Jurnal Teknik Informatika (JUTIF)
P-ISSN: 2723-3863
E-ISSN: 2723-3871

Vol. 6, No. 5, October 2025, Page. 3392-3404
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4877

Comparative Evaluation of Decision Tree and Random Forest for Lung Cancer Prediction Based on Computational Efficiency and Predictive Accuracy

Muhammad Yashlan Iskandar*[1], Handoyo Widi Nugroho[2]

[1,2]Master's Program Informatics Engineering, IIB Darmajaya, Indonesia

Email: [1]yashlan007@gmail.com

Received : Jun 13, 2025; Revised : Jun 19, 2025; Accepted : Jun 24, 2025; Published : Oct 16, 2025

Abstract

Early detection of lung cancer is essential for improving treatment outcomes and patient survival rates. This paper presents a comparative evaluation of two classification algorithms: Decision Tree and Random Forest, focusing on both predictive performance and computational efficiency. The models were tested using 10-fold cross-validation to ensure robustness. Both algorithms achieved the same accuracy of 93.3%. However, Random Forest slightly outperformed Decision Tree in recall (88.8% vs. 87.9%), F1-score (92.2% vs. 92.1%), and AUC (0.94 vs. 0.91), while Decision Tree obtained higher precision (97% vs. 95.9%). In terms of computational efficiency, Decision Tree demonstrated faster training and testing times, lower memory usage, and reduced energy consumption compared to Random Forest. The results reveal a clear trade-off between prediction quality and resource usage, highlighting the importance of selecting algorithms not only for their accuracy but also for their practicality in real-world healthcare scenarios. This comprehensive evaluation provides valuable insights for developing intelligent decision support systems that are both effective and resource-efficient, especially in environments with limited computing capacity. These findings contribute to the advancement of resource-aware intelligent systems in the field of medical informatics.

Keywords : *Classification Performance, Computational Efficiency, Decision Tree, Lung Cancer, Random Forest, Supervised Learning.*

1. INTRODUCTION

Lung cancer is one of the most prevalent types of cancer and a major contributor to cancer-related deaths worldwide. By 2022, approximately 2.5 million individuals were diagnosed with this disease, resulting in over 1.8 million fatalities. The mortality rate of lung cancer is more than twice that of colorectal cancer, which ranks as the second leading cause of cancer-related deaths. However, in many cases, lung cancer is preventable. Smoking remains the primary risk factor, accounting for around 85% of cases. Additionally, exposure to hazardous substances such as second-hand smoke, indoor and outdoor air pollution, diesel engine fumes, welding emissions, and asbestos also significantly contribute to the development of lung cancer [1].

The Detection of diseases such as lung cancer is not a simple task as it involves various complex risk factors such as genetics, lifestyle, environmental conditions, and other health conditions. Conventional methods often used in diagnosis have limitations in dealing with this complexity. In addition, they are often ineffective in providing accurate predictions, require high costs, long examination times, and trained medical personnel [2]. Therefore, a more sophisticated, adaptive and efficient approach is needed to improve early detection, thereby improving the patient's chances of recovery and reducing the mortality rate from lung cancer [3].

Vol. 6, No. 5, October 2025, Page. 3392-3404
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4877

With the technological advancements, artificial intelligence, particularly machine learning, has become a new approach for diagnosing and predicting various diseases, including lung cancer [4], [5]. Machine learning, a subset of artificial intelligence, enables computers to analyze data and identify patterns that may be challenging for humans to detect. In the medical field, this technology has been utilized to enhance diagnostic accuracy, expedite the analysis of medical data, and support clinical decision-making [6]. Some machine learning algorithms that have been widely used in disease classification and prediction as well as predicting other things include methods such as Support Vector Machine (SVM), Random Forest, Naive Bayes, Decision Tree, and K-Nearest Neighbor (KNN) [7], [8], [9]. These algorithms have proven effective in analyzing complex data and producing accurate predictions to detect some diseases at an early stage [10].

In addition to evaluating accuracy, it is essential to consider other metrics that offer a more comprehensive understanding of an algorithm's efficiency and effectiveness in real-world applications, such as execution time while training or testing, memory usage, and also energy consumption. Execution time measures how fast the algorithm can process data and deliver results, which is important for applications that require quick response in clinical situations. Memory usage describes how much resources the algorithm needs to process data, which becomes an important factor when the algorithm is implemented in devices with memory limitations. Meanwhile, energy consumption indicates how efficient the algorithm is in terms of energy usage, which is an important consideration especially in systems operating with limited resources or on a large scale that require long-term operational efficiency. Therefore, although accuracy is an important factor, evaluation of these factors is also important to select an algorithm that is not only accurate but also efficient in resource usage [11], [12].

Study [13] conducted a comparison and optimization for early diagnosis and classification of lung cancer based on multi-feature clinical data, reporting an accuracy of 92% for SVM, 92% for Random Forest, and 91% for Decision Tree. Study [14] analyzed a comparison between Decision Tree and Random Forest in classifying health-related text data. The results showed an accuracy of 75% for Decision Tree and 99% for Random Forest. Study [15] compared several algorithms for flood prediction in rural areas. Random Forest achieved the best performance with an average accuracy of 99.05%, while Naive Bayes demonstrated the fastest computational time. Study [16] measuring accuracy and also measured memory and energy consumption with the best accuracy results achieved by KNN (92.88%) and Random Forest (92.73%). Naive Bayes has the lowest energy consumption (12,387 J CPU, 10,036 J DRAM) but the accuracy is low (51.05%). Decision Tree uses the least memory (2.949 MB). Study [17] conducted a performance analysis of four algorithms, with K-Nearest Neighbor (KNN) achieving the highest accuracy of 95.16%.

Unlike several previous studies, this research aims to comprehensively evaluate the performance of Decision Tree and Random Forest algorithms in predicting lung cancer by not only assessing predictive metrics such as accuracy, precision, recall, F1-score, and AUC, but also analyzing computational efficiency factors including training and testing time, memory usage during training and testing, as well as energy consumption during training and testing.

By integrating these two aspects—predictive performance and resource utilization—this study provides a more holistic assessment that is rarely explored in the context of lung cancer prediction. The findings are expected to contribute to the development of medical decision support systems that are both accurate and computationally efficient, particularly for healthcare environments with limited resources. Selecting the most appropriate algorithm based on these combined criteria will help enable practical and cost-effective solutions for early lung cancer detection and potentially reduce its associated mortality rate.

Jurnal Teknik Informatika (JUTIF)
P-ISSN: 2723-3863
E-ISSN: 2723-3871
Vol. 6, No. 5, October 2025, Page. 3392-3404
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4877

2. METHOD

In order to facilitate the course of this research, a picture is made that can provide a systematic visualization of the steps taken in this research. The following is shown in Figure 1 which is the research flow:

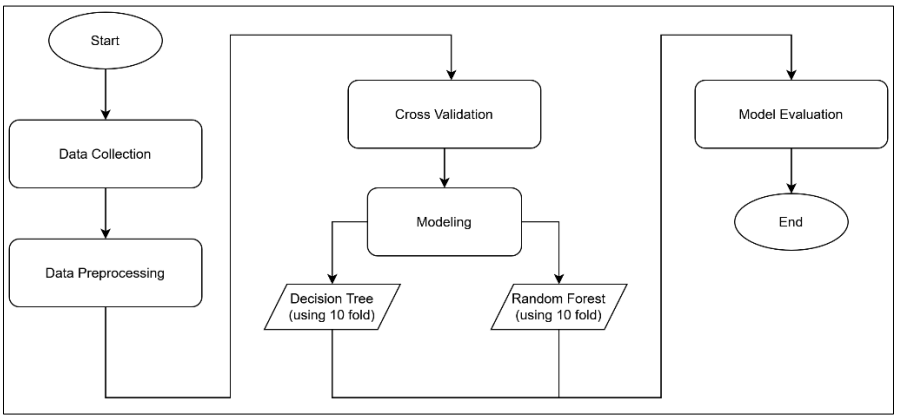

Figure 1. Research Flow

Based on Figure 1, this research process begins with data collection, where relevant lung cancer datasets are gathered from available sources. The next step is data preprocessing, which includes cleaning and preparing the dataset to ensure quality and suitability for modeling. After preprocessing, 10-fold cross-validation is applied to improve the generalizability of the model and prevent overfitting. The modeling stage involves building classification models using two algorithms: Decision Tree and Random Forest, each evaluated through the 10-fold cross-validation technique. Finally, the models are assessed in the model evaluation stage using various performance metrics such as accuracy, precision, recall, F1-score, AUC, as well as computational efficiency indicators including training time, testing time, memory usage, and energy consumption.

All experiments in this study were carried out on a single hardware platform. The stages of notebook creation, coding, training, testing, and evaluation were performed on a personal computer equipped with an Intel i7-10610U CPU @ 1.80 GHz, 16 GB RAM, and running Linux Ubuntu 24.04 LTS. Coding was conducted using Visual Studio Code with Jupyter and Python extensions to provide an integrated environment for model development, testing, and performance evaluation. To facilitate reproducibility and further development, the complete source code is publicly available at the following GitHub repository link https://github.com/yashlan/lung-cancer-prediction.

2.1. Data Collection

Datasets used in a study are generally obtained from various sources. One platform that is often used to obtain and analyze datasets is Kaggle. The dataset used must contain data that is relevant to the research objectives [18]. In this context, the dataset used to build the prediction model must be related to lung cancer detection.

2.2. Data Preprocessing

At this stage, a series of pre-processing steps will be carried out to enhance data quality. The first step involves data cleaning, which includes eliminating irrelevant or corrupted entries, such as missing or duplicate values. Subsequently, data transformation is applied to convert raw data into an appropriate format, including normalization and encoding for categorical variables. Lastly, feature selection is conducted to identify the most relevant attributes based on prior research and initial data exploration [19].

Vol. 6, No. 5, October 2025, Page. 3392-3404
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4877

2.3. Cross Validation

In this research, a 10-fold cross-validation technique was implemented to evaluate the model's performance more reliably. The dataset was split into ten equal parts. In every iteration, one part served as the test set, while the remaining nine were used for training. This process was carried out ten times, ensuring that each subset was used once as test data. This strategy helps minimize bias and maximizes the usage of available data for training. Additionally, during each iteration, the model's parameters were updated, contributing to an overall improvement in its performance [20]. The general workflow is illustrated in Figure 2.

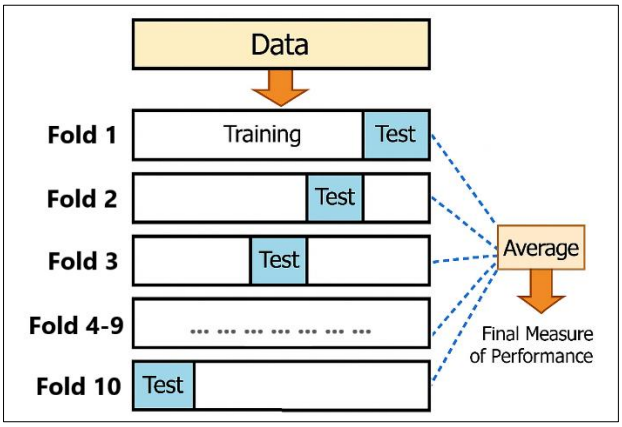

Figure 2. 10-fold Cross Validation Workflow

2.4. Modeling

During the modeling phase, the Decision Tree and Random Forest algorithms are employed to develop a predictive model. Decision Tree is a non-parametric supervised learning method commonly used for both classification and regression tasks. It features a hierarchical tree-like structure composed of root nodes, branches, internal nodes, and leaf nodes. The foundation of several widely used decision tree algorithms can be traced back to Hunt's algorithm, which was introduced in the 1960s to simulate human learning in psychology. This algorithm served as the basis for many well-known decision tree methods, including the following:

a) ID3: Developed by Ross Quinlan, the ID3 algorithm, or "Iterative Dichotomizer 3," applies entropy and information gain to assess potential splits in the data. Quinlan's research on this method, dating back to 1986, provides insights into its development and application.
b) C4.5: As an improved version of the ID3 algorithm, which was also created by Quinlan, C4.5 enhances decision tree construction by incorporating both information gain and gain ratio as criteria for determining optimal split points.
c) CART: Short for "Classification and Regression Trees," this algorithm was introduced by Leo Breiman. It commonly employs Gini impurity to determine the optimal attribute for splitting. Gini impurity quantifies the likelihood of a randomly chosen attribute being misclassified, with lower values indicating a more effective split [21].

This research will use the CART type Decision Tree method for the classification process. The CART method is used to build decision trees in classification and regression. The process includes attribute selection, data splitting, tree building, pruning to prevent overfitting, and evaluation of results to ensure good performance on new data. In decision tree development, impurity is used to select splitting attributes. One method is the Gini Index, which determines the optimal split point. A lower

Vol. 6, No. 5, October 2025, Page. 3392-3404
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4877

Gini Index value indicates a higher degree of similarity. The formula for Gini Index on a dataset with m classes is given in equation (1):

$$Gini(T) = 1 - \sum_{i=1}^{m} p_i^2 \quad (1)$$

The dataset is partitioned into two subsets based on the attribute with the lowest Gini Index value. When the data is split into two groups, D1 and D2, the Gini Index is determined using the following equation (2):

$$\text{Gini}_A(D) = \frac{|D_1|}{|D|} \cdot \text{Gini}(D_1) + \frac{|D_2|}{|D|} \cdot \text{Gini}(D_2) (2)$$

This process results in a decision tree that can classify new data by identifying patterns learned from the training dataset [22].

Random Forest, developed by Leo Breiman and Adele Cutler, is a machine learning algorithm that enhances prediction accuracy by combining multiple decision trees. It improves upon traditional bagging by introducing feature bagging, where a random subset of features is chosen at each tree split to reduce correlation and improve generalization. Before training, key hyperparameters such as the number of trees, node size, and selected features must be set. Each tree is trained using bootstrap sampling, where data points are randomly selected with replacement, while about one-third of the dataset is left out as out-of-bag (OOB) samples for performance evaluation. This method helps assess accuracy without needing a separate validation set. For regression, predictions are averaged across all trees, whereas classification relies on majority voting. By incorporating both bootstrapping and feature selection, Random Forest enhances robustness, minimizes overfitting, and remains a reliable choice for various machine learning applications. The OOB samples serve as a form of cross-validation, enhancing the reliability and accuracy of the model [23]. The following Equation (3) can be used to determine the most dominant class in Random Forest [24]:

$$f(x) = \text{Average}\big(f_1(x), f_2(x), \ldots, f_n(x)\big) \quad (3)$$

2.5. Model Evaluation

The evaluation at this stage is conducted to compare the performance of the models. By measuring accuracy, precision, recall, and F1-score metrics, this evaluation reveals the effectiveness of each model. In this experiment, standard 10-fold K-Fold Cross Validation is used. An evaluation using AUC (Area Under the Curve) was also conducted to assess the classification performance of the model. AUC has a value of 0 to 1, the closer to 1 the better the model performance. AUC provides a good measure for binary and multiple-class classification problems, indicating the ability of the model to distinguish between positive and negative categories. The following equations (4) - (7) will be used [25]:

$$Accuracy = \frac{\text{True Positive+True Negative}}{TP+TN+FP+FN} \times 100\% (4)$$

$$Precision = \frac{\text{True Positive}}{\text{True Positive+False Positive}} \times 100\% (5)$$

$$Recall = \frac{\text{True Positive}}{\text{True Positive+False Negative}} \times 100\% \quad (6)$$

$$F1\text{-}Score = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision+Recall}} \times 100\% \quad (7)$$

In addition, the computational efficiency of each algorithm is evaluated based on several performance metrics, namely execution time, memory usage, and energy consumption. Execution time

Vol. 6, No. 5, October 2025, Page. 3392-3404
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4877

refers to the duration in seconds or minutes required by the algorithm to complete one training or testing iteration, which was measured using Python's built-in time module. Memory usage indicates the amount of memory used during the process in one Megabyte (MB), tracked with the tracemalloc module [26], [27]. While energy consumption measures the amount of energy consumed by the algorithm during training or testing in Joules, obtained using the pyRAPL library, which utilizes the Running Average Power Limit (RAPL) interface [28]. This can provide important information about the practicality of each approach in detecting lung cancer. The energy equation (8) is as follows [29]:

$$\text{Energy (J)} = \text{Power (W)} \times \text{Time (s)} \qquad (8)$$

3. RESULT

3.1. Data Collection

The dataset used in this study consists of 1157 instances and a total of 16 attributes. The attributes include patient-related information such as Gender, Age, as well as various risk factors and medical symptoms experienced by the patient. The risk factors include Smoking, Yellow Finger, Anxiety, and others. Meanwhile, medical symptoms include Wheezing, Shortness of Breath, Difficulty Swallowing, Chest Pain, and Lung Cancer which is the target variable. The last column, Lung Cancer, serves as a target variable with a binary category indicating whether the patient is diagnosed with lung cancer. These columns collectively offer a detailed summary of the patient's health status, potential risk factors, and symptoms, which is crucial for evaluating and predicting the lung cancer probability by using machine learning techniques. The structure of the lung cancer dataset is shown in Figure 3.

	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC DISEASE	FATIGUE	ALLERGY	WHEEZING	ALCOHOL CONSUMING	COUGHING	SHORTNESS OF BREATH	SWALLOWING DIFFICULTY	CHEST PAIN	LUNG_CANCER
0	M	69	1	2	2	1	1	2	1	2	2	2	2	2	2	YES
1	M	74	2	1	1	1	2	2	2	1	1	1	2	2	2	YES
2	F	59	1	1	1	2	1	2	1	2	1	2	2	1	2	NO
3	M	63	2	2	2	1	1	1	1	1	2	1	1	2	2	NO
4	F	63	1	2	1	1	1	1	1	2	1	2	2	1	1	NO
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1152	F	74	1	1	1	1	1	1	1	1	1	2	1	1	1	NO
1153	F	75	1	1	1	1	1	1	1	1	1	2	1	1	1	NO
1154	F	76	1	1	1	1	1	1	1	1	1	2	1	1	1	NO
1155	F	77	1	1	1	1	1	1	1	1	1	2	1	1	1	NO
1156	F	78	1	1	1	1	1	1	1	1	1	2	1	1	1	NO

1157 rows × 16 columns

Figure 3. Lung Cancer Dataset

Figure 3 displays the structure of the lung cancer dataset, where each row represents an individual patient and each column corresponds to a specific feature, such as demographic attributes, risk factors, and symptoms. Categorical variables such as Gender are numerically encoded (1 = Male, 0 = Female), and the target variable Lung Cancer is labeled as '1' for patients diagnosed with lung cancer and '0' for those without. This representation ensures that the data is suitable for processing by machine learning algorithms.

3.2. Data Preprocessing

In this stage, the data will be converted into a dataframe, then a data type check is performed. Furthermore, it will be checked whether there are null or missing values in these features. In the Gender and Lung Cancer features, the data type needs to be changed to integer because the contents of these features must be represented in numeric form. The value on the Gender feature will be converted, with 'M' converted to 1 and 'F' to 0, while on the Lung Cancer feature, the value 'YES' will be converted to 1 and 'NO' to 0. And for the final result of preprocessing can be seen in Figure 4.

Jurnal Teknik Informatika (JUTIF)
P-ISSN: 2723-3863
E-ISSN: 2723-3871
Vol. 6, No. 5, October 2025, Page. 3392-3404
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4877

	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC DISEASE	FATIGUE	ALLERGY	WHEEZING	ALCOHOL CONSUMING	COUGHING	SHORTNESS OF BREATH	SWALLOWING DIFFICULTY	CHEST PAIN	LUNG_CANCER
0	1	69	1	2	2	1	1	2	1	2	2	2	2	2	2	1
1	1	74	2	1	1	1	2	2	2	1	1	1	2	2	2	1
2	0	59	1	1	1	2	1	2	1	2	1	2	2	1	2	0
3	1	63	2	2	2	1	1	1	1	1	2	1	1	2	2	0
4	0	63	1	2	1	1	1	1	1	2	1	2	2	1	1	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1152	0	74	1	1	1	1	1	1	1	1	1	2	1	1	1	0
1153	0	75	1	1	1	1	1	1	1	1	1	2	1	1	1	0
1154	0	76	1	1	1	1	1	1	1	1	1	2	1	1	1	0
1155	0	77	1	1	1	1	1	1	1	1	1	2	1	1	1	0
1156	0	78	1	1	1	1	1	1	1	1	1	2	1	1	1	0

1157 rows × 16 columns

Figure 4. Preprocessing Data Result

Figure 4 illustrates the final result of the preprocessing stage, where all categorical data have been numerically encoded. For example, the 'Gender' feature is represented as 1 for male and 0 for female, while the 'Lung Cancer' target variable is encoded as 1 for positive cases and 0 for negative cases. This transformation ensures compatibility with machine learning algorithms that require numerical input.

3.3. Cross Validation

To thoroughly evaluate the performance of the model and ensure more stable results, a 10-fold cross validation technique is used. The data will be divided into 10 equal parts (folds). One fold is used as test data, while the other nine folds are used as training data. The following code implementation can be seen in Figure 5.

kFold 10 Split

```
[ ]  # Angka 42 sering dipakai sebagai default seed "tradisional" di dunia pemrograman dan machine learning.
     # Ini adalah referensi dari buku "The Hitchhiker's Guide to the Galaxy" di mana 42 adalah "the answer to the
     # Menurut dokumentasi resmi juga mengatakan "Popular random seeds are 0 and 42". link: https://scikit-learn
     kf = KFold(n_splits=10, shuffle=True, random_state=42)
```

Figure 5. Cross Validation Setup

Figure 5 shows the implementation of the 10-fold cross-validation technique using the KFold function from the Scikit-learn library. The parameter n_splits=10 specifies that the data is divided into 10 folds. The shuffle=True option ensures that the data is randomly shuffled before splitting, helping to reduce bias in the training and testing sets. The random_state=42 parameter is used to maintain result reproducibility, where the number 42 is commonly used as a standard seed value in machine learning practices, as also noted in the Scikit-learn documentation. This setup helps ensure that the model evaluation is both reliable and consistent across different runs.

3.4. Model Evaluation

At this stage, the two classification models used, namely Decision Tree (DT) and Random Forest (RF), were evaluated. The evaluation is conducted from two main aspects, namely predictive performance and computational efficiency. Predictive performance was assessed based on metrics such as accuracy, and other using a 10-fold cross-validation technique to ensure stable and reliable results.

Furthermore, the computational efficiency of both models was analyzed by measuring time, memory, and CPU or DRAM consumption. The purpose of this analysis is to understand not only how well the models predict, but also how efficiently computational resources are used, so as to assess the feasibility of implementing them in real systems.

3.4.1. Model Performance Evaluation

The following are the average predictive performance evaluation results of the Decision Tree and Random Forest models based on the 10-fold cross-validation technique presented in Table 1.

Vol. 6, No. 5, October 2025, Page. 3392-3404
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4877

Table 1. Model Performances

Metric	Decision Tree	Random Forest
Accuracy (%)	93.3%	93.3%
Precision (%)	97%	95.9%
Recall (%)	87.9%	88.8%
F1-Score (%)	92.1%	92.2%
AUC (0.00)	0.91	0.94

Based on the 10-fold cross-validation results, both models have the same accuracy of 93.3%. Decision Tree recorded higher precision (97%) than Random Forest (95.9%), while Random Forest excelled in recall (88.8% vs 87.9%), F1-score (92.2% vs 92.1%), and AUC (0.94 vs 0.91). The differences between metrics are relatively small. These results suggest that while both models provide comparable accuracy, Random Forest may be preferable when sensitivity (recall) is prioritized, such as in early cancer detection scenarios where missing a positive case is critical. On the other hand, Decision Tree offers higher precision, which may reduce false positives in certain screening applications.

3.4.2. Model Efficiency Evaluation

The following are the results of the model efficiency evaluation based on execution time, memory usage, and energy consumption during the training and testing process. Details of these results can be seen in Table 2.

Table 2. Model Computational Efficiencies

Model	Execution Time (s)		Training Memory Usage (MB)		Testing Memory Usage (MB)		Training Energy Usage (J)		Testing Energy Usage (J)	
	Training	Testing	Current	Peak	Current	Peak	CPU	DRAM	CPU	DRAM
Decision Tree	0.0140	0.0066	0.0027	0.1860	0.0012	0.0230	0.0627	0.0177	0.0369	0.0121
Random Forest	0.7139	0.0386	0.0699	0.2270	0.0087	0.0248	2.8809	1.1854	0.1721	0.0666

Decision Tree has a much faster training (0.0140 s) and testing (0.0066 s) time than Random Forest (0.7139 s and 0.0386 s). In terms of memory, Decision Tree also uses less memory both during training (current 0.0027 MB, peak 0.1860 MB) and testing (current 0.0012 MB, peak 0.0230 MB) than Random Forest. For energy, Decision Tree requires less CPU and DRAM energy in training (0.0627 J and 0.0177 J) and testing (0.0369 J and 0.0121 J) than Random Forest. In terms of this computational efficiency, Decision Tree consistently outperformed Random Forest across all resource usage metrics, making it more suitable for deployment in resource-limited environments such as portable medical devices.

3.5. Comparison With Other Studies

Comparisons were made based on all performance and computational efficiency metrics, as shown in Table 3 and Table 4 below.

The Random Forest model in this study produced competitive performance compared to other studies, with an accuracy value of 93.3%, F1-score of 92.2%, and AUC of 0.94. Although some studies such as Study [15] show very high accuracy on smaller datasets, most studies do not present complete metrics, especially AUC values. This shows the importance of reporting metrics thoroughly for a more comprehensive evaluation of model performance.

Vol. 6, No. 5, October 2025, Page. 3392-3404
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4877

Table 3. Comparison of the Proposed Performance Model with Other Studies

Reference	Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	AUC (0.00)
Study [13]	Decision Tree	91%	95.7%	93.7%	83%	0.87
	Random Forest	92.8%	95.8%	95.8%	85.5%	0.96
Study [14]	Decision Tree	75.2%	-	-	-	-
	Random Forest	99.4%	-	-	-	-
Study [15]	Decision Tree	99.6%	99.6%	99.6%	98.6%	-
	Random Forest	99.4%	99.5%	99.5%	99.5%	-
Study [16]	Decision Tree	78.4%	79.5%	78.4%	74.3%	-
	Random Forest	94%	93.7%	94%	93.6%	-
Study [17]	Decision Tree	91.9%	73.6%	70%	77.7%	-
	Random Forest	91.9%	73.6%	70%	77.7%	-
This Study	Decision Tree	93.3%	97%	87.9%	92.1%	0.91
	Random Forest	93.3%	95.9%	88.8%	92.2%	0.94

Table 4. Comparison of the Proposed Computational Efficiency Model with Other Studies

Reference	Dataset Size (Instances)	Model	Execution Time (s)		Training Memory Usage (MB)		Testing Memory Usage (MB)		Training Energy Usage (J)		Testing Energy Usage (J)	
			Training	Testing	Current	Peak	Current	Peak	CPU	DRAM	CPU	DRAM
Study [13]	309	Decision Tree	-	-	-	-	-	-	-	-	-	-
		Random Forest	-	-	-	-	-	-	-	-	-	-
Study [14]	7.570	Decision Tree	-	-	-	-	-	-	-	-	-	-
		Random Forest	-	-	-	-	-	-	-	-	-	-
Study [15]	1.460	Decision Tree	4.17	0.25	-	-	-	-	-	-	-	-
		Random Forest	4.45	0.24	-	-	-	-	-	-	-	-
Study [16]	257.673	Decision Tree	-	-	2.949	4018.422	-	-	14184.381	10641.125	-	-
		Random Forest	-	-	38.149	4018.421	-	-	12749.070	10143.532	-	-
Study [17]	3.001	Decision Tree	-	-	-	-	-	-	-	-	-	-
		Random Forest	-	-	-	-	-	-	-	-	-	-
This Study	1.157	Decision Tree	0.0140	0.0066	0.0027	0.1860	0.0012	0.0230	0.0627	0.0177	0.0369	0.0121
		Random Forest	0.7139	0.0386	0.0699	0.2270	0.0087	0.0248	2.8809	1.1854	0.1721	0.0666

Vol. 6, No. 5, October 2025, Page. 3392-3404
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4877

Table 4 shows that in Study [15], Decision Tree has a faster training time than Random Forest. In Study [16], the current memory usage for Decision Tree is also smaller, similar to the results in this study. However, in terms of energy consumption during training, Random Forest in Study [16] showed lower CPU and DRAM energy usage than Decision Tree. In contrast, in this study, Decision Tree actually consumed lower CPU and DRAM energy than Random Forest. This difference shows that computational efficiency is not only affected by the type of algorithm, but also by the implementation context and the characteristics of the dataset used.

4. DISCUSSIONS

This study systematically compared the predictive performance and computational efficiency of Decision Tree and Random Forest algorithms in predicting lung cancer based on a dataset consisting of 1,157 patient records. The evaluation was conducted using a 10-fold cross-validation technique to ensure reliable and consistent results.

4.1. Predictive Performance Analysis

The results show that both algorithms achieved the same accuracy (93.3%). However, Random Forest outperformed Decision Tree in several other metrics, including recall (88.8% vs. 87.9%), F1-score (92.2% vs. 92.1%), and AUC (0.94 vs. 0.91). This indicates that Random Forest is better at identifying positive lung cancer cases, making it a more suitable option for medical applications that require high sensitivity.

Although Decision Tree recorded a slightly higher precision (97%) than Random Forest (95.9%), its lower recall suggests it is more prone to missing actual positive cases, which is a critical limitation in cancer diagnosis scenarios.

4.2. Computational Efficiency Analysis

In terms of computational efficiency, Decision Tree was significantly faster and lighter. The model training time for Decision Tree was only 0.0140 seconds, and the testing time was 0.0066 seconds, whereas Random Forest required 0.7139 seconds for training and 0.0386 seconds for testing. Memory usage and energy consumption were also considerably lower in Decision Tree compared to Random Forest, confirming its suitability for environments with limited computational resources.

These findings highlight the trade-off between accuracy and efficiency, where Decision Tree excels in lightweight computation but with slightly reduced robustness, and Random Forest provides higher prediction quality at the cost of more resources.

4.3. Comparison with Previous Studies

The performance of the models in this study is competitive when compared to previous works. While some studies reported higher accuracy values, most did not include complete evaluation metrics, especially AUC, energy consumption, and memory usage as presented in this study (see [13]-[17]).

Unlike most previous studies that focused solely on predictive accuracy, this study provides a novel contribution by offering a dual-perspective evaluation that includes both predictive performance and computational efficiency. It also incorporates metrics such as memory usage and energy consumption, which are rarely reported in related research. These additions offer practical insights for selecting machine learning models suitable for real-world healthcare systems, particularly those with limited computational resources.

This dual-perspective evaluation contributes to the development of resource-aware intelligent systems in medical informatics, particularly for early lung cancer detection. However, this study has certain limitations, as the dataset used was obtained from Kaggle, an open-source data platform, rather than from real clinical environments. As a result, the models may not fully capture the variability and

Vol. 6, No. 5, October 2025, Page. 3392-3404
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4877

complexity present in actual hospital or clinical data, which could affect their generalizability to real-world medical applications.

5. CONCLUSION

This study has demonstrated that both Decision Tree and Random Forest algorithms are capable of producing high accuracy in predicting lung cancer based on clinical datasets. Nevertheless, Random Forest consistently outperformed Decision Tree in several critical metrics, particularly recall, F1-score, and AUC. These results suggest that Random Forest provides a more balanced and reliable predictive capability, especially when the primary objective is to correctly identify patients at risk.

Conversely, the Decision Tree algorithm exhibited considerably better computational efficiency. It required less memory, consumed less energy, and executed faster, making it a more practical choice for deployment in environments with limited computational resources or in systems that demand real-time processing.

Therefore, the selection of the most appropriate algorithm should align with the intended application. For scenarios that prioritize predictive reliability and robustness, Random Forest is recommended. In contrast, when computational efficiency and speed are more critical, Decision Tree can serve as a viable and effective alternative.

This research enriches the understanding of machine learning algorithm suitability for real-time medical diagnostic applications, particularly in resource-constrained healthcare environments. Future research is encouraged to extend this work by investigating more sophisticated ensemble learning methods such as XGBoost or LightGBM, as well as incorporating explainable AI techniques to improve model transparency and trustworthiness. Furthermore, validation using actual clinical data from hospitals is crucial to ensure the applicability and generalizability of these models in real-world medical settings.

REFERENCES

[1] International Agency for Research on Cancer (IARC), “Lung Cancer.” Accessed: Feb. 06, 2025. [Online]. Available: https://www.iarc.who.int/cancer-type/lung-cancer/

[2] E. Safitri, D. Rofianto, N. Purwati, H. Kurniawan, and S. Karnila, “Prediksi Penyakit Diabetes Melitus Menggunakan Algoritma Machine Learning,” *Jurnal Sistem dan Teknologi Informasi (JUSTIN)*, vol. 12, no. 4, pp. 760–766, Oct. 2024, doi: 10.26418/justin.v12i4.84620.

[3] A. C. Pacurari *et al.*, “Diagnostic Accuracy of Machine Learning AI Architectures in Detection and Classification of Lung Cancer: A Systematic Review,” *Diagnostics*, vol. 13, no. 13, pp. 1–12, Jul. 2023, doi: 10.3390/diagnostics13132145.

[4] Y. Wiratama and R. Abdul Aziz, “Perbandingan Prediksi Penyakit Stunting Balita Menggunakan Algoritma Support Vektor Machine dan Random Forest,” *Technology and Science (BITS)*, vol. 6, no. 2, pp. 1159–1168, Sep. 2024, doi: 10.47065/bits.v6i2.5543.

[5] Khodijah, Sriyanto, R. Abdul Aziz, and Suhendro, “Perbandingan Kinerja Lima Algoritma Klasifikasi Dasar Untuk Prediksi Penyakit Jantung ‘Classifier: NB, DTC4.5, KNN, ANN & SVM,’” *Jurnal Jaringan Sistem Informasi Robotik (JSR)*, vol. 8, no. 2, pp. 230–234, Sep. 2024, doi: 10.58486/jsr.v8i2.

[6] I. Akbar, F. Supriadi, and D. I. Junaedi, “Pemanfaatan Machine Learning di Bidang Kesehatan,” *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 9, no. 1, pp. 1744–1749, Feb. 2025, doi: 10.36040/jati.v9i1.12663.

[7] M. Andani, J. Triloka, S. Y. Irianto, and H. W. Nugroho, “Performance Comparison of K-Nearest Neighbor, Naive Bayes, and Random Forest Algorithms in Obesity Prediction,” *Sinkron: Jurnal dan Penelitian Teknik Informatika*, vol. 9, no. 1, Jan. 2025, doi: 10.33395/sinkron.v9i1.14478.

[8] M. S. Hasibuan and D. Fransisca, “Prediksi Stroke Otak Menggunakan Algoritma Naïve Bayes dan Particle Swarm Optimization (PSO),” *INTEGER: Journal of Information Technology*, vol. 9, no. 1, pp. 109–118, Mar. 2024, doi: 10.31284/j.integer.0.v9i1.5738.

Vol. 6, No. 5, October 2025, Page. 3392-3404
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4877

[9] M. Haris Luthfi and Chairani, "Penerapan Algoritma C4.5 Berbasis Particle Swarm Optimization (PSO) Untuk Deteksi Kanker Payudara," *JUPITER: Jurnal Penelitian Ilmu Dan Teknologi Komputer*, vol. 16, no. 2, pp. 613–622, Oct. 2024, doi: 10.5281/zenodo.13293601.

[10] J. Fahmi Idris, R. Ramadhani, and M. Malik Mutoffar, "Klasifikasi Penyakit Kanker Paru Menggunakan Perbandingan Algoritma Machine Learning," *JURNAL MEDIA AKADEMIK (JMA)*, vol. 2, no. 2, pp. 1981–2000, Feb. 2024, doi: 10.62281/v2i2.145.

[11] M. F. A. Sayeedi, J. F. Deepti, A. M. I. M. Osmani, T. Rahman, S. S. Islam, and M. M. Islam, "A Comparative Analysis for Optimizing Machine Learning Model Deployment in IoT Devices," *Applied Sciences*, vol. 14, no. 13, pp. 1–18, Jul. 2024, doi: 10.3390/app14135459.

[12] N. V. D. S. S. V. Prasad Raju and P. N. Devi, "A Comparative Analysis of Machine Learning Algorithms for Big Data Applications in Predictive Analytics," *International Journal of Scientific Research and Management (IJSRM)*, vol. 12, no. 10, pp. 1608–1630, Oct. 2024, doi: 10.18535/ijsrm/v12i10.ec09.

[13] Q. Lan, R. Wang, and H. Fan, "Early Diagnosis and Classification of Lung Cancer Driven by Multi-Feature Data: A Comparison and Optimization of Three Machine Learning Methods," *J Mech Med Biol*, vol. 24, no. 9, pp. 1–21, Nov. 2024, doi: 10.1142/S0219519424400797.

[14] H. Oktavianto, H. W. Sulistyo, G. Wijaya, D. Irawan, and G. Abdurrahman, "Analisis Komparasi Kinerja Metode Decision Tree dan Random Forest dalam Klasifikasi Teks Data Kesehatan," *Bina Insani ICT Journal*, vol. 11, no. 1, pp. 56–65, Jun. 2024, doi: 10.51211/biict.v11i1.2928.

[15] M. B. A. Darmawan, F. Dewanta, and S. Astuti, "Analisis Perbandingan Algoritma Decision Tree, Random Forest, dan Naïve Bayes untuk Prediksi Banjir di Desa Dayeuhkolot," *TELKA: Jurnal Telekomunikasi, Elektronika, Komputasi dan Kontrol*, vol. 9, no. 1, pp. 52–61, May 2023, doi: 10.15575/telka.v9n1.52-61.

[16] C. Y. M. Baidoo, W. Yaokumah, and E. Owusu, "Estimating Overhead Performance of Supervised Machine Learning Algorithms for Intrusion Detection," *International Journal of Information Technologies and Systems Approach*, vol. 16, no. 1, pp. 1–19, Feb. 2023, doi: 10.4018/IJITSA.316889.

[17] K. Ghosh and V. Bhattacharjee, "Lung cancer prediction: A performance analysis of machine learning classifiers," *International Journal of Statistics and Applied Mathematics*, vol. 9, no. 5, pp. 28–33, Sep. 2024, doi: 10.22271/maths.2024.v9.i5a.1799.

[18] T. Hayashi, T. Shimizu, and Y. Fukami, "Collaborative Problem Solving on a Data Platform Kaggle," *IEICE Tech Report*, vol. 120, no. 362, pp. 37–40, Feb. 2021, doi: 10.48550/arXiv.2107.11929.

[19] A. S. Sitio and F. A. Sianturi, "Penerapan Algoritma Machine Learning dalam Analisis Pola Perilaku Penggunaan Internet," *Dike: Jurnal Ilmu Disiplin*, vol. 2, no. 2, pp. 46–51, Aug. 2024, doi: 10.69688/dike.v2i2.102.

[20] S. M. Malakouti, M. B. Menhaj, and A. A. Suratgar, "The usage of 10-fold cross-validation and grid search to enhance ML methods performance in solar farm power generation prediction," *Clean Eng Technol*, vol. 15, no. 9, pp. 1–7, Jul. 2023, doi: 10.1016/j.clet.2023.100664.

[21] IBM, "What is a Decision Tree | IBM." Accessed: Feb. 17, 2025. [Online]. Available: https://www.ibm.com/think/topics/decision-trees

[22] A. Jananto, S. Sulastri, E. Nur Wahyudi, and S. Sunardi, "Data Induk Mahasiswa sebagai Prediktor Ketepatan Waktu Lulus Menggunakan Algoritma CART Klasifikasi Data Mining," *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, vol. 10, no. 1, pp. 71–78, Feb. 2021, doi: 10.32736/sisfokom.v10i1.991.

[23] IBM, "What Is Random Forest? | IBM." Accessed: Feb. 17, 2025. [Online]. Available: https://www.ibm.com/think/topics/random-forest

[24] R. Rizky Pratama and R. R. Suryono, "Performance Comparison of Naive Bayes, Support Vector Machine and Random Forest Algorithms for Apple Vision Pro Sentiment Analysis," *Jurnal Teknik Informatika (JUTIF)*, vol. 6, no. 1, pp. 31–40, Feb. 2025, doi: 10.52436/1.jutif.2025.6.1.4035.

[25] S. Sathyanarayanan and B. Roopashri Tantri, "Confusion Matrix-Based Performance Evaluation Metrics," *African Journal of Biomedical Research*, vol. 27, no. 4s, pp. 4023–4031, Nov. 2024, doi: 10.53555/AJBR.v27i4S.4345.

Vol. 6, No. 5, October 2025, Page. 3392-3404
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4877

[26] A. Reddy Eppa, "A Comparative Study on Efficiency, Effectiveness, and Practical Applications Using the GRA Method," *Journal of Artificial intelligence and Machine Learning*, vol. 3, no. 1, pp. 1–12, Jan. 2025, doi: 10.55124/jaim.v3i1.260.

[27] P. Kulkarni and V. Lavanya, "A Comparative Study of Machine Learning Algorithms on Structured Data," *JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)*, vol. 11, no. 9, pp. d156–d159, Sep. 2024, [Online]. Available: www.jetir.org

[28] C. Brosch, "Influence of Static Code Analysis on Energy Consumption of Software," in *EnviroInfo 2023: Sustainable Software Engineering and Energy Efficiency*, Garching, Germany: Gesellschaft für Informatik e.V., Oct. 2023, pp. 111–120. doi: 10.18420/env2023-010.

[29] J. A. Firdaus, A. Setia Budi, and E. Setiawan, "Analisis Performa Algoritma Machine Learning Pada Perangkat Embedded ATmega328P," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, vol. 10, no. 2, pp. 245–254, Apr. 2023, doi: 10.25126/jtiik.2023106196.