Detection of lung cancer mutation based on clinical and morphological features using adaptive boosting method

Lailil Muflikhah*, Amira G. Nurfansepta, Edy Santoso, Agus Wahyu Widodo

Department of Informatics Engineering, Faculty of Computer Science, Brawijaya University, Malang, East Java, Indonesia

*Corresponding author E-mail: lailil@ub.ac.id

(Received 20 November 2024; Final version received 19 April 2025; Accepted 17 June 2025)

Abstract

Lung cancer is a leading cause of cancer-related mortality worldwide, and accurate detection of epidermal growth factor receptor mutations is essential for personalized treatment. However, non-invasive identification of these mutations remains challenging due to the complexity of clinical and morphological patterns. This study develops an adaptive boosting (AdaBoost)-based machine learning model for detecting lung cancer mutations using clinical and morphological data. The dataset consists of clinical and morphological attributes from 80 patients, which processed through comprehensive preprocessing steps, including imputation, outlier removal, and feature selection. One-hot encoding increased the feature count beyond the original 28, and analysis of variance was employed to retain the most relevant 33 features. AdaBoost was trained with optimized hyperparameters, including learning rate and the number of estimators, which were tuned using grid search to ensure robustness. The model's performance was evaluated using an 80/20 train-test split and k-fold cross-validation to assess generalization capability. Experimental results demonstrated that AdaBoost outperformed other models, achieving an accuracy of 83% and an area under the curve of 0.90 after feature selection. The model maintained superior cross-validation scores compared to Naive Bayes, decision tree, K-nearest neighbors, and support vector machine, reinforcing its reliability in mutation detection. The study highlights the significance of preprocessing steps in improving classification performance and suggests that AdaBoost can serve as an effective, non-invasive tool for assisting clinical decision-making in lung cancer mutation detection.

Keywords: Adaptive Boosting, Analysis of Variance, Lung Cancer, Machine Learning, Mutation

1. Introduction

Lung cancer is a leading cause of cancer-related mortality, with early and accurate detection of genetic mutations playing a crucial role in optimizing treatment strategies (Rakesh & Baskar, 2024). Among these mutations, epidermal growth factor receptor (EGFR) mutations are particularly significant for targeted therapies (Wang et al., 2019). Traditional detection methods rely on invasive procedures such as tissue biopsies, which pose risks to patients and may not always be feasible (Kanan et al., 2024).

Advancements in machine learning (ML) offer promising non-invasive alternatives for mutation detection using clinical and morphological data (Yu et al., 2019). Various ML models, including support vector machines (SVM), decision trees, and K-nearest neighbors (KNN), have been applied in cancer diagnosis, but their performance often depends heavily on extensive feature engineering and preprocessing (Jain et al., 2024). Boosting algorithms, particularly adaptive boosting (AdaBoost), have demonstrated superior performance by combining multiple weak learners into a robust predictive model (Bushara et al., 2023).

This study leverages the AdaBoost algorithm to enhance the accuracy and sensitivity of EGFR mutation detection based on clinical and morphological features. Through rigorous preprocessing, including outlier removal, feature encoding, and analysis of variance (ANOVA)-based feature selection, we optimize the dataset for improved classification performance. The effectiveness of AdaBoost is evaluated in comparison to other ML models, emphasizing its potential as a reliable, non-invasive alternative for assisting clinical decision-making in lung cancer mutation detection.

2. Related Work

The application of ML in lung cancer detection has been extensively explored, with various studies highlighting the effectiveness of traditional ML models in identifying cancerous mutations. Previous research has applied ML approaches such as SVM, decision trees, KNN, and ensemble techniques such as Random Forest in cancer classification (Maurya et al., 2024). These methods have demonstrated success in distinguishing cancer subtypes and predicting patient outcomes, but often require extensive feature engineering to enhance model performance (Li, 2023).

Boosting methods, particularly AdaBoost, have been increasingly recognized for their ability to enhance classification accuracy in medical applications by combining multiple weak learners into a strong predictive model (Gautam et al., 2024). Unlike deep learning techniques, which require large datasets and substantial computational resources, AdaBoost provides a more interpretable and computationally efficient approach, making it suitable for clinical applications with limited data availability (Jain et al., 2024).

Several studies have applied ML techniques in lung cancer detection using clinical and morphological data. For instance, Kwon et al. (2023) demonstrated that integrating multiple blood markers and clinico-pathological features significantly improved classification performance (Kwon et al., 2023). Similarly, Wang et al. (2019) investigated the use of imaging and clinical attributes for mutation detection, showing that feature selection played a crucial role in model optimization (Wang et al., 2019). However, most prior research has focused on deep learning-based models, such as convolutional neural networks (CNNs), for lung cancer diagnosis (Le et al., 2021). While CNNs have demonstrated high accuracy in radiomics-based studies, their black-box nature and high computational requirements limit their practical use in clinical settings (Kanan et al., 2024, p. 20).

In contrast, this study focuses on leveraging AdaBoost to detect EGFR mutations based on clinical and morphological features. The rationale for using AdaBoost over deep learning approaches lies in its ability to handle smaller datasets while maintaining high classification performance. In addition, AdaBoost enables easier interpretation of feature importance, which is crucial in clinical decision-making (Sachdeva et al., 2024). By incorporating feature selection techniques such as ANOVA, this study aims to further enhance model robustness and accuracy, addressing gaps in existing literature that often overlook the impact of feature selection on ML performance.

3. Research Methods

This research was designed as a predictive study aimed at detecting lung cancer mutations, specifically EGFR mutations, using ML on clinical and morphological data. The study involved several phases, as shown in Fig. 1, including data acquisition, preprocessing, feature selection, model development, and performance evaluation. By employing the AdaBoost algorithm, this study sought to improve mutation detection accuracy, leveraging its capacity to enhance predictive accuracy through boosting weak learners into a stronger predictive model.

This study employs the AdaBoost algorithm, an ensemble learning method that iteratively combines weak classifiers to create a more robust predictive model. As illustrated in Fig. 1, AdaBoost operates by assigning weights to training instances and adjusting them iteratively based on model performance. A properly referenced workflow diagram depicting this process is included to enhance comprehension.

3.1. Data Sets

The data set used in this study consists of clinical and morphological data collected from 80 lung cancer patients, initially comprising 28 features. To ensure data quality, several preprocessing steps were performed. Missing values in numerical features were addressed using KNN imputation, whereas categorical features underwent mode imputation to maintain data completeness. Outlier detection and removal were conducted using the interquartile range (IQR) method, minimizing the impact of extreme values that could potentially bias the model. In addition, categorical variables such as lobe location and emphysema type were transformed using one-hot encoding, which increased the total number of features beyond the initial 28. Feature selection was then performed using the ANOVA method, refining the feature set to 33 relevant predictors.

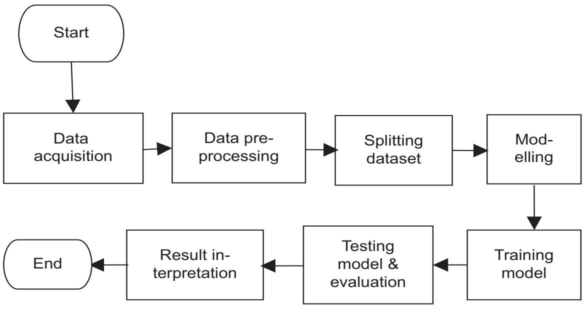

Fig. 1. General proposed method

The boxplots in Fig. 2 depict the distribution of various features related to diabetic nephropathy, such as age, calcification, tumor location, and metastasis. "Age" is well-distributed with a median in the mid-60s, whereas features such as "dimension" and "density" show minimal variability, suggesting limited diagnostic relevance. "Calcification" and "tumor location" exhibit greater variability, indicating potential significance in disease progression. Metastasis-related features, such as "liver" and "bone metastasis," show rare occurrences with occasional outliers. Overall, the plots highlight patterns and anomalies that may aid in understanding the variability of clinical features associated with diabetic nephropathy.

3.2. Preprocessing

Data preprocessing is a crucial step in the data analysis workflow, aimed at preparing raw data into a cleaner and more usable form for analysis models or ML. This process includes various techniques, such as data cleaning to remove missing values, handling outliers, data transformation – which may involve converting data types or encoding categorical variables into numeric formats – and feature selection. The main goal of preprocessing is to enhance data quality so that the model used can produce more accurate results, while also reducing the risk of bias and errors in further analysis (Benhar et al., 2020).

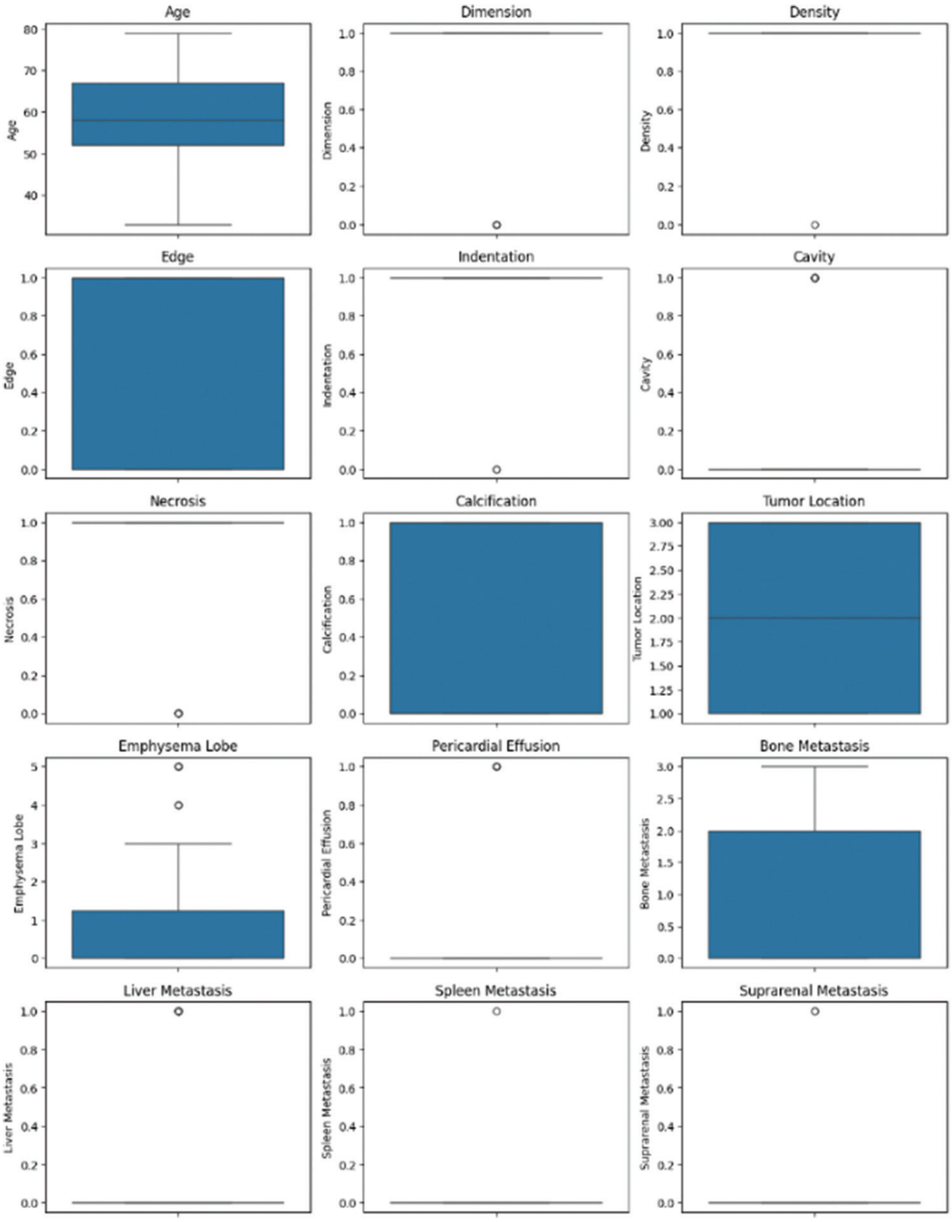

Fig. 2. Boxplot of the distribution of several features in data sets

3.2.1. Data Cleansing

The data were first checked for duplicates, and no duplicate entries were found. Next, an assessment of missing values revealed that several features contained missing data, requiring either removal or imputation to ensure data quality. To streamline the dataset for modeling, the features “no” and “test number” were removed, as they were not relevant to the modeling process. In addition, patient records with missing values across nearly all features were excluded, resulting in the removal of four patient records due to the high proportion of missing data.

3.2.2. Outlier Detection

Outlier detection using the IQR method, also known as the Tukey method, is a highly effective technique for identifying and removing extreme values from a dataset. This method is based on the IQR measurement, which is the range between the first quartile (*Q1*) and the third quartile (*Q3*), encompassing the central portion of the data distribution (Berger & Kiefer, 2021). This process helps produce more accurate models and analysis results, especially in situations where data distributions are non-normal or highly variable. The IQR formula, along with the Tukey method for outlier identification, is shown in Eq. (1), Eq. (2), and Eq. (3).

$$IQR = Q_3 - Q_1 \tag{1}$$

$$Outlier > Q_3 + 1.5 \times IQR \tag{2}$$

$$Outlier > Q_1 + 1.5 \times IQR \tag{3}$$

3.2.3. One-Hot Encoding

Performing one-hot encoding for several features, such as lobe location, emphysema type, emphysema location, lymphadenopathy, pulmonary nodule, and pleural effusion, can yield new features that are more relevant and have a higher correlation with EGFR mutation. Results are improved when one-hot encoding is applied. To handle categorical variables, we applied one-hot encoding, which converts each categorical feature into multiple binary variables, ensuring compatibility with ML models. In this study, categorical features such as lobe location and emphysema type were transformed using this method.

3.2.4. Feature Selection

Feature selection plays a crucial role in improving model performance by eliminating irrelevant or redundant features. In this study, categorical variables such as lobe location and emphysema type were transformed using one-hot encoding, which expanded the feature space. To reduce dimensionality and retain only the most relevant predictors, the ANOVA method was applied. ANOVA evaluates the statistical significance of each feature in relation to the target variable, ensuring that only features with strong discriminative power are selected. As a result, the feature set was reduced to 33 features, which were subsequently used for model training and evaluation.

Feature selection using the ANOVA method is a technique that identifies features that have a significant impact on the target variable in a dataset. ANOVA is applied to compare the means of groups generated by different features to determine if these differences are substantial enough to influence the target variable (Nasiri & Alavi, 2022). Features showing significant differences are considered important and are retained in the model, whereas non-significant features may be removed to simplify the model and reduce the risk of overfitting. This method is particularly useful in regression or classification analysis, where selecting the right features can significantly enhance model accuracy and computational efficiency. The ANOVA formula involves the total sum of squares (*SST*) and sum of squares between (*SSB*) as shown in Eq. (4) and Eq. (5).

$$SST = \sum_{i=1}^{N} \left(X_i - \bar{X} \right)^2 \tag{4}$$

Remark:
X_i: i^{th} data point
$\bar{X}$: Means of all data

N: Total number of observations for all groups

$$SSB = \sum_{j=1}^{k} n_j \left(\bar{X}_j - \bar{X} \right)^2 \tag{5}$$

Remark:
n_j: Number of observations (data points) in group j
k: Number of groups
$n_1, n_2, \ldots n_k$: Sample size of each group

Furthermore, the *F*-statistics in ANOVA is a measure used to compare variances, and it is calculated based on the SSB and the Within-Group Sum of Squares (SSW) as shown in Eq. (8). Specifically, the *F*-value is obtained by dividing the mean square between groups (MSB) by the mean square within groups (MSW) as shown in Eq. (6) and Eq. (7).

$$MSB = \frac{SSB}{k-1} \tag{6}$$

$$MSW = \frac{SSW}{N-k} \tag{7}$$

$$F = \frac{MSB}{MSW} \tag{8}$$

3.3. Adaptive Boosting Algorithm

ML methods typically assume that data are well-distributed, though in practice, this ideal condition is rarely met. In real-world classification tasks, class imbalance is a common challenge, where certain classes have significantly fewer samples than others. This imbalance can negatively impact model performance, as standard classification methods may be biased toward the majority class, leading to poor generalization on minority classes. Unlike traditional ML, which builds a single model from a dataset that ensembles learning methods and combines multiple models to enhance predictive performance (Zhou, 2012).

Ensemble methods aim to reduce model errors and improve accuracy by leveraging the strengths of multiple classifiers. Several techniques are used in ensemble learning: Stacking integrates outputs from different models, where a meta-model predicts the final outcomes based on the outputs of base models, while bagging (e.g., random forest) improves stability by training models on bootstrapped subsets of data. Boosting is a powerful ensemble learning technique designed to improve prediction accuracy by sequentially combining multiple weak learners into a single strong learner (Rincy & Gupta, 2020).

The boosting process works iteratively, where each new model is trained to focus on the mistakes made by its predecessors. Specifically, misclassified instances are assigned to higher weights, making them more influential in training subsequent models. This iterative process continues, with each new weak learner refining the overall prediction by correcting previous errors (González et al., 2020). Finally, the predictions of all models are aggregated – using weighted voting for classification or weighted summation for regression – to generate the final output. A workflow diagram of the boosting process is shown in Fig. 3, illustrating how multiple models contribute to building a more robust predictor.

This study leverages the AdaBoost algorithm, a robust ensemble method, particularly effective in handling class imbalance and enhancing weak classifiers. AdaBoost assigns higher weights to misclassified instances, ensuring that hard-to-classify cases receive more focus in subsequent iterations. The AdaBoost algorithm follows these key steps, and a detailed illustration of the AdaBoost process is provided in Fig. 3, depicting how misclassified samples influence model training at each iteration.

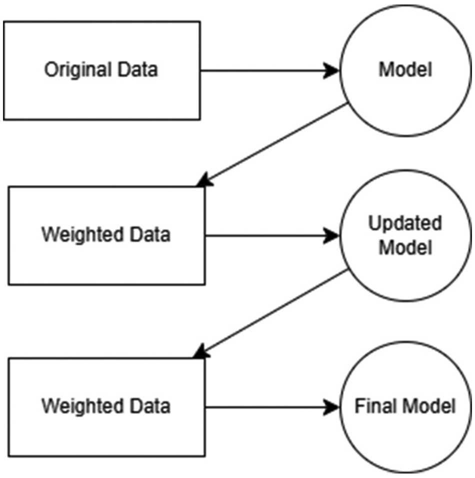

Fig. 3. The stages of the AdaBoost process

1. Initializing sample weights: Initially, all training samples x_i are assigned equal weights ω_i, ensuring a uniform distribution across the dataset. This step is mathematically represented in Eq. (9).

$$\omega_i = \frac{1}{N} \tag{9}$$

2. Training weak learners: At each boosting iteration *t*, a weak classifier $h_t(x)$ is trained using the weighted dataset. The classification error $\in_t$, which quantifies the misclassification rate of the weak learner, is computed as defined in Eq. (10).

$$\in_t = \sum_{i=1}^{N} \omega_i . I\left(h_t\left(x_i\right) \neq y_i\right) \tag{10}$$

3. Updating classifier weight: The importance of each weak classifier is determined based on its classification error. A higher weight is assigned to more accurate classifiers, as shown in Eq. (11), where α_t is calculated as a function of $\in_t$.

$$\alpha_t = \frac{1}{2} \ln\left(\frac{1-\in_t}{\in_t}\right) \tag{11}$$

2. Updating sample weights: The weights of misclassified samples are increased to ensure they receive more attention in subsequent iterations. The new sample weight distribution is determined using Eq. (12), ensuring that harder-to-classify instances influence future classifiers more significantly.

$$\omega_i \leftarrow \omega_i . exp\left(-\alpha . y_i h_i\left(x_i\right)\right) \tag{12}$$

3. Normalization of weights: To maintain a valid probability distribution, the sample weights are normalized, as represented in Eq. (13).

$$\omega_i \leftarrow \frac{\omega_i}{\sum_{j=1}^{N} \omega_i} \tag{13}$$

4. Final model (strong learner): The final strong classifier H_x is obtained by combining all weak classifiers, weighed according to their performance, as stated in Eq. (14).

$$H_x = sign\left(\sum_{t=1}^{T}\alpha_t . h_t(x)\right) \quad (14)$$

4. Results and Discussion

4.1. Experimental Result

To evaluate model performance, five ML models, KNN, Naive Bayes, SVM, decision tree, and AdaBoost, were tested under different preprocessing scenarios. The dataset was initially split into 80% training and 20% testing, ensuring that the models were trained on a substantial portion of the data while preserving a separate test set for final evaluation.

To optimize hyperparameters and assess model generalization, *k*-fold cross-validation (with *k* = 5 or 10, as specified per experiment) was applied exclusively to the training set. This procedure ensured that model selection was based on performance across multiple validation splits, preventing overfitting to a single subset. After cross-validation, the best-performing hyperparameters were used to train the final model, which was then evaluated on the unseen test set to provide an independent measure of performance. Hyperparameter tuning was conducted using a grid search approach, systematically exploring multiple parameter values for each model. For KNN, the number of neighbors was tested with values (3, 5, 7, and 9). Naive Bayes was implemented using the Gaussian Naive Bayes approach, assuming a normal distribution for continuous features. For SVM, the radial basis function kernel was applied, with the penalty parameter tested over the range (0.1, 1, 10, and 100). Decision tree models were optimized by varying the maximum depth between (5, 10, 15, and 20), and the minimum number of samples per leaf was tested at (1, 5, and 10). For AdaBoost, the number of estimators was set to (50, 100, and 200), whereas the learning rate was tuned within the range (0.01, 0.1, and 1).

The final hyperparameter configurations were determined based on the highest cross-validation accuracy. Once optimized, the models were evaluated on the test set, and their performance was measured using key classification metrics: accuracy, precision, recall, F1 score, and AUC-receiver operating characteristics (ROC). The results, presented in Tables 1-3, highlight the impact of different preprocessing strategies on model performance. These findings demonstrate that AdaBoost consistently outperformed other models across various preprocessing scenarios, specifically after applying ANOVA-based feature selection.

The results in Table 1 highlight the performance of various models for lung cancer mutation detection without feature selection or outlier removal. KNN performs best with an accuracy of 68.8% and a high AUC-ROC of 0.746. It shows stable performance even without tuning. Naive Bayes improves significantly after tuning, matching KNN's performance and achieving high precision (81.8%). SVM and AdaBoost initially performed poorly but improved with tuning, with SVM reaching 68.8% accuracy and AdaBoost improving its F1-score to 0.613. The decision tree shows moderate performance with minimal gains from tuning. Overall, the results indicate that preprocessing steps such as feature selection and outlier removal are crucial for improving model performance.

The results in Table 2 demonstrate the significant improvement in performance for lung cancer mutation detection when outliers are removed, even without feature selection. AdaBoost achieves perfect scores across all metrics, indicating exceptional model performance with complete alignment between

Table 1. Comparison of performance results (without feature selection and outlier removal)

Model	Accuracy	F1-score	Precision	Recall	AUC
KNN	0.688	0.689	0.695	0.688	0.746
KNN (tuning)	0.688	0.689	0.695	0.688	0.698
Naive bayes	0.625	0.588	0.798	0.625	0.698
Naive Bayes (Tuning)	0.688	0.689	0.818	0.688	0.695
SVM	0.563	0.405	0.316	0.563	0.240
SVM (tuning)	0.688	0.684	0.731	0.688	0.238
Decision tree	0.563	0.557	0.556	0.563	0.546
Decision tree (tuning)	0.563	0.564	0.570	0.563	0.516
AdaBoost	0.500	0.450	0.577	0.500	0.399
AdaBoost (tuning)	0.625	0.613	0.689	0.625	0.508

Abbreviations: AdaBoost: Adaptive boosting; AUC: Area under the curve; KNN: K-nearest neighbors; SVM: Support vector machine.

precision, recall, and AUC. KNN shows remarkable improvement with tuning, reaching 87.5% accuracy and strong F1, precision, and recall scores. Naive Bayes also performs consistently well, achieving high accuracy (87.5%) and an excellent AUC of 0.933, both with and without tuning. The decision tree delivers strong performance, with tuned and untuned versions achieving 87.5% accuracy and a high AUC of 0.900. SVM exhibits moderate results, maintaining consistent scores before and after tuning. Overall, the removal of outliers significantly enhances the models' robustness and effectiveness, particularly boosting tuned algorithms such as AdaBoost and KNN.

The results in Table 3 demonstrate the effect of combining feature selection and outlier removal on lung cancer mutation detection. AdaBoost and its tuned version maintain perfect scores across all metrics, achieving 100% accuracy, precision, recall, F1-score, and AUC, showcasing its effectiveness in handling the refined dataset. KNN and Naive Bayes perform consistently well, with tuning enhancing their performance to 87.5% accuracy and achieving an AUC of 0.933. SVM delivers moderate results, maintaining a balanced performance with 75% across all metrics, showing that feature selection and outlier removal have a limited impact on this algorithm. The decision tree shows significant improvement with tuning, increasing accuracy and recall to 87.5% and achieving an AUC of 0.906. Overall, combining feature selection with outlier removal enhances model robustness, especially for tuned models such as AdaBoost, KNN, and decision trees, leading to better classification performance and improved detection capability.

4.2. Discussion

The results of this study demonstrate the effectiveness of AdaBoost in detecting EGFR mutations using clinical and morphological features. Compared to other models, AdaBoost consistently achieved the highest classification performance across different preprocessing scenarios. These findings align with previous research by Kwon et al. (2023), who reported improved cancer classification by integrating

Table 2. Comparison of performance results (without feature selection but with outlier removal)

Model	Accuracy	F1-score	Precision	Recall	AUC
KNN	0.625	0.631	0.656	0.625	0.500
KNN (tuning)	0.875	0.868	0.896	0.875	0.500
Naive Bayes	0.875	0.868	0.896	0.875	0.933
Naive Bayes (tuning)	0.875	0.868	0.896	0.875	0.933
SVM	0.750	0.750	0.750	0.750	0.667
SVM (tuning)	0.750	0.750	0.750	0.750	0.667
Decision tree	0.875	0.877	0.906	0.875	0.900
Decision tree (tuning)	0.875	0.877	0.906	0.875	0.900
AdaBoost	1.000	1.000	1.000	1.000	1.000
AdaBoost (tuning)	1.000	1.000	1.000	1.000	1.000

Abbreviations: AdaBoost: Adaptive boosting; AUC: Area under the curve; KNN: K-nearest neighbors; SVM: Support vector machine.

Table 3. Comparison of performance results (with both feature selection and outlier removal)

Model	Accuracy	F1-score	Precision	Recall	AUC
KNN	0.625	0.631	0.656	0.625	0.667
KNN (tuning)	0.875	0.868	0.896	0.875	0.933
Naive Bayes	0.875	0.868	0.896	0.875	0.933
Naive Bayes (tuning)	0.875	0.868	0.896	0.875	0.933
SVM	0.750	0.750	0.750	0.750	0.750
SVM (tuning)	0.750	0.750	0.750	0.750	0.750
Decision tree	0.750	0.750	0.850	0.750	0.800
Decision tree (tuning)	0.875	0.877	0.906	0.875	0.906
AdaBoost	1.000	1.000	1.000	1.000	1.000
AdaBoost (tuning)	1.000	1.000	1.000	1.000	1.000

Abbreviations: AdaBoost: Adaptive boosting; AUC: Area under the curve; KNN: K-nearest neighbors; SVM: Support vector machine.

multiple clinical biomarkers. Similarly, Wang et al. (2019) emphasized the importance of feature selection, showing that carefully curated features significantly enhance predictive accuracy (Wang et al., 2019).

The role of preprocessing was particularly notable in this study. The removal of outliers and the application of ANOVA-based feature selection resulted in improved model performance, which highlighted the impact of feature selection in medical ML applications (Jain et al., 2024). Moreover, the superiority of ensemble methods in handling complex, heterogeneous data has been previously established by Gautam et al. (2024), supporting the efficacy of AdaBoost in this study (Gautam et al., 2024).

While deep learning approaches such as CNNs have been extensively used in lung cancer detection. Their application is often constrained by high computational demands and the need for large datasets (Le et al., 2021). The findings of this study further validate the viability of traditional ML models, particularly ensemble methods such as AdaBoost, as practical alternatives in scenarios where data availability and interpretability are crucial factors.

Despite these promising results, certain limitations remain. The dataset used in this study was relatively small, which may affect the model's generalizability. Future studies should explore external validation on larger datasets and incorporate additional features, such as genetic and radiomic data, to enhance model robustness. In addition, further comparisons with deep learning models could provide deeper insights into the trade-offs between interpretability and predictive performance.

5. Conclusion

This study demonstrated the effectiveness of the AdaBoost algorithm in detecting EGFR mutations in lung cancer patients using clinical and morphological features. Compared to other ML models such as SVM, decision tree, and KNN, AdaBoost achieved superior classification performance, emphasizing its potential as a non-invasive diagnostic tool. The preprocessing steps, including outlier removal, feature encoding, and ANOVA-based feature selection, played a crucial role in optimizing the dataset and improving model accuracy. Furthermore, hyperparameter tuning using grid search ensured optimal model performance, highlighting the importance of systematic parameter selection in ML-based medical applications.

6. Future Work

Despite these promising findings, this study has some limitations. The dataset was relatively small, consisting of 80 patient records, which may impact the generalizability of the results. Therefore, to address these limitations and build upon the findings of this study, several future research directions are proposed. First, the model should be validated on larger and more diverse datasets to assess its robustness and generalizability. Second, incorporating additional features, such as genetic markers and radiomic data, may enhance classification performance and provide a more comprehensive assessment of mutation status.

Acknowledgment

This research is supported by the Faculty of Computer Science, Brawijaya University, under DIPA Research Grant with contract number: 4559/UN10.Fl 5/PN/2024.

References

Benhar, H., Idri, A., & Fernández-Alemán, J.L. (2020). Data preprocessing for heart disease classification: A systematic literature review. *Computer Methods and Programs in Biomedicine*, 195, 105635. https://doi.org/10.1016/j.cmpb.2020.105635

Berger, A., & Kiefer, M. (2021). Comparison of different response time outlier exclusion methods: A simulation study. *Frontiers in Psychology*, 12, 675558. https://doi.org/10.3389/fpsyg.2021.675558

Bushara, A.R., Vinod Kumar, R.S., & Kumar, S.S. (2023). An ensemble method for the detection and classification of lung cancer using computed tomography images utilizing a capsule network with visual geometry group. *Biomedical Signal Processing and Control*, 85, 104930. https://doi.org/10.1016/j.bspc.2023.104930

Gautam, N., Basu, A., & Sarkar, R. (2024). Lung cancer detection from thoracic CT scans using an ensemble of deep learning models. *Neural Computing and Applications*, 36(5), 2459–2477. https://doi.org/10.1007/s00521-023-09130-7

González, S., García, S., Del Ser, J., Rokach, L., & Herrera, F. (2020). A practical tutorial on bagging and boosting based ensembles for machine learning: Algorithms, software tools, performance study, practical perspectives and opportunities. *Information Fusion*, 64, 205–237. https://doi.org/10.1016/j.inffus.2020.07.007

Jain, R., Singh, P., Abdelkader, M., & Boulila, W. (2024). Efficient lung cancer detection using computational intelligence and ensemble learning. *PLOS ONE*, 19(9), e0310882. https://doi.org/10.1371/journal.pone.0310882

Kanan, M., Alharbi, H., Alotaibi, N., Almasuood, L., Aljoaid, S., Alharbi, T., et al. (2024). AI-driven models for diagnosing and predicting outcomes in lung cancer: A systematic review and meta-analysis. *Cancers (Basel)*, 16(3), 674. https://doi.org/10.3390/cancers16030674

Kwon, H.J., Park, U.H., Goh, C.J., Park, D., Lim, Y.G., Lee, I.K., et al. (2023). Enhancing lung cancer classification through integration of liquid biopsy multi-omics data with machine learning techniques. *Cancers (Basel)*, 15(18), 4556. https://doi.org/10.3390/cancers15184556

Le, N.Q.K., Kha, Q.H., Nguyen, V.H., Chen, Y.C., Cheng, S.J., & Chen, C.Y. (2021). Machine learning-based radiomics signatures for EGFR and KRAS mutations prediction in non-small-cell lung cancer. *International Journal of Molecular Sciences*, 22(17), 9254. https://doi.org/10.3390/ijms22179254

Li, X. (2023). Lung cancer risk prediction and feature importance analysis with machine learning algorithm. *Applied and Computational Engineering*, 19, 205–210. https://doi.org/10.54254/2755-2721/19/20231034

Maurya, S.P., Sisodia, P.S., Mishra, R., & Singh, D.P. (2024). Performance of machine learning algorithms for lung cancer prediction: A comparative approach. *Scientific Reports*, 14(1), 18562. https://doi.org/10.1038/s41598-024-58345-8

Rakesh, M., & Baskar, R. (2024). A support vector machine for lung cancer detection with classification and compared with KNN for better accuracy. *AIP Conference Proceedings*, 2853(1), 020067. https://doi.org/10.1063/5.0198176

Rincy, T.N., & Gupta, R. (2020). Ensemble Learning Techniques and its Efficiency in Machine Learning: A Survey. *2nd International Conference on Data, Engineering and Applications (IDEA)*. p1–6. https://doi.org/10.1109/IDEA49133.2020.9170675

Sachdeva, R.K., Bathla, P., Rani, P., Lamba, R., Ghantasala, G.S.P., & Nassar, I.F. (2024). A novel K-nearest neighbor classifier for lung cancer disease diagnosis. *Neural Computing and Applications*. 36, 22403-22416. https://doi.org/10.1007/s00521-024-10235-w

Wang, S., Shi, J., Ye, Z., Dong, D., Yu, D., Zhou, M., et al. (2019). Predicting EGFR mutation status in lung adenocarcinoma on computed tomography image using deep learning. *European Respiratory Journal*. 53, 1800986. https://doi.org/10.1183/13993003.00986-2018

Yu, L., Tao, G., Zhu, L., Wang, G., Li, Z., Ye, J., et al. (2019). Prediction of pathologic stage in non-small cell lung cancer using machine learning algorithm based on CT image feature analysis. *BMC Cancer*, 19(1), 464. https://doi.org/10.1186/s12885-019-5646-9

Zhou, Z.H. (2012). *Ensemble Methods: Foundations and Algorithms*. 1st ed. Chapman and Hall/CRC, Boca Raton.

AUTHOR BIOGRAPHIES

Lailil Muflikhah is a professor in the field of machine learning at the Faculty of Computer Science, Brawijaya University. She received a B.Sc. degree in computer science from the Institut Teknologi Sepuluh Nopember (ITS). She holds an M.Sc. degree in Computer Science from Universiti Teknologi Petronas (UTP), Malaysia, and a Ph.D. degree in Bioinformatics from Brawijaya University. Her research interests include soft computing, machine learning, and intelligent systems. She can be contacted at email: lailil@ub.ac.id.

Amira Ghina Nurfansepta is a student of Informatics Engineering at the Faculty of Computer Science, Brawijaya University. She has experience working as a research assistant, and her research interests include machine learning. She can be contacted at amiragn25@student.ub.ac.id

Edy Santoso is a lecturer in Informatics Engineering, Faculty of Computer Science, Brawijaya University. He is a member of the Intelligent Computing research group. He holds a Bachelor’s degree in Mathematics with a focus on Computer Science from Brawijaya University, and a Master’s degree in Informatics Engineering from the Sepuluh Nopember Institute, Surabaya, Indonesia.

Agus Wahyu Widodo is a senior lecturer and currently serves as the Vice Dean for General Administration and Finance at the Faculty of Computer Science, Universitas Brawijaya (FILKOM UB). He earned his Bachelor’s degree in Electrical Engineering from Universitas Brawijaya and a master’s degree in Computer Science from Universitas Gadjah Mada. His teaching portfolio includes fundamental and advanced courses in programming, data mining, machine learning, and evolutionary algorithms. His research interests encompass artificial intelligence, machine learning applications, data science, and intelligent systems, as reflected in his extensive publication record in both national and international journals and conferences. He has also contributed to society through various community service programs, particularly in digital literacy, smart village systems, and AI-based education and health-care initiatives. In addition, he has authored academic books and is actively involved in curriculum development and academic leadership.