Building of Informatics, Technology and Science (BITS)
Volume 5, No 2, September 2023 Page: 498−508
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v5i2.4096

Implementation of Hyperparameters to the Ensemble Learning Method for Lung Cancer Classification

Ridlo Yanuar*, Siti Sa'adah, Prasti Eko Yunanto

School of Computing, Telkom University, Bandung, Indonesia
Email: [1,*]ridloyanuarr@student.telkomuniversity.ac.id, [2]sitisaadah@telkomuniversity.ac.id, [3,]gppras@telkomuniversity.ac.id
Correspondence Author Email: ridloyanuarr@student.telkomuniversity.ac.id

Submitted: **16/08/2023**; Accepted: **30/09/2023**; Published**: 30/09/2023**

Abstract−Lung cancer is the most common cause of death in someone who has cancer. This happens because of remembering the importance of lung function as a breathing apparatus and oxygen distribution throughout the body. Early identification of lung cancer is crucial to reduce its mortality rate. Accuracy is crucial since it indicates how accurately the model or system makes the right predictions. High levels of accuracy show that the model can produce trustworthy and accurate findings, essential for making effective decisions based on available data. In this research, ensemble learning approaches, namely bagging and boosting methods, were employed for classifying lung cancer. Hyperparameters, a class of parameters, are crucial to this model's effectiveness. In order to increase the lung cancer classification model's accuracy, a thorough investigation was conducted to identify the best hyperparameter combination. In this study, the dataset used is a medical dataset that contains a history of patients who have been diagnosed with lung cancer or not. The dataset is taken from Kaggle mysarahmadbhat and cancerdatahp from data world. To evaluate the model's accuracy, this study used the confusion matrix method which compares the model's prediction results with the ground truth. the study findings revealed that employing a dataset split ratio of 70:30 produced the best results, with the Random Forest, CatBoost, and XGBoost models achieving an impressive 98% accuracy, 0.98 precision, 0.98 recall, and 0.98 f1-score. but for AdaBoost, the best results were obtained on a dataset with a ratio of 80:20 with an accuracy of 96%, 0.97 precision, 0.96 recall, and 0.96 f1-score.

Keywords: Lung Cancer; Classification; Ensemble Learning; Bagging; Boosting

1. INTRODUCTION

Cancer is the second leading cause of death globally. Lung cancer occurs when there is uncontrolled growth of abnormal cells in the lungs. This abnormal cell growth can damage the normal cells around it and spread to other organs in the body [1]. In the United States, it is estimated that there will be 1,918,030 new cancer cases and 609,360 deaths in 2022 [2]. Smoking is by far the most common risk factor for lung cancer. But not a few people who do not smoke also get lung cancer. Other factors such as passive smoking, genetic heredity, infection with other diseases such as tuberculosis, and asthma are the causes of lung cancer [3], [4]. Given the intricacy of differences in the kinds and phases of the disease's growth, it is becoming more and more necessary to use the categorization system for lung cancer. Without a suitable categorization strategy, the danger of an incorrect diagnosis and an ineffective course of care would persist, having a detrimental influence on the efficiency of therapy and the prognosis of the patient. Modern categorization techniques that make use of top-notch medical data and tools like machine learning will be used to address this problem. The accuracy of the diagnosis can be increased, the therapy can be more specifically suited to the needs of the patient, and the door is open for additional studies that might lead to better solutions. This step is therefore not only necessary but also highly doable to accomplish. The categorization of lung cancer has the potential to significantly improve patient care and scientific advancements in medicine by utilizing current technology and innovations in data analysis.

Several studies using machine learning to classify lung cancer were found in [5] with an accuracy of 90%, precision of 87.82%, recall of 83.71%, and 85.71 of f1 score in the Gradient Boosted Tree algorithm that outperforms Support Vector Machine, C4.5 Decision Tree, Multi-Layer Perceptron, Neural Network, and Naive Bayes. Another study by [6] used Logistic Regression, Decision tree, Naive Bayes, and Support Vector Machine. In this study, two distinct datasets exist. Lung cancer data from UCI Machine Learning is utilized in the first dataset, and lung cancer data from data world is used in the second dataset. On the UCI Machine Learning dataset, Logistic Regression performs best, with an accuracy score of 96.9%, while for the second dataset from data world, SVM performs best, with an accuracy score of 99.2%. There is another study regarding the classification of breast cancer using machine learning by [7], it was found that KNN outperforms NB classifier (96.19%) in accuracy while having a lower error rate (97.51%). Lung cancer risk prediction was also carried out by [8], the suggested model was successful with an AUC of 99.3%, F-Measure, precision, recall, and accuracy of 97.1% when the RotationForest technique and SMOTE were used with 10-fold cross-validation. Other related studies using an ensemble classifier were also carried out by [9] but with CT scan image data. the proposed method obtains an accuracy of 85%, precision of 85%, recall of 89%, and f1-score of 87%. Another prediction of lung cancer with the ensemble method by [10], XGBoost is the best method compared to other ensemble methods and produces a fairly good accuracy of 94.41%. Other machine learning methods such as the Gaussian Naive Bayes used by [11] also show quite good accuracy with a value of 97.5%. In a study [12] that compared a deep neural network (DNN) with six conventional machine learning approaches, the accuracy was 88.58%.

Based on the research above, This research will continue to be conducted using two different datasets. The ensemble learning method used in this research is the bagging and boosting method. For the bagging method, Random Forest will be used as the model. As for the boosting method, Adaptive Boosting (AdaBoost), CatBoost, and XGBoost will be used. To support the classification of lung cancer in this study, datasets from Kaggle mysarahmadbhat and cancerdatahp data world will be used. The problem to be examined in this study is how the results of the ensemble learning method classify the possibility of lung cancer.

2. RESEARCH METHODOLOGY

2.1 Design Implementation

The system flow process in lung cancer classification starts with collecting datasets. Then the data that has been collected will go through the stages of data processing. After processing the data, it will then be continued by dividing the dataset into training data and test data. The training data will later be used to build the Random Forest, AdaBoost, CatBoost, and XGBoost. After the model has been created, model testing will be carried out using training data. Finally, performance calculations will be carried out using the Confusion Matrix to find out the results of each model used. The system flow process can be seen in Figure 1.

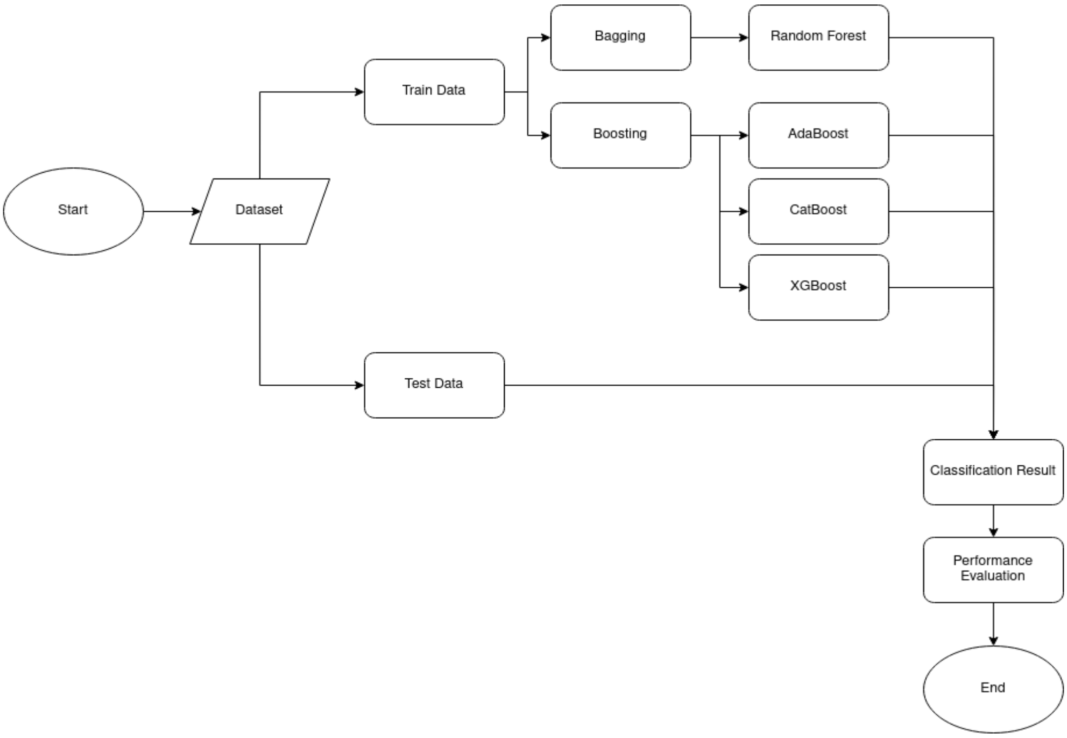

Figure 1. System Flow Classification Lung Cancer

2.2 Ensemble Learning

Ensemble learning is a machine learning technique used to improve models' accuracy and stability by combining several models developed with different methods. This method is based on the concept that several models developed from different data will give better results than just one model.

Ensemble Learning is divided into 3 methods namely, boosting, bagging, and stacking [13], [14]. In addition, Ensemble learning can be used in various types of applications, such as classification, regression, and pattern recognition [15].

2.2.1 Bagging

Bagging is a method based on creating several models developed from random data taken from the original data. Each model is developed from different random data and is predicted independently [15]. bagging process simulation can be seen in Figure 2.

The bagging process is performed by taking random samples from the original data and using those samples to train the model. This process runs iteratively, developing multiple models from different random data sets. The results of each model are then predicted separately, and the results of each model are accumulated to give the final result.

Bagging is used to reduce variability in developed models. This improves model stability and reduces overfitting. Additionally, by using multiple models developed from different random data, bagging can improve the accuracy of the developed models [17].

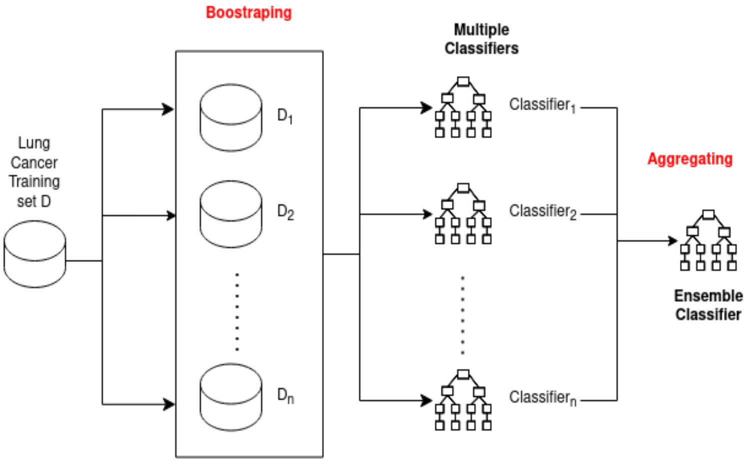

Figure 2. Bagging [16]

Figure 2 illustrates how the bootstrap approach was used to divide the lung cancer training dataset D into n datasets: D_1, D_2, ..., D_n. The numerous datasets (D_1, D_2, ..., D_n) were used to train the chosen classification method, yielding multiple classifiers ($Classfier_1$, $Classifier_2$, ..., $Classifier_n$). The numerous classifiers were then combined to create the ensemble classifier.

The model used in this study is the bagging method or random forest. Random Forest uses a bagging technique that can reduce the effects of overfitting on the model. This technique creates a number of subsets of the data used in the learning process, which are used to generate various decision trees. Each decision tree generated makes a different prediction, and the results of the entire decision tree are voted on to make the final prediction [14], [15], [17].

2.2.2 Boosting

Boosting is one of the ensemble learning methods. This method is based on the concept of making several models which are developed in stages by prioritizing data that was not well predicted by the previous model [18]. The boosting strategy used in this investigation is depicted in Figure 3.

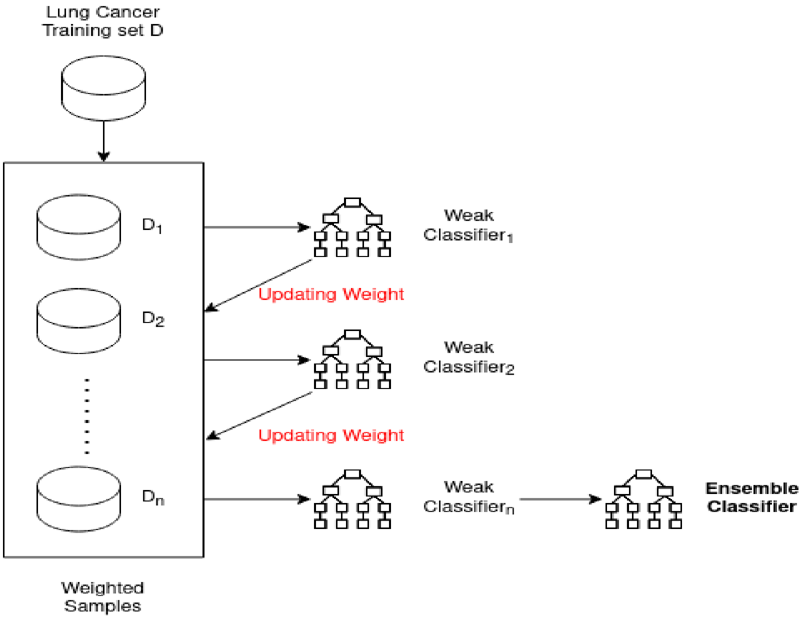

Figure 3. Boosting [16]

To create the D_1 subset, equal-weighted samples were first taken from the diabetic training dataset D. The chosen algorithm was then trained with D_1 to produce the first weak learner. The new subset (D_2) was created by reweighting the samples in the subset in accordance with classification accuracy. Until a strong classifier (ensemble classifier) was created, the procedure of creating weak learners and subsets was repeated.

The boosting process in lung cancer classification begins by taking a sample from available patient data and using the sample to train a model. After the model is trained, patient data for which the model fails to predict well is given more weight in the next model-building process. As this process repeats, multiple models are developed from different patient data. The results of each model are then predicted separately, and the results of each model are accumulated to provide a better final result in classifying lung cancer.

a. AdaBoost

AdaBoost is a learning technique that improves the performance of weak machine learning algorithms (weak learners) by combining multiple weak machine learning algorithms into one strong machine learning algorithm (strong learners) [18], [19]. The AdaBoost method gives weights to each occurrence in the training set at each iteration, it is formulated as follows.

$$D_1(i) = \frac{1}{n}, i = 1,2,\ldots,m \tag{1}$$

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} exp\,(-\alpha_t y_i h_t(x_i)), i = 1,2,\ldots, \tag{2}$$

Where Z_t stands for a normalization factor, $h_t(x)$ is the base classifier, t = 1,..., T is the number of iterations, and t is the weight of the classifier. When determining the final classifier prediction, the weight t indicates how significant the classifier $h_t(x)$ is.

A base classifier is trained using an AdaBoost learning algorithm utilizing a base algorithm, often a decision tree. The second classifier is trained using the modified samples once the sample weights have been updated in light of the classifier's predictions. In order to ensure that the following classifiers pay greater attention to the misclassified samples, the correctly classified examples are given fewer weights and the incorrectly classified samples are given bigger weights [18], [20].

b. CatBoost

CatBoost offers a variety of solutions that enable categorical characteristics. These processes are designed to be used during tree splitting, not preprocessing. CatBoost uses one-hot encoding for features with a limited number of categories [15]. Depending on how frequently a category appears, CatBoost can also convert categorical characteristics to numbers. The categories are changed to their average aim for a more complicated solution. To avoid over-fitting, the average for a sample $x_{\alpha_{i,k}}$ is calculated using the target values of the samples preceding $x_{\alpha_{i,k}}$ in a random permutation of the data set, where $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$. This is illustrated in the following equation:

$$x_{\alpha_{i,k}} = \frac{\sum_{j=1}^{i=1} \; [x_{\alpha_{i,k}} = x_{\alpha_{j,k}}] y_{\alpha_j} + a * P}{\sum_{j=1}^{i=1} \; [x_{\alpha_{i,k}} = x_{\alpha_{j,k}}] y_{\alpha_j} + a} \tag{3}$$

where $[x_{\alpha_{i,k}} = x_{\alpha_{j,k}}]$ takes the value 1 when the condition is met, P is a previous value, and the parameter a weights the prior. In the case of a regression task or a classification job, P is set to the prior probability or the average for the entire set of data.

CatBoost classifier is a further machine learning technique effective in categorical feature prediction. Gradient boosting is implemented in CatBoost, which uses binary decision trees as the basis predictor.

Oblivious trees, also known as decision tables, are used by CatBoost. These trees apply the same splitting criterion for each level of the tree. These trees learn more quickly during the prediction stage because they are balanced, symmetric, and less prone to over-fitting.

CatBoost also handles missing values. Missing values are handled as a distinct category when it comes to categorical features. The occurrences of a certain numerical characteristic lacking values are segregated in their leaf during the splitting phase [21].

c. XGBoost

XGBoost (eXtreme Gradient Boosting) is a machine-learning algorithm developed by Tianqi Chen and Guestrin. Parallel, distributed, out-of-core, and cache-aware computing speeds up the technique more than 10 times quicker than common models used in machine learning and deep learning. Another advantage of this technique is that it is well-tuned and scalable [22]. This invention makes it possible to process billions of instances in distributed or memory-constrained environments. This state-of-the-art use of gradient boosting machines was created to address practical issues where input data sparsity is a typical concern. The algorithm is aware of missing values, an excessive number of zero values in the dataset, and the outcomes of feature engineering approaches that have been used. The ensemble approach involves adding new models repeatedly until doing so no longer significantly improves the performance of the original models.

2.3 Dataset

This research explores specific problems using two different datasets. Each dataset represents different conditions or factors so as to provide a comprehensive picture of the phenomenon under study. The first dataset was taken from the Kaggle website https://www.kaggle.com/datasets/mysarahmadbhat/lung-cancer_which consisted of 284 data samples with 16 attributes, including gender, age, smoking, yellow fingers, anxiety, peer pressure work, chronic illness, fatigue, allergies, wheezing, alcohol consumption, coughing, shortness of breath, difficulty swallowing, chest pain, and lung cancer. The number label "1" can be interpreted as "no" and the number label "2" is interpreted as "yes". An example of a dataset from Kaggle can be seen in Table 1.

Table 1. Lung Cancer Datasets

GENDER	AGE	SMOKING	…	SHORTNESS OF BREATH	CHEST PAIN	LUNG CANCER
M	69	1	…	2	2	YES
M	74	2	…	2	2	YES
F	59	1	…	2	2	NO

Another lung cancer dataset taken from https://data.world/cancerdatahp/lung-cancer-data consists of 25 attributes, including patient id, age, sex, air pollution, alcohol use, dust allergies, occupational hazards, genetic risk, chronic pulmonary disease, balanced diet, obesity, smoking, second-hand smoke, chest pain, coughing up blood, fatigue, weight loss, shortness of breath, wheezing, difficulty swallowing, clubbing fingers, frequent colds, dry cough, snoring, and level. An example of a world dataset can be seen in Table 2.

Table 2. Lung Cancer Data World

PATIENT ID	AGE	AIR POLLUTION	…	DRY COUGH	SNORING	LEVEL
P1	33	2	…	3	4	LOW
P10	17	3	…	7	2	MEDIUM
P100	35	4	…	7	2	HIGH

Based on the examples of the two datasets above, several attributes can be used as material for classification of lung cancer. Referring to the background of the problem of several causes of lung cancer such as genetic factors, an active or passive smoker, age, gender, and several other factors. The dataset above is of a categorical type, where the dataset is suitable for lung cancer classification.

The processing of data at this level includes removing duplicate data, empty data, and other data disruptions. In order to make the modeling process easier, this stage is completed. Different data processing methods were employed to extract information from the two datasets used in this investigation. In the first dataset, 33 of the 284 data share the same information, and there is one column with "yes" and "no" labels indicating the presence of lung cancer. Consequently, the identical information was removed and the labels "yes" and "no" were altered to "1" and "0". in the second dataset there is no the same data and there is no empty data. the label changes in the level column containing “low”, “medium” and “high” to “0”, “1” and “2”. changes to this label are expected so that the label is standardized with other features.

2.4 Hyperparameter Tuning

Hyperparameter tweaking takes a snapshot of a model's performance right now and compares it to earlier snapshots. Hyperparameters must be established before a model begins training in any machine learning method. The performance of the model on a validation set is maximized by fine-tuning the model hyperparameters. A hyperparameter in machine learning is a parameter whose value is set before the learning process is started. On the other hand, model parameters are determined by data training. The weights and coefficients that the algorithm derives from the data are referred to as model parameters [23].

The grid search method is a technique for finding the best classifier parameters that will enable a model to correctly predict certain unlabeled data. Some hyperparameters that cannot be directly learned from the training phase are tuned using the Grid Search approach. Finding the ideal mix of the various hyperparameters in the classification model is a difficult task. The Grid Search technique is a better approach for this [24].

The study uses 4 models, therefore each model will have a different set of parameters. Table 3 contains the random forest model hyperparameter, Table 4 contains the AdaBoost model hyperparameter, Table 5 contains the CatBoost hyperparameter, and Table 6 contains the XGBoost hyperparameter.

Table 3. Grid Parameter for Random Forest

Attribute	Explanation	Parameter Values
n_estimators	Number of Decision Trees in the Random Forest	50, 100, 200
max_depth	Maximum depth of the random forest	40, 60
max_features	Number of features to consider at each split	sqrt, log2

Table 4. Grid Parameter for AdaBoost

Attribute	Explanation	Parameter Values
n_estimators	Number of base estimator	50, 100, 200
learning_rate	learning rate	0.1, 0.5, 1.0

Table 5. Grid Parameter for CatBoost

Attribute	Explanation	Parameter Values
iterations	Number of boosting iterations	50, 100, 200, 300
learning_rate	learning rate for boosting	0.01, 0.05, 0.1

depth	maximum depth of trees	2, 4, 6, 8
l2_leaf_reg	L2 regularization coefficient	1, 3, 5

Table 6. Grid Parameter for XGBoost

Attribute	Explanation	Parameter Values
learning_rate	learning rate for boosting	0.01, 0.05, 0.1
max_depth	maximum depth of trees	2, 3, 4, 5
n_estimators	Number of base estimator	50, 100, 200, 300
gamma	Minimum loss reduction required for further splitting	0, 0.1, 0.2
subsample	ratio for each tree	0.8, 0.9, 1.0
colsample_bytree	subsample ratio of columns for each tree	0.8, 0.9, 1.0

2.4 Performance Evaluation

The confusion matrix is a method used to evaluate the classification results of a model. The confusion matrix displays the number of correct and incorrect predictions made by the model on the test data. The Confusion Matrix is a table with 4 different combinations of predicted and actual values. There are 4 terms in the Confusion Matrix: True Positive, True Negative, False Positive, and False Negative [25] can be seen in Table 7.

Table 7. Confusion Matrix

Prediction	Actual	
	True	False
True	True Positive (TP)	False Positive (FP)
False	False Negative (FN)	True Negative (TN)

In the table above it can be interpreted as True Positive, which means the amount of positive data that is classified correctly. True Negative, which means the number of negative data that are classified correctly. False Positive, which means the number of positive data that is classified as wrong. False Negative, which means the number of negative data that is classified as wrong. Based on the data above, it can be used to measure the performance of a model with Accuracy, Precision, Recall, and F-1 Score.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \tag{4}$$

$$\text{Precision} = \frac{TP}{FP + TP} \tag{5}$$

$$\text{Recall} = \frac{TP}{FN + TP} \tag{6}$$

$$\text{F-1 Score} = 2 * \frac{precision * recall}{precision + recall} \tag{7}$$

Equation (4) Accuracy is the ratio of the number of correct predictions to the amount of test data. Equation (5) Precision is an accuracy metric that shows how often the model issues correct positive predictions. Equation (6) Recall is one of the metrics used to measure model accuracy in classifying. The higher the recall value, the better the model identifies the correct position. Equation (7) F-1 Score is a measure that combines Precision and Recall. F-1 Score is used as a performance measure to compare different models

3. RESULT AND DISCUSSION

This study offers an ensemble learning performance analysis of lung cancer categorization. The accuracy of the classification algorithm used in this work was evaluated using datasets on lung cancer that were made accessible to the public. the results of hyperparameter adjustment in terms of increasing accuracy. Using a confusion matrix, the value of the model accuracy level is assessed.

3.1 Proportion Dataset

The dataset used in this study serves as the basis for creating and assessing the model's performance. A purposeful segmentation of the dataset has been done to guarantee the robustness and dependability of the model. A training set and a test set are the two subsets that must be created in this division. The test set serves as an independent benchmark to judge the model's capacity to generalize to new data, whereas the training set is used to educate the model on the underlying patterns and connections contained within the data.

Three alternative ratios, 90:10, 80:20, and 70:30, have been used to examine the effects of varied data allocations. The percentage of the dataset that is divided between the training and test sets is determined by these ratios. In a 90:10 ratio, training uses 90% of the data while testing uses the remaining 10%. Similar to the 80:20 ratio, which divides the data into 80% for training and 20% for testing, the 70:30 ratio divides the data into 70% for training and 30% for testing.

It was purposefully decided to use these ratios in order to thoroughly assess the model's performance under various circumstances. Testing these ratios allows one to determine whether the model regularly maintains its performance levels and exhibits strong generalization skills. The model's behavior, potential strengths, and capacity to adjust to different training and test datasets may all be better understood using this method.

3.2 Best Hyperparameter

The outcomes of the hyperparameter tuning experiment with three distinct dataset ratios were different for each split of the data. These variations are brought about by the particular traits of each dataset ratio, such as the various training set sizes and levels of data complexity. Statistical uncertainties in the relatively limited datasets and random components in the hyperparameter tuning procedure are additional factors that affect the variation in the outcomes. Additionally, the dataset ratio affects the degree of overfitting or underfitting in the model, which affects how well certain models perform.

The discrepancies in the generated hyperparameters between the two datasets utilized in this investigation are the greatest. The best hyperparameter outcomes are affected by features and the amount of data in the two datasets. **Table 8** shows the findings for the best hyperparameters for the first dataset, while **Table 9** shows the results for the second dataset's best hyperparameters.

Table 8. Best Hyperparameter Result Kaggle Dataset

Ensemble Method	Best Hyperparameter		
	90:10	80:20	70:30
Random Forest	'max_depth': 60, 'max_features': 'sqrt', 'n_estimators': 200	'max_depth': 60, 'max_features': 'sqrt', 'n_estimators': 100	'max_depth': 60, 'max_features': 'sqrt', 'n_estimators': 50
AdaBoost	'learning_rate': 0.5, 'n_estimators': 50	'learning_rate': 0.5, 'n_estimators': 200	'learning_rate': 0.5, 'n_estimators': 100
CatBoost	'depth': 4, 'iterations': 200, 'l2_leaf_reg': 1, 'learning_rate': 0.05	'depth': 4, 'iterations': 100, 'l2_leaf_reg': 1, 'learning_rate': 0.1	'depth': 4, 'iterations': 100, 'l2_leaf_reg': 1, 'learning_rate': 0.1
XGBoost	'colsample_bytree': 0.8, 'gamma': 0, 'learning_rate': 0.01, 'max_depth': 4, 'n_estimators': 200, 'subsample': 0.8	'colsample_bytree': 0.9, 'gamma': 0, 'learning_rate': 0.1, 'max_depth': 3, 'n_estimators': 100, 'subsample': 0.9	'colsample_bytree': 0.9, 'gamma': 0.2, 'learning_rate': 0.1, 'max_depth': 5, 'n_estimators': 100, 'subsample': 0.8

Table 9. Best Hyperparameter Result Data World Dataset

Ensemble Method	Best Hyperparameter		
	90:10	80:20	70:30
Random Forest	'max_depth': 2, 'max_features': 'sqrt', 'n_estimators': 50	'max_depth': 2, 'max_features': 'sqrt', 'n_estimators': 50	'max_depth': 4, 'max_features': 'log2', 'n_estimators': 100
AdaBoost	'learning_rate': 0.1, 'n_estimators': 50	"learning_rate': 0.1, 'n_estimators': 20	'learning_rate': 0.1, 'n_estimators': 40
CatBoost	'depth': 4, 'iterations': 300, 'l2_leaf_reg': 5, 'learning_rate': 0.001	"depth': 2, 'iterations': 20, 'learning_rate': 0.1	"depth': 2, 'iterations': 50, 'l2_leaf_reg': 1, 'learning_rate': 0.1
XGBoost	'colsample_bytree': 0.8, 'gamma': 0, 'learning_rate': 0.01, 'max_depth': 2, 'n_estimators': 20, 'subsample': 1.0	'colsample_bytree': 0.6, 'gamma': 0, 'learning_rate': 0.01, 'max_depth': 2, 'n_estimators': 20, 'subsample': 0.6	'colsample_bytree': 0.6, 'gamma': 0, 'learning_rate': 0.01, 'max_depth': 2, 'n_estimators': 20, 'subsample': 0.6

The differences in the results of the parameters discovered through the parameter tuning process with GridSearchCV at different split dataset train-test ratios can be interpreted as the result of complex interactions between the ensemble algorithm parameters with varying characteristics at each ratio. The parameters that control how the model adapts to and learns patterns from a given dataset are found in ensemble algorithms like Random Forest, Adaboost, CatBoost, and XGBoost. varied comparisons between the data used for training and testing will impact how well these parameters function when utilized with varied train-test split ratios.

The varying parameter results represent the reaction of the algorithm to changes in the distribution of data in each scenario while conducting parameter modifications using the GridSearchCV technique at each train-test split ratio. This explains the pressing need to comprehend how each parameter interacts with the features of the data and how the right parameters may be chosen depending on the nature of the analysis's goal and the characteristics of the relevant dataset. The divergence of the parameters collected at each ratio in the context of lung cancer classification trials demonstrates the algorithm's adaptability to the shifting dynamics of the data distribution, and this influence can eventually have a substantial impact on the model's performance.

3.3 Result

Based on the best hyperparameters generated from GridSearchCV, these hyperparameters will be used to train the Random Forest, AdaBoost, CatBoost, and XGBoost models. The results of the first dataset modeling can be seen in Table 10 and the results of the second dataset modeling can be seen in Table 11.

Table 10. Accuracy Result Kaggle Dataset

Ensemble Method	**Accuracy**		
	90:10	**80:20**	**70:30**
Random Forest	93%	89%	89%
AdaBoost	86%	86%	87%
CatBoost	89%	89%	88%
XGBoost	89%	89%	88%

In this experiment using Random Forest, three variations of the train-test split were used: 90:10 (93% accuracy), 80:20 (89% accuracy), and 70:30 (89% accuracy). The results show that the model is able to recognize lung cancer well at various training data ratios. Although accuracy tends to be higher at larger data ratios, the model is still effective at classifying with less data.

However, when using the Adaboost algorithm, there are interesting findings regarding accuracy. The best results from Adaboost were found at a train-test split ratio of 70:30, with an accuracy of 87%. Whereas in other ratios, namely 90:10 and 80:20, Adaboost's accuracy reaches 86%. This shows that Adaboost has a tendency to perform better at 70:30 ratios, but the difference in accuracy between 70:30 ratios and other ratios remains relatively small.

In addition, interesting results are also seen in the use of the CatBoost and XGBoost algorithms. In this case, the highest accuracy was found in the train-test split ratio of 90:10 and 80:20, both of which reached 89%. At a ratio of 70:30, the accuracy of CatBoost and XGBoost reaches 88%. This shows that these two algorithms tend to provide consistent performance at various ratios, with stable accuracy around 88-89%.

Table 11. Accuracy Result Data World Dataset

Ensemble Method	**Accuracy**		
	90:10	**80:20**	**70:30**
Random Forest	93%	96%	98%
AdaBoost	93%	96%	88%
CatBoost	93%	85%	98%
XGBoost	93%	96%	98%

In this experiment, the Random Forest algorithm was employed to yield measurable accuracy results in the categorization of lung cancer. The accuracy of the train-test splits is 93% for the 90:10 split, 96% for the 80:20 split, and 98% for the 70:30 split. These findings show that the model becomes better at identifying lung cancer trends as the amount of training data grows. The new dataset's properties and data distribution have a significant impact on model performance, highlighting the significance of choosing features that are both useful and representative.

Meanwhile, the Adaboost algorithm. At a ratio of 90:10, Adaboost achieved 93% accuracy, while at a ratio of 80:20 its performance increased to 96%, indicating a significant improvement. However, at a ratio of 70:30, Adaboost's accuracy has decreased to 88%. These findings suggest that Adaboost is sensitive to changes in data distribution, and while its performance varies depending on the ratio, the model still has potential in lung cancer classification.

On the other hand, the experimental results with the CatBoost algorithm. At a 90:10 ratio, CatBoost achieves 93% accuracy, illustrating its good ability to recognize lung cancer patterns when given more training data. However, there is a drop in accuracy at 80:20 ratio, reaching 85%. However, this is offset by excellent performance at 70:30 ratio, where CatBoost achieves a peak accuracy of 98%.

The outcomes of tests using the XGBoost algorithm also provide intriguing results. When given more training data, XGBoost was able to detect lung cancer patterns with an accuracy of 93% at a ratio of 90:10. The accuracy rises to 96% in the 80:20 ratio, demonstrating the performance improvement of XGBoost's capacity to handle pattern complexity with bigger data sets. It is particularly intriguing that XGBoost reaches a high accuracy of 98% at a ratio of 70:30, revealing its exceptional capacity to diagnose lung cancer even with less training data. These results demonstrate the adaptability and durability of XGBoost in handling changes in data distribution while retaining outstanding performance.

3.4 Performance Evaluation

A confusion matrix will be used in this study to evaluate the model. There are four types of results produced in the confusion matrix: True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). The evaluation being tested will focus on the greatest accuracy value in the second dataset.

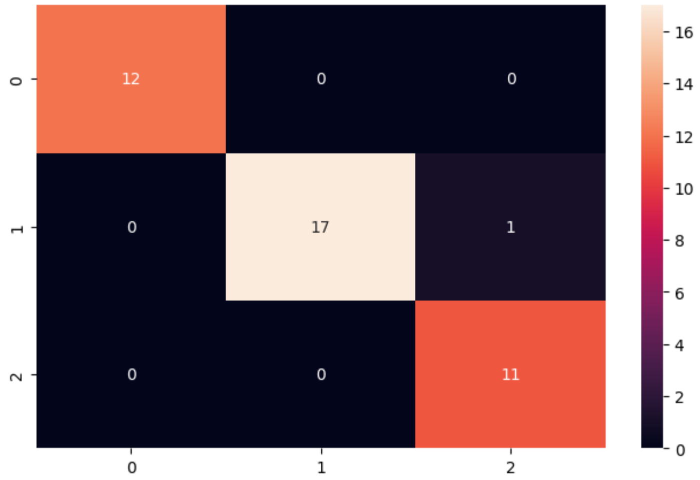

Figure 4. Confusion Matrix Visualization for Random Forest Ratio 70:30

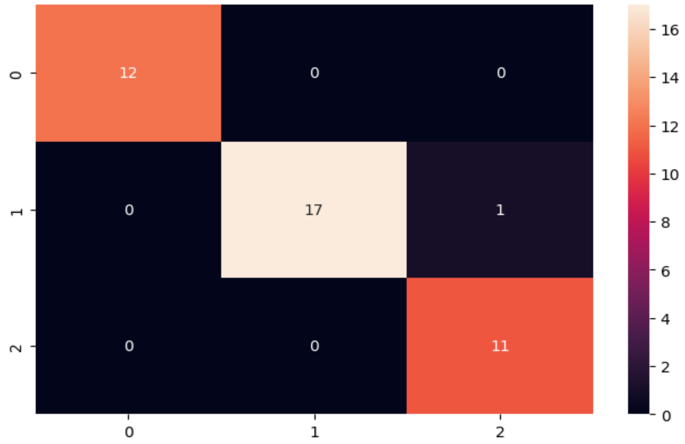

Figure 5. Confusion Matrix Visualization for CatBoost Ratio 70:30

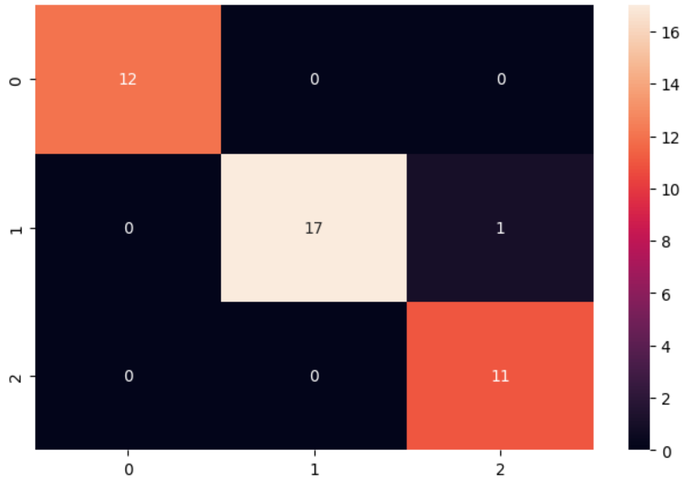

Figure 6. Confusion Matrix Visualization for XGBoost Ratio 70:30

In Figure 4, Figure 5, and Figure 6 high accuracy, precision, recall, and F1 score are all strong performance indicators for this classification model. The precision and recall for each class are 1.0, and the model's accuracy is 98%, showing that it can categorize positive data completely and accurately. The class 1 recall, however, was just 0.944, which suggests that there were some mistakes made while identifying positive data in that class. Class 0 and class 2 both had excellent F1 scores (1.0), but class 1's F1 score was just 0.971, indicating a solid balance between recall and accuracy. This assessment offers crucial information for determining the model's dependability and aids in enhancing performance in those unique situations.

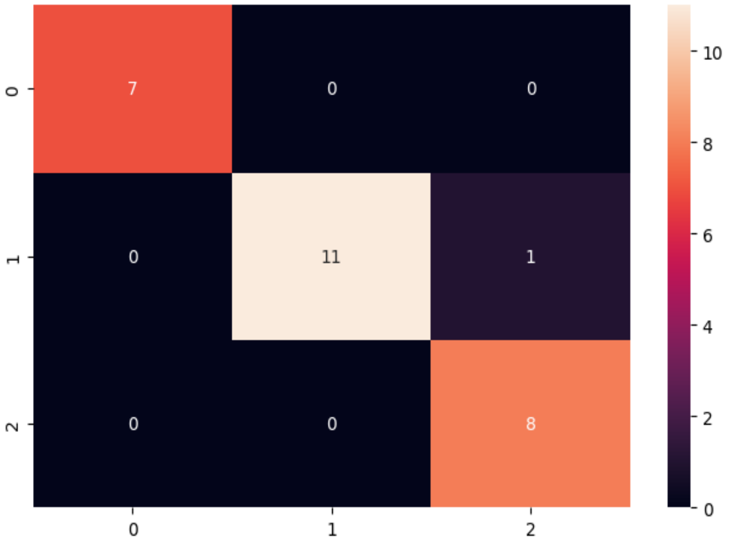

Figure 7. Confusion Matrix Visualization for AdaBoost Ratio 80:20

In Figure 7 High ratings of AdaBoost for accuracy, precision, recall, and F1 scores across all classes demonstrate the classification model's good performance. With recall and precision values of 1.0 for each class, accuracy reaches 96.3%, showing that the model has a remarkable capacity for accurately classifying positive data. A strong F1 score for each class implies that accuracy and recall are balanced well. The results of this evaluation give confidence in the model's dependability and suitability for the classification task it must do, but it is crucial to continue regularly checking the model's performance to make sure the level of accuracy is maintained.

3.5 Comparison Research

In Table 12, several bibliographies are collected which will be used as a reference in this research and to compare the results of previous studies.

Table 12. Comparison of previous similar studies

NO	Title	Method	Best Method	Accuracy
1	A Comparative Study of Lung Cancer Detection Using Machine Learning Algorithm	SVM, Naive Bayes, Logistic Regression, Decision Tree	SVM	99.2%
2	An Evaluation of Machine Learning Classifiers and Ensembles for Early Stage Prediction of Lung Cancer	SVM, Naive Bayes, Decision Tree, MLP, Neural Network, Gradient Boosted Tree	Gradient Boosted Tree	90%
3	Breast Cancer Classification Using Machine Learning	Naive Bayes, KNN	KNN	97.51%

By doing this comparison, it is possible to determine how much the new ensemble model with the suggested hyperparameters is able to overcome the current difficulties and outperform the traditional approach. A major contribution to creating ensemble methods and optimizing hyperparameters in the context of data analysis may be made if the findings of the most recent study are able to compete with or even surpass those of earlier studies. This comparison offers the chance to pinpoint areas for development that require attention and offer guidelines for further study.

4. CONCLUSION

Based on the findings of this study, it can be concluded that the performance of the ensemble algorithm in the context of lung cancer classification is strongly related to the ratio of the division between the training and testing data, as well as the complexity of the characteristics of the dataset used. The Random Forest algorithm shows performance stability, where the peak accuracy is achieved at a ratio of 70:30 with a value of 98%. Meanwhile, Adaboost showed significant reactivity to changes in the data ratio, with the peak accuracy being located at a ratio of 80:20 and reaching 96%. CatBoost highlights its adaptation flexibility by achieving the highest accuracy at 70:30 ratio, also at 98%. Furthermore, XGBoost maintains consistent performance, and peak accuracy is achieved at a ratio of 70:30, reaching 98%. In the context of implementing a lung cancer classification model, the selection of an algorithm must be based on a thorough evaluation of the characteristics of the data and research objectives. A deeper analysis of the variation in data distribution will provide more accurate insights into the model's performance in various scenarios. By considering the advantages as well as limitations of each algorithm and their response to data variations, more appropriate measures can be taken in adapting the optimal classification method to the conditions at hand.

REFERENCES

[1] R. Kumar *et al.*, “Effect of Covid-19 in Management of Lung Cancer Disease: A Review,” *Asian Journal of Pharmaceutical Research and Development*, vol. 10, no. 3, pp. 58–64, Jun. 2022, doi: 10.22270/ajprd.v10i3.1131.

[2] R. L. Siegel, K. D. Miller, H. E. Fuchs, and A. Jemal, “Cancer statistics, 2022,” *CA Cancer J Clin*, vol. 72, no. 1, pp. 7–33, Jan. 2022, doi: 10.3322/caac.21708.

[3] P. M. de Groot, C. C. Wu, B. W. Carter, and R. F. Munden, “The epidemiology of lung cancer,” *Translational Lung Cancer Research*, vol. 7, no. 3. AME Publishing Company, pp. 220–233, Jun. 01, 2018. doi: 10.21037/tlcr.2018.05.06.

[4] L. Corrales, R. Rosell, A. F. Cardona, C. Martín, Z. L. Zatarain-Barrón, and O. Arrieta, “Lung cancer in never smokers: The role of different risk factors other than tobacco smoking,” *Critical Reviews in Oncology/Hematology*, vol. 148. Elsevier Ireland Ltd, Apr. 01, 2020. doi: 10.1016/j.critrevonc.2020.102895.

[5] Muhammad Imran Faisal, Saba Bashir, Zain Sikandar Khan, and Farhan Hassan Khan, *An Evaluation of Machine Learning Classifiers and Ensembles for Early Stage Prediction of Lung Cancer*. IEEE, 2018. doi: 10.1109/ICEEST.2018.8643311.

[6] R. P.R., R. A. S. Nair, and V. G., *A Comparative Study of Lung Cancer Detection using Machine Learning Algorithms*. pp. 1-4, 2019. doi: 10.1109/ICECCT.2019.8869001.

[7] M. Amrane, S. Oukid, I. Gagaoua, and T. Ensarİ, “Breast Cancer Classification Using Machine Learning,” Istanbul, 2018. doi: 10.1109/EBBT.2018.8391453.

[8] E. Dritsas and M. Trigka, “Lung Cancer Risk Prediction with Machine Learning Models,” *Big Data and Cognitive Computing*, vol. 6, no. 4, Dec. 2022, doi: 10.3390/bdcc6040139.

[9] G. A. Shanbhag, K. A. Prabhu, N. V. S. Reddy, and B. A. Rao, “Prediction of Lung Cancer using Ensemble Classifiers,” in *Journal of Physics: Conference Series*, IOP Publishing Ltd, Jan. 2022. doi: 10.1088/1742-6596/2161/1/012007.

[10] M. Mamun, A. Farjana, M. Al Mamun, and M. S. Ahammed, “Lung cancer prediction model using ensemble learning techniques and a systematic review analysis,” in *2022 IEEE World AI IoT Congress, AIIoT 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 187–193. doi: 10.1109/AIIoT54504.2022.9817326.

[11] C. S. Anita, G. Vasukidevi, D. Rajalakshmi, K. Selvi, and T. Ramesh, “Lung cancer prediction model using machine learning techniques,” *Int J Health Sci (Qassim)*, pp. 12533–12539, Jun. 2022, doi: 10.53730/ijhs.v6ns2.8306.

[12] S. Huang, I. Arpaci, M. Al-Emran, S. Kılıçarslan, and M. A. Al-Sharafi, “A comparative analysis of classical machine learning and deep learning techniques for predicting lung cancer survivability,” *Multimed Tools Appl*, 2023, doi: 10.1007/s11042-023-16349-y.

[13] L. Wen and M. Hughes, “Coastal wetland mapping using ensemble learning algorithms: A comparative study of bagging, boosting and stacking techniques,” *Remote Sens (Basel)*, vol. 12, no. 10, May 2020, doi: 10.3390/rs12101683.

[14] S. Bagga, A. Goyal, N. Gupta, and A. Goyal, “Credit Card Fraud Detection using Pipeling and Ensemble Learning,” in *Procedia Computer Science*, Elsevier B.V., 2020, pp. 104–112. doi: 10.1016/j.procs.2020.06.014.

[15] S. González, S. García, J. Del Ser, L. Rokach, and F. Herrera, “A practical tutorial on bagging and boosting based ensembles for machine learning: Algorithms, software tools, performance study, practical perspectives and opportunities,” *Information Fusion*, vol. 64, pp. 205–237, Dec. 2020, doi: 10.1016/j.inffus.2020.07.007.

[16] P. Y. Taser, “Application of Bagging and Boosting Approaches Using Decision Tree-Based Algorithms in Diabetes Risk Prediction,” MDPI AG, Mar. 2021, p. 6. doi: 10.3390/proceedings2021074006.

[17] M. H. D. M. Ribeiro and L. dos Santos Coelho, “Ensemble approach based on bagging, boosting and stacking for short-term prediction in agribusiness time series,” *Applied Soft Computing Journal*, vol. 86, Jan. 2020, doi: 10.1016/j.asoc.2019.105837.

[18] I. D. Mienye and Y. Sun, “A Survey of Ensemble Learning: Concepts, Algorithms, Applications, and Prospects,” *IEEE Access*, vol. 10. Institute of Electrical and Electronics Engineers Inc., pp. 99129–99149, 2022. doi: 10.1109/ACCESS.2022.3207287.

[19] W. Wang and D. Sun, “The improved AdaBoost algorithms for imbalanced data classification,” *Inf Sci (N Y)*, vol. 563, pp. 358–374, Jul. 2021, doi: 10.1016/j.ins.2021.03.042.

[20] Y. N. Lin, T. Y. Hsieh, J. J. Huang, C. Y. Yang, V. R. L. Shen, and H. H. Bui, “Fast Iris localization using Haar-like features and AdaBoost algorithm,” *Multimed Tools Appl*, vol. 79, no. 45–46, pp. 34339–34362, Dec. 2020, doi: 10.1007/s11042-020-08907-5.

[21] A. A. Ibrahim, R. L. Ridwan, M. M. Muhammed, R. O. Abdulaziz, and G. A. Saheed, “Comparison of the CatBoost Classifier with other Machine Learning Methods,” vol. 11, no. 11, 2020. doi: 10.14569/IJACSA.2020.0111190.

[22] A. Paleczek, D. Grochala, and A. Rydosz, “Artificial breath classification using xgboost algorithm for diabetes detection,” *Sensors*, vol. 21, no. 12, Jun. 2021, doi: 10.3390/s21124187.

[23] E. Elgeldawi, A. Sayed, A. R. Galal, and A. M. Zaki, “Hyperparameter tuning for machine learning algorithms used for arabic sentiment analysis,” *Informatics*, vol. 8, no. 4, Dec. 2021, doi: 10.3390/informatics8040079.

[24] S. George and B. Sumathi, “Grid Search Tuning of Hyperparameters in Random Forest Classifier for Customer Feedback Sentiment Prediction,” vol. 11, no. 9, 2020. doi: 10.14569/IJACSA.2020.0110920.

[25] M. Hasnain, M. F. Pasha, I. Ghani, M. Imran, M. Y. Alzahrani, and R. Budiarto, “Evaluating Trust Prediction and Confusion Matrix Measures for Web Services Ranking,” *IEEE Access*, vol. 8, pp. 90847–90861, 2020, doi: 10.1109/ACCESS.2020.2994222.