Breast Cancer Detection using Decision Tree and Random Forest

Fergie Joanda Kaunang [1]*, Bhustomy Hakim [2], Fedelis Fraderic [3]*, Sherren Hartono [4]*, Andrew Kristanto Mulyanto [5]***
* Informatics, Universitas Bunda Mulia
** Information System, Universitas Bunda Mulia
fkaunang@bundamulia.ac.id [1], bhakim@bundamulia.ac.id [2], s32220096@student.ubm.ac.id [3], s32220107@student.ubm.ac.id [4], s32220118@student.ubm.ac.id [5]

Article Info

Article history:

Received 2025-01-10
Revised 2025-01-21
Accepted 2025-01-30

Keywords:

Breast Cancer,
Decision Tree,
Random Forest,
Machine Learning,
Bootstrap Aggregating.

ABSTRACT

Cancer is one of the most challenging diseases to cure and is a chronic condition that contributes significantly to global mortality. With advancements in artificial intelligence (AI) technology, AI-integrated systems can provide quick and accurate diagnoses based on collected medical data. By leveraging machine learning techniques, this study aims to compare the performance of two models using the Decision Tree (DT) and Random Forest (RF) algorithms on routine blood test data. The research process involves data preprocessing techniques such as handling missing values, detecting outliers, and feature selection, followed by applying the bootstrap aggregating technique to enhance model performance. Feature selection is used to identify the most significant features in the data that contribute to cancer detection. Using the KBest feature selection technique, the study found that the features age, BMI, leptin, adiponectin, and MCP-1 had the highest correlation with the target variable. The resulting models were evaluated to compare the performance of each algorithm. The evaluation results showed that the RF algorithm outperformed DT, achieving an accuracy of 89.65% on the processed dataset using the bootstrap technique, compared to DT's accuracy of 80.17%. Additionally, the RF algorithm demonstrated superior metric values, including a precision of 91.66% and an F1-score of 87.12%. This study concludes that the RF algorithm is more effective than DT for detecting cancer in limited datasets, especially when used with the bootstrap technique. The findings are expected to support the development of decision support systems in healthcare services for more accurate early cancer detection.

I. Introduction

Cancer is one of the leading causes of death worldwide. Based on data from the Global Cancer Observatory, in 2024 there were 19.9 million new cancer cases and 9.7 million cancer-related deaths worldwide [1]. This disease is a global health threat because of its chronic nature, progressive development, and high mortality rate if not detected and treated early. Early detection of cancer plays an important role in increasing the chances of healing, especially through more appropriate and targeted treatment [2]. One type of cancer that is the main focus of this study is breast cancer. Breast cancer is the leading cause of cancer death in women worldwide [3]. Breast cancer is traditionally diagnosed through mammography, biopsy, or specific tumor markers like CA 15-3. However, this study explores routine blood tests as a potential alternative or complementary screening tool. Previous research has identified metabolic dysregulation and biomarkers such as resistin and leptin as associated with breast cancer, particularly in obesity-related cases [4]. Routine blood data is considered relevant due to its ability to reflect metabolic and inflammatory states, which are often disrupted in cancer [5]. The dataset in this study includes both general parameters (e.g. age, BMI) and specific markers (e.g. resistin, glucose). Resistin, a pro-inflammatory cytokine, has previously been linked to cancer in postmenopausal women.

Glucose levels and BMI provide insight into metabolic health, which is often impaired in cancer patients [6].

Artificial Intelligence (AI) is a transformative field in computer science, focused on developing systems capable of performing tasks that normally require human intelligence. These tasks include reasoning, learning, problem solving, perception, and language comprehension [7]. Machine Learning (ML), a subset of AI, focuses on creating algorithms that allow computers to learn from data and make predictions or decisions. ML models are trained using large datasets, so they are able to recognize certain patterns and can continuously improve their abilities over time. This change in approach has driven major advances in a variety of application areas, such as natural language processing [8][9], prediction [10], computer vision and image processing [11], and robotics.

In recent years, the development of artificial intelligence (AI) technology has opened up new opportunities in medical diagnosis, including early detection of cancer [12][13][14]. By using machine learning algorithms, AI-based systems are able to analyze medical data quickly and accurately, thus assisting doctors in making clinical decisions. This approach has been shown to provide promising results, especially in processing large and complex data such as medical images and biomarkers [15][16].

Algorithms such as Decision Tree (DT) and Random Forest (RF) have been widely used for early detection of breast cancer. DT offers easy interpretation due to its simple structure, while RF, which is a DT-based ensemble method, is able to improve prediction accuracy by reducing model variance through techniques such as bootstrap aggregating (bagging) [17]. A study conducted by Shiny et al. (2024) showed that the RF algorithm can overcome the challenges of high-dimensional data and the risk of overfitting, which are often found in medical data analysis, thus providing more reliable results in breast cancer diagnosis [3]. In addition, these algorithms have also been used to evaluate digital mammography images and associated biomarkers, with results showing that RF is able to distinguish suspicious lesions with high accuracy, reaching 91.66% in a particular study [3]. This approach not only improves diagnostic efficiency but also helps support more informed decision-making by physicians, especially in detecting cancer at an early stage.

Algorithms such as Gradient Boosting, SVM, or deep learning have the potential to provide higher accuracy, especially on large and complex datasets. However, this study uses Decision Tree (DT) and Random Forest (RF) for several reasons. First, DT and RF are more suitable for small datasets such as the one used in this study (116 samples), where simpler models can perform well without requiring large amounts of data. Second, DT offers a simple and easily interpretable structure, which is important for medical diagnosis, while RF maintains process efficiency with increased accuracy through ensemble learning. Finally, deep learning requires much larger computational resources than DT and RF, making DT and RF a more practical choice for small datasets and limited computing devices.

Research in cancer detection often faces the challenge of limited data. Small datasets can cause machine learning models to be susceptible to overfitting, where the model cannot generalize well to new data. To overcome this, techniques such as bootstrap aggregating are used to improve model reliability. This technique involves creating multiple sample datasets by resampling the original dataset, which are then used to train multiple models. This process helps reduce model variability and improve prediction accuracy on limited datasets [18]. This study aims to evaluate the performance of DT and RF algorithms in detecting cancer by utilizing medical biomarker datasets. The evaluation was carried out using various metrics, including accuracy, precision, recall, and F1-score. In addition, this study also examines how bootstrap aggregating techniques can improve model performance in limited dataset conditions. It is hoped that the results of this study can provide significant contributions to the development of a more accurate and efficient AI-based early cancer detection system, especially in the context of breast cancer.

II. METHODS

This study has four main stages, namely data preprocessing to handle missing data (missing values), handling outliers/unusual values, feature selection, and bootstrap aggregating. The processed data is then given to the Decision Tree and Random Forest models to be trained and tested to obtain predictions. The predictions are then categorized with a confusion matrix containing True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). The predictions made by the model will then be evaluated with 4 indicators, namely: accuracy, precision, recall, and F1-score. The stages of this study can be seen in Figure 1. The dataset used in this study is collected from UC Irvine Machine Learning Repository. This study investigated 116 individuals: 64 diagnosed with breast cancer and 52 healthy participants. Ten quantitative attributes, including anthropometric measurements and routine blood test results, were collected for each individual. The primary outcome was the presence or absence of breast cancer, represented as a binary variable (0 or 1). The attributes are Age, BMI, Glucose, Insulin, HOMA, Leptin, Adinopectin, Resistin, MCP.1, and the classification attribute. In the classification attribute, a value of 1 indicates "not cancer", and a value of 2 for "cancer".

A. Data Preprocessing

The first stage in data pre-processing is classification mapping to check the class distribution. It can be seen in Figure 2 that the distribution between classes is relatively balanced. The next step is to handle missing data (missing values). This dataset does not have any missing values initially. However, human adiponectin is generally around 3 μg/mL to 30 μg/mL [19]. If there is an adiponectin level of more than 30 μg/mL in the dataset, then the data is considered a missing value. In addition to adiponectin, in the serological Enzyme-linked immunosorbent assay (ELISA), there is a tool designed to detect and measure MCP-1 levels for humans. This tool can only detect around 15.6 pg/mL to 1,000 pg/mL MCP-1 in the human body [20]. This study assumes MCP-1 levels of more than 1000 pg/mL as missing values.

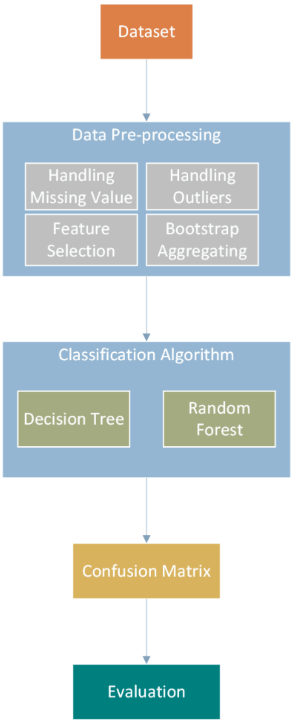

Figure 1. Research Flow

Based on that criteria, this dataset has 12 missing values (9 for MCP.1, and 3 for adiponectin). After knowing the number of missing values, a skewness check will be carried out. The skewness for adiponectin is quite significant with a value of 1.1579, so the missing value will be imputed using the median value (middle value). While the skewness for MCP.1 is not too large with a skewness value of 0.3575, so the missing value is imputed using the mean (average value).

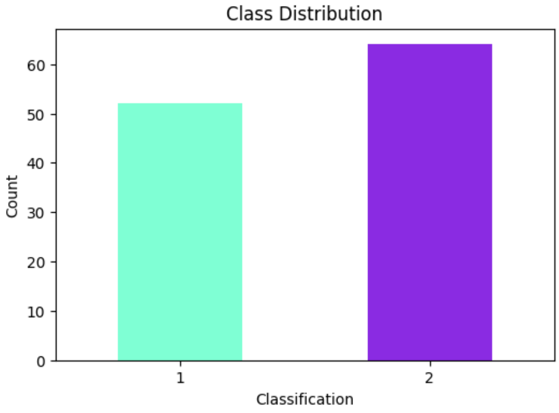

Figure 2. Class Distribution for all dataset

A boxplot is a powerful visualization tool that provides insights into data distribution, including the median, quartiles, and extreme values. Outliers are typically represented as points lying outside the interquartile range (IQR) on a boxplot. The boxplot visualization used to detect outliers is shown in Figure 3, where each feature is mapped to give a clear overview of the data distribution and the presence of extreme values. These extreme data points are then further analyzed to determine how they should be handled—whether they should be removed, replaced, or transformed—based on their relevance and potential impact on the model.

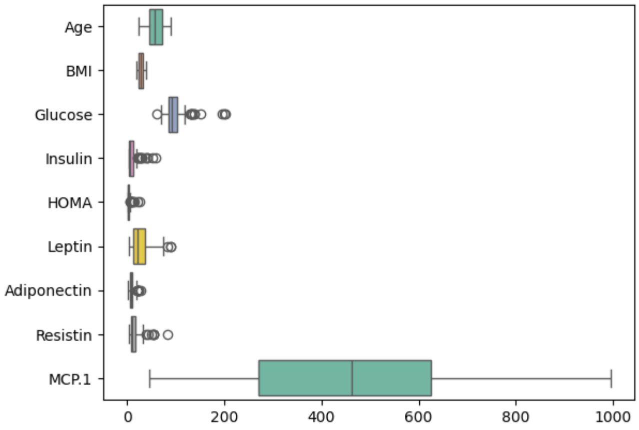

Figure 3. Outliers visualization boxplot for each attribute

To check for outliers in each column, it can be used the Interquartile Range (IQR) formula to determine the lower bound and upper bound, with the formula:

$$IQR = Q1 - Q3 \quad (1)$$

$$Lower\ bound = Q1 - (1.5 \times IQR) \quad (2)$$

$$Upper\ bound = Q3 + (1.5 \times IQR) \quad (3)$$

Data outside these limits are considered outliers, and will be considered missing values. These data will then be replaced (impute) with the mean value of each attribute column. After handling missing values and outliers, the feature selection process is carried out by removing attributes that do not have a large correlation with the classification results. Feature selection will begin by selecting features using KBest (K highest score). The SelectKBest method is a feature selection technique designed to identify and retain the most relevant features in a dataset based on their scores as determined by a specific scoring function. In this process, the scoring function evaluates the relationship between each feature and the target variable, assigning a numerical score that reflects their relevance to the classification or regression task. In this context, since the dataset in this study are numerical features, the **`f_classif`** function from the scikit-learn library is utilized as the scoring metric. For each feature, **`f_classif`** computes the ANOVA F-value by comparing the means of the feature's values across different classes in the target variable. Features that show a significant difference in means across classes (i.e., strong correlation with the target) will have higher F-statistic scores. The results of KBest will be mapped into a heatmap to make it easier to see the correlation between attributes. The attributes that will be trimmed are attributes with correlation values close to 0.

Figure 4 shows the results of the attribute correlation visualization with a heatmap where in this study the attributes with correlation values close to 0 are age, BMI, leptin, adiponectin, and MCP.1. These attributes will be trimmed and stored in different variables from the intact dataset. This aims to compare the performance of the model with the trimmed dataset and the intact dataset.

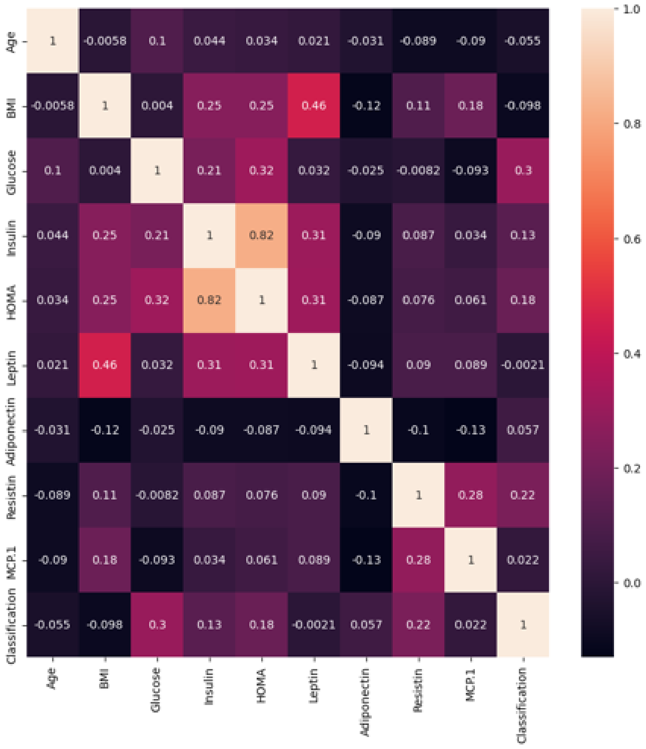

Figure 4. Heatmap Visualization of Attributes Correlation

B. *Model Evaluation*

After preprocessing phase, model evaluation is conducted. A 10-fold cross-validation technique is employed in this study to ensure the model's generalizability. This method is recognized for its ability to produce robust and unbiased evaluations of the model performance. 10-fold cross-validation works by dividing the dataset into ten equal parts, or folds. Each fold serves as a test set exactly once, while the remaining nine folds will be used for training. To obtain a robust assessment of the model's predictive abilities, this process is executed ten times. The performance metrics collected from each iteration are then averaged to arrive at a comprehensive measure of the model's predictive performance [21]. The use of 10-fold cross-validation in this study aims to reduce overfitting and ensure that model performance is not overly dependent on a particular data partition. By testing on different subsets of the data, this method allows for a more accurate assessment of how well the model will generalize to unseen data. Furthermore, this method ensures that all data points are used for training and validation, optimizing dataset utilization, which is especially important in scenarios with limited data availability.

After doing 10-fold cross-validation, the data in the dataset will be divided into 80% for training the model and 20% for testing the model. Both datasets are set with random state 1, which means that every time the program is run, the order of the data in the dataset will be the same, to facilitate the debugging process and get more certain results. If the random state is not declared, the program will continue to randomize the data order according to the seed generated at the time the program is run.

Given the relatively small size of the dataset in this study, techniques to prevent overfitting are crucial. Small datasets may not accurately reflect the true diversity and distribution of real-world data. This can lead to models learning spurious patterns specific to the training data, potentially including noise and outliers. Therefore, in addition to dividing the dataset, bootstrap aggregating (bagging) also occurs at this stage.

Bagging, as introduced by Breiman in 1996, is a straightforward yet effective technique for creating an ensemble of classifiers. Its primary goal is to enhance the performance of a classifier by aggregating the outputs of multiple models. In bagging, an ensemble is constructed using a single base learner (inducer), which generates multiple hypotheses. These hypotheses are produced independently, as each iteration of the training process involves the inducer working on a randomly selected subset of instances from the training data, with replacement. To ensure the model has adequate data for training, the size of each subset is equal to the size of the original training dataset. This approach allows for overlap among the subsets, meaning that some training examples may appear in multiple subsets, while others may be excluded altogether. This randomness leads to diverse predictions by the individual classifiers within the ensemble [22]. Bagging takes random samples in a dataset, and may re-take rows of data that have been taken or rows of data not taken at all. Bagging will produce a new dataset, and will be stored in a different variable from the truncated and intact datasets.

Model performance will be evaluated using four metrics, namely accuracy, precision, recall, and F1-score. The formulas for these indicators are as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (4)$$

$$Precision = \frac{TP}{TP + FP} \quad (5)$$

$$Recall = \frac{TP}{TP + FN} \quad (6)$$

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (7)$$

III. Results And Discussions

Model development with the Decision Tree algorithm is carried out using the DecisionTreeClassifier from the sklearn library where the first parameter, namely the criterion parameter, states the function to measure the split quality of the tree to be built and the second parameter used is max_depth which states the maximum depth of the tree. The criterion used is entropy with max_depth = 3. Because the amount of data in the dataset is quite small, the max_depth used is three to avoid the risk of overfitting. Where with a small dataset, if the decision tree is allowed to grow to the maximum depth (without limit), the model will tend to learn all the details or noise in the data. As a result, the model becomes too complex so that it is only suitable for training data, but has poor performance on new data (test data).

The development of the model with the Random Forest algorithm was carried out using the RandomForestClassifier from the sklearn library where the parameters used were n_estimators, criterion, and max_depth. The n_estimators parameter is the number of decision trees in a random forest. The more decision trees, the better the performance. Because the amount of data is quite small, this model only uses ten decision trees. The number of n_estimators that is too large will cause overfitting in the model. max_depth for the RF model is five, this is because RF is random, so max_depth is greater than the DT model so that RF performance is more stable.

Table 1.
10-fold Cross Validation Results

Algorithm	Dataset	Accuracy	Precision	Recall	F1 score
Decision Tree (DT)	All Dataset	0.7068	0.7138	0.6333	0.6543
	Trimmed Dataset	0.5500	0.5317	0.4833	0.4740
Random Forest (RF)	All Dataset	0.7409	0.7148	0.7133	0.7036
	Trimmed Dataset	0.5841	0.4750	0.5400	0.4979

Table 1 presents the evaluation metrics of two machine learning algorithms, Decision Tree (DT) and Random Forest (RF), on two different datasets: the complete dataset and a trimmed dataset. The evaluation was conducted using a 10-fold cross-validation technique. On all dataset, DT achieved an accuracy of 70.68%, precision of 71.38%, recall of 63.33%, and an F1-score of 65.43% while the performance dropped significantly with an accuracy of 55.00%, precision of 53.17%, recall of 48.33%, and an F1-score of 47.40% on trimmed dataset. Random Forest, however, outperformed the Decision Tree on the complete dataset with an accuracy of 74.09%, precision of 71.48%, recall of 71.33%, and an F1-score of 70.36%. Similar to the Decision Tree, the Random Forest model also experienced a performance drop on the trimmed dataset, achieving an accuracy of 58.41%, precision of 47.50%, recall of 54.00%, and an F1-score of 49.79%.

Table 2.
Evaluation Results

Algorithm	Dataset	Accuracy	Precision	Recall	F1 score
Decision Tree (DT)	All Dataset	0.583333	0.166667	0.166667	0.166667
	Trimmed Dataset	0.583333	0.3	0.5	0.375
	Bootstrap Dataset	0.801724	0.837209	0.692308	0.757895
Random Forest (RF)	All Dataset	0.75	0.5	0.666667	0.571429
	Trimmed Dataset	0.583333	0.3	0.5	0.375
	Bootstrap Dataset	0.896552	0.916667	0.846154	0.871287

Table 2 shows the evaluation results for all kind of dataset. For all dataset, the Decision Tree model achieved an accuracy of 0.583333, indicating that the model correctly predicted approximately 58.33% of all test data. The precision value of 0.166667 implies that only 16.67% of the positive predictions made by the model were actually correct. With a recall of 0.166667, the model identified only 16.67% of the actual positive cases in the dataset. An F1 Score of 0.166667 reflects a very low balance between precision and recall for this dataset. The Random Forest model achieved an accuracy of 0.75, which is higher than the Decision Tree on the same dataset. Precision was 0.5, meaning that 50% of the positive predictions were correct. The recall was 0.666667, indicating the model identified 66.67% of the actual positive cases. The F1 Score of 0.571429 reflects a moderate balance between precision and recall.

For the trimmed dataset, the accuracy remained the same as in the All Dataset, at 0.583333. Precision increased to 0.3, suggesting improved performance in predicting positive cases compared to the All Dataset. Recall also increased to 0.5, indicating better ability to capture positive cases. The F1 Score improved to 0.375, demonstrating a better balance between precision and recall compared to the All Dataset. With Random Forest model, the accuracy decreased to 0.583333, matching the performance of the Decision Tree on the same dataset. Precision remained at 0.3, similar to the Decision Tree on the Trimmed Dataset. Recall also remained at 0.5, showing no difference compared to the Decision Tree. The F1 Score was 0.375, identical to the Decision Tree, indicating no significant advantage of using Random Forest on this dataset.

Finally, for the boostrap dataset, the Decision Tree model showed a significant improvement in accuracy, reaching 0.801724, meaning it correctly predicted approximately 80.17% of the test data. Precision drastically increased to 0.837209, indicating the model performed very well in predicting positive cases with much lower error rates. Recall reached 0.692308, showing the model identified 69.23% of the actual positive cases. With an F1 Score of 0.757895, the model achieved a much better balance between precision and recall compared to the previous datasets. The Random Forest model's accuracy significantly improved to 0.896552, meaning it correctly predicted nearly 89.66% of the test data. Precision reached 0.916667, indicating that 91.67% of the positive predictions were correct. Recall was 0.846154, showing the model identified approximately 84.62% of all positive cases in the dataset. With an F1 Score of 0.871287, the model demonstrated excellent balance between precision and recall.

Next, statistical analysis needs to be performed to show whether the performance differences between RF and DT are statistically significant using McNemar's Test. Table 3 shows the confusion matrix for Random Forest and Decision Tree using the bootstrap dataset. Using the data in the confusion matrix, McNemar's test is conducted to find the $p - value$. After conducting the McNemar's test it is found that the $p - value$ is 0.01612. Meaning that there is a statistically significant difference between the two models.

TABLE 3
CONFUSION MATRIX RESULTS WITH BOOTSTRAP DATASET

Algorithm	TP	FP	TN	FN
Decision Tree (DT)	60	4	30	22
Random Forest (RF)	56	8	48	4

Figure 5 shows the Receiver Operating Characteristic (ROC) Curve for two models, namely Decision Tree (DT) and Random Forest (RF). The ROC Curve illustrates the relationship between True Positive Rate (TPR) (y-axis) and False Positive Rate (FPR) (x-axis) at various decision thresholds. Model performance can also be assessed through the Area Under the Curve (AUC), where a higher value indicates better performance in distinguishing positive and negative classes. The AUC value for DT in this study is 0.4352, while RF is 0.3843. The DT curve shows better performance than the diagonal line (baseline), but is not smooth, indicating limitations in capturing complex patterns.

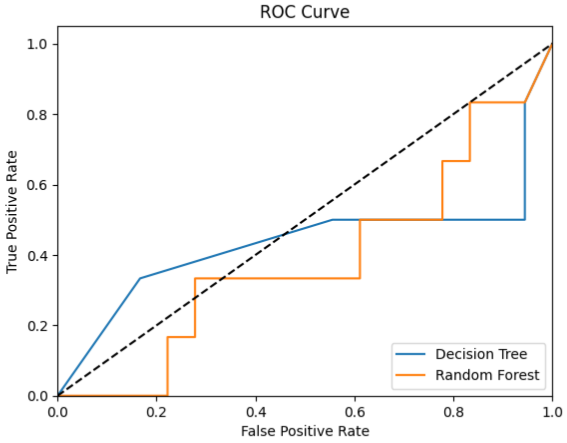

Figure 5. AUC-ROC curves for DT and RF models without bootstrapping

TPR and FPR at some points indicate that this model can detect some positive classes well, although at certain thresholds the FPR increases significantly. The AUC for DT is below Random Forest, confirming that Decision Tree's performance is not as good as Random Forest in this classification task. The RF curve shows a slower increase than DT at the beginning, but remains above the baseline (diagonal line). Although RF's performance seems less consistent at low FPR, this model shows better ability at certain points to achieve higher TPR with smaller FPR. The AUC for RF is larger than DT, indicating that RF has an advantage in detecting positive classes more accurately.

The graphs presented in Figure 6 show the ROC curves for two classification models—Decision Tree (DT) and Random Forest (RF)—after bootstrap resampling. The AUC value obtained for DT with bootstrap is 0.8294, and the value for RF with bootstrap is 0.9312. The DT model shows a sharp increase in TPR at low FPR values, then levels off. This indicates good performance in the early stages, but struggles to maintain performance when the threshold changes. This behavior may be due to overfitting or sensitivity to noise in the data. The ROC curve of the Random Forest model is overall above the curve of the Decision Tree model. This indicates that the Random Forest model performs better in classifying the data. The Random Forest model is able to achieve higher TPR at lower FPR, meaning that it is better at identifying true positive instances without overclassifying negative instances as positive.

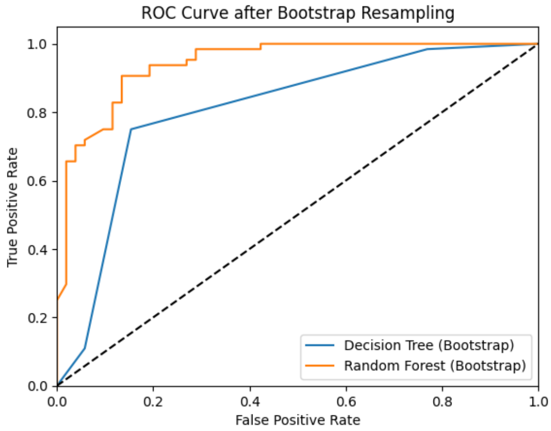

Figure 6. ROC curves for DT and RF models with bootstrap

The results presented in Table 1 highlight the differences in performance between the Decision Tree (DT) and Random Forest (RF) algorithms across datasets processed with different techniques. On the unprocessed (all) dataset, the DT algorithm exhibits poor performance, with Accuracy at 58.33% and equally low Precision, Recall, and F1-Score values of 16.67%. These results suggest that the DT model struggled to identify meaningful patterns, likely due to the presence of noise, irrelevant features, or imbalanced class distributions in the raw dataset. The poor ability to distinguish between classes is evident in the low Precision and Recall values, indicating significant misclassification of both positive and negative cases.

When the dataset is trimmed, the DT model shows modest improvement, with Precision increasing to 30%, Recall to 50%, and an F1-Score of 37.5%, reflecting a better trade-off between precision and recall. However, the unchanged Accuracy (58.33%) indicates that the trimming process, while helping reduce noise, may have also removed critical information, limiting the model's ability to capture patterns comprehensively. A remarkable improvement is observed in the DT performance on the bootstrap dataset, with Accuracy increasing to 80.17%, Precision to 83.72%, Recall to 69.23%, and an F1-Score of 75.79%. This suggests that the bootstrap technique successfully reduced overfitting and enhanced the model's generalization by providing diverse training subsets that capture more robust relationships in the data.

The RF algorithm consistently outperforms DT across all datasets, emphasizing its robustness in handling complex data. On the unprocessed dataset, RF achieves an Accuracy of 75%, Precision of 50%, Recall of 66.67%, and an F1-Score of 57.14%, reflecting its ability to manage noise and extract significant patterns even without preprocessing. The use of ensemble methods in RF enables better decision boundaries by averaging multiple trees, mitigating the impact of overfitting commonly observed in DT. On the trimmed dataset, RF performance drops significantly, with metrics aligning closely with those of DT (Accuracy = 58.33%, Precision = 30%, Recall = 50%, and F1-Score = 37.5%). This decline likely stems from the removal of important features or instances during trimming, which limits the algorithm's ability to construct robust decision boundaries.

RF shows its strongest performance on the bootstrap dataset, achieving an Accuracy of 89.65%, Precision of 91.67%, Recall of 84.62%, and an F1-Score of 87.13%. These results highlight the algorithm's ability to capitalize on the diverse training samples generated by the bootstrap technique, which not only enhances feature utilization but also improves the model's stability and accuracy. The high Precision indicates that RF minimizes false positives effectively, while the high Recall demonstrates its capability to capture a significant portion of actual positives. The superior performance of RF compared to DT across all datasets underscores the advantage of ensemble methods, particularly when paired with techniques like bootstrapping, which address overfitting and enable the model to generalize better in complex datasets.

The significant improvement in performance observed for both algorithms after applying the bootstrap technique highlights the critical role of preprocessing in machine learning. It emphasizes that well-preprocessed data, especially with techniques like bootstrapping, can greatly enhance model generalization and reduce noise. The consistent superior performance of Random Forest across all datasets demonstrates the robustness of ensemble methods compared to single-model algorithms like Decision Tree. This can lead to a discussion about how Random Forest leverages multiple decision trees to overcome the limitations of overfitting and noise sensitivity in Decision Tree. Moreover, both Decision Tree and Random Forest experienced a drop in performance when trained on the trimmed dataset. This could be attributed to the loss of critical information during the trimming process, sparking a discussion on the trade-off between noise reduction and information preservation during preprocessing.

The Random Forest algorithm's ability to perform well even on noisy or unprocessed datasets suggests its suitability for small and noisy datasets. This could open discussions on the practicality of RF in scenarios where extensive data preprocessing is not feasible. The differences in Precision, Recall, and F1-Score across datasets provide insights into each algorithm's strengths and weaknesses. For example, Random Forest demonstrated higher Precision and Recall on the bootstrap dataset, making it ideal for applications where both false positives and false negatives are critical.

IV. CONCLUSION

In this study, the Random Forest algorithm significantly performed better than Decision Tree. Performance evaluation was performed using accuracy, precision, recall, and F1-score metrics. Both algorithms achieved the best performance on the bootstrapped dataset. Random Forest achieved the highest score with an accuracy of 89.65%, precision of 91.66%, recall of 84.61%, and F1-score of 87.12%. This shows that adjusting the dataset through the bootstrap technique can significantly improve model performance. Overall, Random Forest on the bootstrap dataset was the most effective for detecting cancer in this dataset, with better results than Decision Tree which had an accuracy of 80.17%, precision of 83.72%, recall of 69.23%, and F1-score of 75.78%. Based on the results of this study, Random Forest is more suitable for data with complex patterns or interacting features. This is because the ensemble approach allows the model to capture patterns more accurately. Decision Tree, although simpler, shows quite good results but is more susceptible to overfitting, especially on small or noisy datasets.

The results of this study also show that the use of bootstrap resampling has provided benefits in improving the stability and generalization of both models. For the development of this research in the future, it is recommended to expand the dataset with more diverse variations to improve the accuracy and generalization of the model such as combining medical image data and genomic analysis. In addition, the application of deep learning methods such as CNN or transfer learning can be explored to improve detection performance. Hyperparameter optimization and deeper feature selection can also be done to strengthen the prediction results. The development of web-based or mobile applications integrated with hospital systems will expand the application of this model in the clinical environment. This development is expected to increase the effectiveness of the breast cancer early detection system and support more accurate medical decision making.

REFERENCES

[1] Ferlay J *et al.*, “Global Cancer Observatory: Cancer Today,” Lyon, France: International Agency for Research on Cancer. Accessed: Jun. 01, 2024. [Online]. Available: https://gco.iarc.who.int/today

[2] National Cancer Institute, “What Is Cancer?,” National Cancer Institute at the National Institutes of Health. Accessed: Jun. 03, 2024. [Online]. Available: https://www.cancer.gov/about-cancer/understanding/what-is-cancer

[3] K. V Shiny, A. K. Ajnabi, A. Kumar, B. K. Singh, and A. Gupta, “A Machine Learning Approach for Breast Cancer Detection using Random Forest Algorithm,” *International Journal of Research in Engineering, Science and Management*, vol. 7, no. 4, pp. 14–18, 2024.

[4] J. Crisostomo *et al.*, “Hyperresistinemia and metabolic dysregulation: a risky crosstalk in obese breast cancer,” *Endocrine*, vol. 53, pp. 433–442, 2016.

[5] H. Sun, C. Yin, Q. Liu, F. Wang, and C. Yuan, “Clinical significance of routine blood test-associated inflammatory index in breast cancer patients,” *Med Sci Monit*, vol. 23, p. 5090, 2017.

[6] Y.-Y. Wang, A. C. Hung, S. Lo, and S.-S. F. Yuan, “Adipocytokines visfatin and resistin in breast cancer: Clinical relevance, biological mechanisms, and therapeutic potential,” *Cancer Lett*, vol. 498, pp. 229–239, 2021.

[7] S. J. Russell and P. Norvig, *Artificial intelligence: a modern approach*. Pearson, 2022.

[8] D. B. Rarasati and J. C. A. Putra, “Correlation between Twitter sentiment analysis with three kernels using algorithm support vector machine (SVM) governor candidate electability level,” in *AIP Conference Proceedings*, AIP Publishing, 2023.

[9] B. Hakim, “Analisa Sentimen Data Text Preprocessing Pada Data Mining Dengan Menggunakan Machine Learning,” *Jbase-Journal of business and audit information systems*, vol. 4, no. 2, 2021.

[10] A. Thenata and M. Suryadi, “Machine Learning Prediction of Anxiety Levels in the Society of Academicians During the Covid-19 Pandemic,” *Jurnal Varian*, vol. 6, no. 1, Nov. 2022, doi: https://doi.org/10.30812/varian.v6i1.2149.

[11] D. Sulaiman and T. Mulyana, “Web-Based Writing Learning Application of Basic Hanacaraka Using Convolutional Neural Network Method,” *Ultimatics : Jurnal Teknik Informatika*, vol. 15, no. 1, Jun. 2023, doi: https://doi.org/10.31937/ti.v15i1.2993.

[12] C. Kaur and U. Garg, “Artificial intelligence techniques for cancer detection in medical image processing: A review,” *Mater Today Proc*, vol. 81, pp. 806–809, 2023.

[13] A. B. Nassif, M. A. Talib, Q. Nasir, Y. Afadar, and O. Elgendy, “Breast cancer detection using artificial intelligence techniques: A systematic literature review,” *Artif Intell Med*, vol. 127, p. 102276, 2022.

[14] D. Patel, Y. Shah, N. Thakkar, K. Shah, and M. Shah, “Implementation of artificial intelligence techniques for cancer detection,” *Augmented Human Research*, vol. 5, pp. 1–10, 2020.

[15] M. Shehab *et al.*, “Machine learning in medical applications: A review of state-of-the-art methods,” *Comput Biol Med*, vol. 145, p. 105458, 2022.

[16] K. Marias, “The constantly evolving role of medical image processing in oncology: from traditional medical image processing to imaging biomarkers and radiomics,” *J Imaging*, vol. 7, no. 8, p. 124, 2021.

[17] T.-H. Lee, A. Ullah, and R. Wang, “Bootstrap aggregating and random forest,” *Macroeconomic forecasting in the era of big data: Theory and practice*, pp. 389–429, 2020.

[18] Y. Zhao and R. Duangsoithong, “Empirical analysis using feature selection and bootstrap data for small sample size problems,” in *2019 16th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, IEEE, 2019, pp. 814–817.

[19] M. Choubey and P. Bora, “Emerging role of adiponectin/AdipoRs signaling in choroidal neovascularization, age-related macular degeneration, and diabetic retinopathy,” *Biomolecules*, vol. 13, no. 6, p. 982, 2023.

[20] Q. Dong, Y. Li, J. Chen, and N. Wang, “Azilsartan suppressed LPS-induced inflammation in U937 macrophages through suppressing oxidative stress and inhibiting the TLR2/MyD88 signal pathway,” *ACS Omega*, vol. 6, no. 1, pp. 113–118, 2020.

[21] J. M. Kernbach and V. E. Staartjes, “Foundations of machine learning-based clinical prediction modeling: Part II—Generalization and overfitting,” *Machine Learning in Clinical Neuroscience: Foundations and Applications*, pp. 15–21, 2022.

[22] U. S. Bhutamapuram and R. Sadam, “With-in-project defect prediction using bootstrap aggregation based diverse ensemble learning technique,” *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 10, pp. 8675–8691, 2022.