperparameter Tuning and Pipeline Optimization via Grid Search Method and Tree-Based AutoML in Breast Cancer Prediction. 11(10), 978 (2021).